

**Transformationen zur Vermeidung  
numerischer Dispersion und  
Verfahren zur Steuerung der Zeitschrittweite  
für die numerische Optionsbewertung**

I n a u g u r a l - D i s s e r t a t i o n  
zur  
Erlangung des Doktorgrades  
der Mathematisch-Naturwissenschaftlichen Fakultät  
der Universität zu Köln

vorgelegt von  
Rainer Int-Veen  
aus Uedem

Hundt Druck GmbH, Köln  
2005

---

Berichterstatter: Prof. Dr. R. Seydel  
Prof. Dr. U. Trottenberg

Tag der mündlichen Prüfung: 2. Februar 2005

# Inhaltsverzeichnis

<b>Zusammenfassung/Abstract</b>	<b>7</b>
<b>Notation</b>	<b>8</b>
<b>Einleitung</b>	<b>9</b>
<b>1. Black-Scholes und Erweiterungen</b>	<b>13</b>
1.1. Das Marktmodell nach Black und Scholes . . . . .	13
1.2. Zu den Modellannahmen . . . . .	14
1.2.1. Dividenden . . . . .	14
1.2.2. Zeitabhängigkeit der Parameter . . . . .	15
1.2.3. Abhängigkeit der Parameter vom Aktienkurs . . . . .	15
1.3. Transformationen der Gleichung . . . . .	16
<b>2. Zusammenführung bekannter Arbeiten</b>	<b>19</b>
2.1. TVD-Verfahren und Limiter . . . . .	19
2.1.1. Definitionen . . . . .	19
2.1.2. Numerische Verfahren für Konvektionsgleichungen . . . . .	20
2.1.3. Anwendung der Verfahren auf die Black-Scholes PDE . . . . .	23
2.2. Amerikanische Optionen als Lineares Programm . . . . .	25
2.2.1. Analytische Übertragbarkeit . . . . .	25
2.2.2. Numerische Übertragbarkeit . . . . .	26
2.2.3. Lösungsalgorithmus . . . . .	27
2.3. Kombinierung der beiden Ansätze . . . . .	27
<b>3. Verwenden einer geeigneten Transformation</b>	<b>31</b>
3.1. Dispersion und Dissipation - Eine Analyse . . . . .	31
3.1.1. Einführung . . . . .	31
3.1.2. Dispersion und Dissipation für die Black-Scholes PDE . . . . .	32
3.2. Numerisches Vorgehen . . . . .	34
3.2.1. Log-transformierte Black-Scholes PDE . . . . .	34
3.2.2. Problemanalyse für das Upwind-Schema . . . . .	38
3.3. Suche nach einer geeigneten Transformation . . . . .	39
3.3.1. Wilmotts Transformation auf die Wärmeleitungsgleichung . . . . .	39

3.3.2.	Konvektions-Verschiebung . . . . .	40
3.3.3.	Eine neue Transformation . . . . .	41
3.4.	Eigenschaften der Transformationen . . . . .	43
<b>4.</b>	<b>Modifizierter Algorithmus</b>	<b>45</b>
4.1.	Der Algorithmus von Dempster, Hutton und Richards . . . . .	45
4.2.	Notwendige Abwandlung des Algorithmus . . . . .	48
<b>5.</b>	<b>Vorbereitung der Schrittweitensteuerung</b>	<b>55</b>
5.1.	Motivation . . . . .	55
5.2.	Vorwissen . . . . .	56
5.2.1.	Startwerte sind nur stetig . . . . .	57
5.2.2.	Eingeschränkte Differenzierbarkeit an der Ausübungsgrenze . . . . .	62
5.3.	Verfahrensaufbau . . . . .	66
5.3.1.	Kosten der Berechnung . . . . .	66
5.3.2.	Zusätzlicher Aufwand - Wechsel des Grenzknotens . . . . .	68
5.3.3.	Wahl der Startschrittweite . . . . .	70
<b>6.</b>	<b>Verfahren zur Schrittweitensteuerung</b>	<b>75</b>
6.1.	Mögliche Verfahren . . . . .	75
6.1.1.	Impliziter Euler mit konstanter Schrittweite . . . . .	75
6.1.2.	Steuerung über den Wertzuwachs der Lösung . . . . .	75
6.1.3.	Vergleich mit Verfahren höherer Ordnung . . . . .	78
6.1.4.	Vergleich zweier Lösungen mit unterschiedlicher Schrittlänge . . . . .	81
6.2.	Theoretischer Vergleich der Verfahren . . . . .	82
6.2.1.	Impliziter Euler mit konstanter Schrittweite . . . . .	82
6.2.2.	Steuerung über den Wertzuwachs der Lösung . . . . .	83
6.2.3.	Vergleich mit Verfahren höherer Ordnung . . . . .	83
6.2.4.	Vergleich zweier Lösungen mit unterschiedlicher Schrittlänge . . . . .	84
6.2.5.	Zusammenfassung der Eigenschaften . . . . .	84
<b>7.</b>	<b>Numerische Umsetzung</b>	<b>89</b>
7.1.	Ein erster Praxistest . . . . .	89
7.1.1.	Ein Beispiel zum Einstieg . . . . .	90
7.1.2.	Schrittweitenabsturz bei Implizit/Crank-Nicolson Vergleich . . . . .	92
7.1.3.	Optimierung des Schrittweitenwechsels . . . . .	94
7.2.	Verbinden der Algorithmen . . . . .	97
7.3.	Iteration im Detail . . . . .	99
<b>8.</b>	<b>Leistungsangabe und Vergleich</b>	<b>103</b>
8.1.	Kriterien für die Genauigkeit der Lösung . . . . .	103
8.1.1.	Genauigkeit des Preises . . . . .	103
8.1.2.	Genauigkeit der Ausübungsgrenze . . . . .	104

8.2.	Konsistenztest . . . . .	106
8.2.1.	Optionswert für verschiedene Strikepreise . . . . .	106
8.2.2.	Ausübungsgrenze für verschiedene Strikepreise . . . . .	108
8.3.	Genauigkeitsvergleich . . . . .	108
8.3.1.	Optionswert für verschiedene Steuerungen . . . . .	110
8.3.2.	Ausübungsgrenze für verschiedene Steuerungen . . . . .	111
8.4.	Leistungszuwachs . . . . .	113
8.4.1.	Laufzeiten . . . . .	113
8.4.2.	Durchgeführte Operationen . . . . .	116
8.4.3.	Zusammenfassende Bemerkungen . . . . .	121
<b>9.</b>	<b>Anwendbarkeit auf Exoten</b>	<b>123</b>
9.1.	Basket Optionen . . . . .	123
9.1.1.	Definition . . . . .	123
9.1.2.	Übertragbarkeit der Theorie . . . . .	124
9.1.3.	Reduktion des Aufwandes . . . . .	127
9.1.4.	Steuerung der Zeitschrittweite . . . . .	132
9.2.	Asiatische Optionen . . . . .	133
9.2.1.	Definition . . . . .	133
9.2.2.	Berechnung diskret gemittelter asiatischer Optionen . . . . .	134
9.2.3.	Übertragbarkeit der Theorie . . . . .	137
<b>10.</b>	<b>Zusammenfassung und Ausblick</b>	<b>139</b>
<b>Anhang</b>		<b>141</b>
A.	Berechnungsaufwand . . . . .	141
B.	Bemerkungen zur Implementierung . . . . .	145
C.	Vergleichszahlen . . . . .	147
<b>Verzeichnisse</b>		<b>149</b>
Literaturverzeichnis . . . . .		149
Abbildungsverzeichnis . . . . .		153
Tabellenverzeichnis . . . . .		154
Verzeichnis der Algorithmen und Flussdiagramme . . . . .		155
<b>Erklärung</b>		<b>157</b>



## Zusammenfassung

Die numerische Bewertung amerikanischer Optionen erfordert die Lösung eines linearen Komplementaritätsproblems, dessen Bestandteil die Black-Scholes (Un)gleichung ist. Finite-Differenzen-Diskretisierungen dieser partiellen Differentialgleichung (PDE) können numerische Dispersion erzeugen. Die daraus resultierenden Oszillationen machen ein Hedging der Option unmöglich. Durch geeignete Transformationen ist es möglich, numerische Dispersion zu vermeiden. Diese Arbeit analysiert die Dispersions-Eigenschaften bekannter und neuer Transformationen der Black-Scholes PDE. Für dispersionsfreie Transformationen wird das Problem in eine zeitabhängige Folge von Linearen Programmen überführt. Aufgrund der vorzeitigen Ausübungsmöglichkeit ist für jeden dieser Zeitschritte eine Glättung der numerischen Lösung mittels L-stabiler Diskretisierungsverfahren notwendig. Um den numerischen Aufwand gering zu halten werden verschiedene Verfahren vorgestellt, die die Anzahl der notwendigen Zeitschritte reduzieren. Der Knackpunkt ist dabei der zeitlichen Verlauf der Lösung der PDE. Die Black-Scholes PDE ist eine Verwandte der Wärmeleitungsgleichung und zeigt ein qualitativ ähnliches Verhalten. Es stellt sich heraus, dass diese Schrittweitensteuerung effektiv arbeitet. Trotz der teureren Verwendung einer alternativen Lösung wird ein Speed-up zu den bisher gebräuchlichen Verfahren erreicht. Allerdings steckt der Aufwand nun hauptsächlich im Auffinden der Ausübungsgrenze. Die gefundenen Methoden lassen sich auf einige exotische Optionstypen übertragen.

## Abstract

The numerical valuation of American Options requires the solution of a linear complementary problem, whose component is the Black-Scholes (in)equality. Finite difference discretisations of this partial differential equation (PDE) may introduce numerical dispersion. The resulting oscillations make it impossible to hedge the option. By the use of adequate transformations it is possible to avoid numerical dispersion. This thesis analyses dispersion properties of both known and new transformations of the Black-Scholes PDE. By applying transformations without dispersion, the problem is transmuted into a time-dependent sequence of linear programs. Due to the early exercise feature smoothing of the numerical solution by L-stable methods is necessary in each timestep. Different methods, which reduce the number of necessary timesteps were introduced in order to keep the numerical effort small. The crux thereby is the time dependent behaviour of the PDE. The Black-Scholes PDE is related to the heat-equation and shows the same qualitative behaviour. It turns out that this time-stepping procedure works well. In spite of calculating an expensive alternative solution, a speed-up compared to methods which were formerly used is achieved. But most effort is now put into finding the early exercise curve. It is possible to transfer the developed methods to some exotic options.

## Notation

### Parameter der Option

$t$	Zeit
$\tau$	Transformation der Zeit $t$
$T$	Zeit bis Fälligkeit; Fälligkeitzeitpunkt
$S$	Kurs des Underlyings (meist einer Aktie)
$S_f$	optimale Ausübungsgrenze ( $S$ Koordinaten)
$x$	Transformation des Kurses $S$
$x_f$	optimale Ausübungsgrenze ( $x$ Koordinaten)
$K$	Ausübungspreis; auch Strike genannt
$r$	risikofreier Zinssatz
$\delta$	Dividende, die ein Underlying zahlt
$\sigma$	Volatilität des Underlyings
$V(S, t)$	Preis der Option
$v(x, \tau)$	Preis der Option nach Transformation
$\Psi(S, t)$	Auszahlungsfunktion / Innerer Wert
$\phi(x, \tau)$	Auszahlungsfunktion / Innerer Wert nach Transformation
$\mathcal{L}$	Black-Scholes Differentialoperator
$\mathcal{H}$	Haltebereich der Option; keine vorzeitige Ausübung
$\mathcal{S}$	Stoppbereich; eine vorzeitige Ausübung der Option ist sinnvoll

### Numerische Approximation

$x_{min}, x_{max}$	Gittergrenzen
$\Delta x$	Maschenweite $x$ -Richtung
$\Delta \tau$	Maschenweite $\tau$ -Richtung
$k$	$k$ -ter Knoten in $x$ -Richtung
$m$	$m$ -ter Zeitschritt
$u_k^{(m)}$	Approximation der kontinuierlichen Optionswertes $v$
$y^{(m)} = u^{(m)} - \phi^{(m)}$	Approximation des Zeitwertes der Option
$A$	Matrix des impliziten Teils der Diskretisierung
$B$	Matrix des expliziten Teils der Diskretisierung
$\Theta$	Gewichtungsparameter
$s^{(m)}$	Schlupfvariablen des Linearen Programms
$D$	Matrix für Komplementaritätslösung
$z^{(m)}$	Hilfsvektor aus Schlupfvariablen $s^{(m)}$ und Zeitwert $y^{(m)}$
$U, L$	UL-Zerlegung der Matrix $D$
$w$	Semidiskretisierung von $v$ nach $x$



# Einleitung

Die vorliegende Arbeit beschäftigt sich mit Aspekten der numerischen Bewertung amerikanischer Optionen. Die Bewertung dieser derivativen Finanztitel ist sowohl aus betriebswirtschaftlicher als auch aus mathematischer Sicht ein interessantes Problem. Aus der betriebswirtschaftlichen Sichtweise sind diese Produkte interessant, weil sie sowohl die Möglichkeit zur Absicherung von Risiken (Hedging) bieten als auch über den Hebeleffekt ein enormes Spekulationspotential aufweisen. Die Aufgabe der Bewertung verknüpft andererseits viele Teildisziplinen der angewandten Mathematik. Über die Modellierung der Kursentwicklung mit stochastischen Methoden können partielle Differential(un)gleichungen abgeleitet werden, deren Lösung wiederum Methoden der Variationsrechnung, numerischen Mathematik oder aber der Optimierung erfordert. In dieser Arbeit stehen die Elemente der numerischen Mathematik im Vordergrund, allerdings sollen die Erfordernisse der betriebswirtschaftlichen Anwendung nicht aus den Augen verloren werden.

## Numerische Optionsbewertung

Im Unterschied zu den europäischen Optionen, deren Bewertung in der fundamentalen Arbeit von Black und Scholes [BS73] beschrieben wurde, beinhalten amerikanische Optionen die Möglichkeit zur vorzeitigen Ausübung während der gesamten Laufzeit des Derivates. Das Bewertungsproblem erschwert sich durch diese Eigenschaft von der Lösung einer partiellen Differentialgleichung (PDE) zur Lösung eines freien Randwertproblems mit zugehöriger partieller Differentialungleichung. In diesem Zusammenhang wird auch von einem Komplementaritätsproblem gesprochen, für dessen Bewertung auf ein großes Spektrum von numerischen Methoden zurückgegriffen werden kann. Dazu zählen auf der Kursentwicklung aufsetzende Methoden, wie die erstmals von Cox, Ross und Rubinstein [CRR79] vorgestellten Baumverfahren oder Monte-Carlo Verfahren, über deren Entwicklungsstand das Buch von Glasserman [Gla04] einen Überblick gibt. Des Weiteren existieren Näherungsverfahren, die von der analytischen Lösung für verwandte europäische Optionen ausgehend, eine Approximation der amerikanischen Option liefern. Dazu zählt z.B. die Arbeit von Geske und Johnson [GJ84].

Nicht zuletzt sei eine Auswahl von Methoden genannt, die auf der Black-Scholes PDE bzw. deren Verallgemeinerungen aufsetzen. Diese reichen von Ansätzen mit Finiten-Differenzen,

wie in der Arbeit von Brennan und Schwartz [BS78], über die Anwendung von numerischen Flussbegrenzern durch Zvan, Forsyth und Vetzal [ZFV98], bis zur Anwendung von Mehrgitterverfahren durch Clark und Parrott [CP99]. Daneben gibt es Nischen, in denen auf dem Newtonverfahren aufbauende Löser, wie der von Coleman, Li und Verma [CLV02], sowie die auf Linearer Optimierung aufbauende Methoden aus der Arbeitsgruppe um Dempster [DH97, DHR98] anzusiedeln sind.

Eine Gemeinsamkeit der auf der PDE aufbauenden Verfahren ist das zeitschichtweise Vorgehen bei der Berechnung der Lösung. Im Zeitablauf, der Lebensdauer der Option, werden hintereinander viele ähnliche Probleme gelöst. Es erweist sich als vorteilhaft, die Struktur dieser Abfolge auszunutzen, um die Laufzeit des Verfahrens zu verkürzen.

In diesem Punkt kommt einer der Konflikte der Anwendung zum Tragen. Der betriebswirtschaftliche Nutzer ist an einer schnellen, aber möglichst genauen Lösung interessiert. Beide Bedingungen gleichzeitig sind schwer zu erfüllen. Eine genauere Berechnung erfordert ein Gitter mit feiner Maschenweite, die gleichzeitig zu einem hohen numerischen Aufwand führt. Die Möglichkeit die Länge der Zeitschritte zu variieren ist in der Numerik von Differentialgleichungen ein gebräuchliches Vorgehen. Für die Bewertung von Optionen ist uns jedoch nur die Verwendung eines von Eriksson und Johnson stammenden Verfahrens durch die Arbeitsgruppe Forsyth/Vetzal, erstmalig in [FV02], bekannt. In dieser Arbeit sollen weitere Algorithmen zur Steuerung der Zeitschrittweite bei der Optionsbewertung hergeleitet werden.

Innerhalb der Aufzählung der numerischen Verfahren ist die Verwendung von Flussbegrenzern erwähnt worden. Diese werden eingesetzt, um entstehende numerische Oszillationen zu eliminieren, so dass die Lösung des Optionsbewertungsproblems weiterhin die Möglichkeit des Hedgings zulässt. Diese nichtlinearen Limiter sind jedoch mit dem Ansatz über das Lineare Programm unvereinbar. Wir wollen zeigen, dass es trotzdem eine Möglichkeit gibt, mit diesem schnellen Algorithmus zu einer oszillationsfreien Lösung des Komplementaritätsproblems zu gelangen.

## Aufbau der Arbeit

Die Arbeit lässt sich grob in vier Blöcke einteilen. Wir starten mit der Vorstellung des erweiterten Modells von Black und Scholes in Kapitel 1. Dabei soll zum einen das nötige Handwerkszeug für die Blöcke zwei und drei bereitgestellt werden, zum anderen soll ein Ausschnitt der Entwicklung des Modells nachvollzogen werden.

Der zweite Block beschäftigt sich mit der Unterdrückung numerischer Dispersion und damit der oszillationsfreien Berechnung von Optionswerten. Kapitel 2 liefert sowohl eine Zusammenfassung der Eigenschaften von Differenzenverfahren für Optionen als auch eine Erläuterung der Übertragbarkeit der Problemstellung in ein Lineares Programm. Mit der durch den Anwender bestimmten Zielvorgabe, schnell eine oszillationsfreie Lösung zu erhalten, zeigt sich die Unvereinbarkeit der Ansätze. Die Ursache des Problems wird in Kapitel 3

untersucht und die Möglichkeit, die Schwierigkeiten durch geschickte Transformation zu umgehen erörtert. Die in der Theorie positiven Ergebnisse machen in der Praxis eine Modifikation des Basisalgorithmus von Dempster, Hutton und Richards (DHR-Algorithmus) notwendig. Diese Abwandlung wird zusammen mit der Analyse des Algorithmus im Kapitel 4 vorgenommen.

Die angesprochene Zeitschrittweitensteuerung bildet den roten Faden für den dritten Block. Das Kapitel 5 motiviert ein solches Vorgehen und beschäftigt sich mit einer a priori Analyse für mögliche Verfahren. Dabei spielen insbesondere die nicht ausreichende Glattheit der Lösung und die Teilelemente des DHR-Algorithmus eine Rolle. Mit Hilfe der Analyse werden die in Kapitel 6 vorgestellten alternativen Verfahren in der Theorie beschrieben und verglichen. Es erfolgen erste numerische Tests in Kapitel 7. Diesen Basistests schließt sich eine detaillierte Betrachtung des Verfahrensablaufs an, um genauere Erkenntnisse über das Verhalten der Algorithmen zu gewinnen. Im Kapitel 8 werden Kriterien für die Praxistauglichkeit der Verfahren entwickelt, mit denen im Anschluss ein Vergleich der Verfahren vorgenommen werden kann. Den Abschluss dieses Kapitels bildet die Gegenüberstellung der Laufzeiten der Algorithmen mit einer Zeitschrittweitensteuerung, bei der ein Leistungsgewinn gegenüber den herkömmlichen Methoden dokumentiert wird.

Im vierten Block wird zuerst ein Blick auf verallgemeinerte Probleme und die Übertragbarkeit der Resultate geworfen. Dazu setzt sich Kapitel 9 mit Basket-Optionen und asiatischen Optionen auseinander. Den Abschluss der Arbeit bildet dann die Zusammenfassung der Ergebnisse im letzten Kapitel.

## Dank

*An dieser Stelle danke ich Herrn Prof. Dr. R. Seydel für seine Unterstützung, vor allem für seine ständige Diskussionsbereitschaft und seine wertvollen Hinweise.*



# 1. Black-Scholes und Erweiterungen

Grundlegend für die Optionsbewertung ist das Marktmodell von Black und Scholes. Auch wenn das Modell in der Zwischenzeit erweitert bzw. abgewandelt wurde, so findet es, oft auch noch in der Urform, Eingang in die Modellierung.

## 1.1. Das Marktmodell nach Black und Scholes

Black und Scholes haben in ihrer fundamentalen Arbeit von 1973 [BS73] die partielle Differentialgleichung (PDE)

$$V_t + \frac{\sigma^2}{2} S^2 V_{SS} + r S V_S - r V = 0 \quad (1.1)$$

für den Preis einer europäischen Option hergeleitet. Für Optionen europäischen Typs mit geeigneten Anfangs- und Randbedingungen liefern Black und Scholes in dieser Arbeit gleichzeitig eine Lösungsformel. In ihrer Modellierung unterstellen sie einen idealen Markt mit den **Annahmen**:

- i) Der risikofreie Zinssatz  $r$  ist bekannt und über die Laufzeit der Option konstant.
- ii) Die Volatilität der Aktie  $\sigma$  ist konstant.
- iii) Die Aktie folgt einer geometrischen Brownschen Bewegung

$$dS(t) = S r dt + S \sigma dW_t.$$

- iv) Die Aktie zahlt keine Dividende oder andere Erträge.
- v) Beim Handeln der Wertpapiere entstehen keine Transaktionskosten.
- vi) Wertpapiere sind beliebig teilbar, insbesondere ist es möglich, nicht ganzzahlige Stückzahlen von Aktien zu handeln.
- vii) Es gibt keine Beschränkungen für Leerverkäufe.

Dieses Modell ist von Merton 1973 [Mer73] um eine kontinuierliche, konstante Dividende auf die Gleichung

$$V_t + \frac{\sigma^2}{2} S^2 V_{SS} + (r - \delta) S V_S - r V = 0 \quad (1.2)$$

und gleichzeitig auch auf zeitabhängige Parameter mit der Gleichung

$$V_t + \frac{\sigma^2(t)}{2} S^2 V_{SS} + (r(t) - \delta(t)) S V_S - r(t) V = 0 \quad (1.3)$$

erweitert worden. Merton hat dabei gezeigt, dass sich die Theorie von Black und Scholes auch unter diesen schwächeren Annahmen verwenden lässt.

Schließlich haben Cox und Ross [CR76] das Modell 1976 auf seine allgemeinste Form, mit sowohl zeit- als auch aktienkursabhängigen Parametern, erweitert. Wir werden jedoch im Folgenden nicht diese am weitesten verallgemeinerte Gleichung

$$V_t + \frac{\sigma^2(S, t)}{2} S^2 V_{SS} + (r(S, t) - \delta(S, t)) S V_S - r(S, t) V = 0 \quad (1.4)$$

verwenden. Statt dessen nehmen wir Einschränkungen der Modellannahmen an  $r$  und  $\delta$  vor und lassen dort lediglich die Zeitabhängigkeit zu.

## 1.2. Zu den Modellannahmen

Natürlich entsprechen die Annahmen von Black und Scholes aus Abschnitt 1.1 nicht der Realität. Wie man schon an den Erweiterungen der klassischen Black-Scholes Gleichung (1.1) sieht, sind insbesondere die Annahmen zu Zinsrate  $r$ , der Zahlung von Dividenden und der Volatilität  $\sigma$  aufgelockert worden.

### 1.2.1. Dividenden

Für die Zahlung von Dividenden existieren im Wesentlichen zwei Modellierungsansätze. Zum einen die technisch einfachere Annahme, dass eine kontinuierliche Auszahlung an den Inhaber der Aktie stattfindet. Sie führt auf die abgewandelte Gleichung (1.2), welche sich von der klassischen Black-Scholes Gleichung (1.1) nur im Konvektionsterm unterscheidet und numerisch auf die gleiche Weise zu handhaben ist.

Zum anderen verwendet man eine Annahme, die von diskreten Dividendenzahlungen zu festgelegten Zeitpunkten ausgeht. Diese Annahme gibt die Realität besser wieder als die kontinuierliche Dividende. Für das Zeitintervall zwischen jeweils zwei Zahlungszeitpunkten ist die klassische Black-Scholes Gleichung zu lösen. Am Zeitpunkt  $t_d$  der Dividendenzahlung ist numerisch eine Sprungbedingung der Form

$$V(S, t_d^-) := V(S(1 - \delta_{t_d}), t_d^+) \quad (1.5)$$

zu implementieren. Die Herleitung solcher Sprungbedingungen findet sich z.B. in dem Buch von Wilmott et al. [WHD95, Seite 94 ff.].

### 1.2.2. Zeitabhängigkeit der Parameter

Eine Abhängigkeit der Parameter  $r$ ,  $\delta$  und  $\sigma$  von der Restlaufzeit  $\tau$  ist ohne weiteres mit dem, um Dividenden erweiterten, Black-Scholes Modell zu vereinbaren. Zeitabhängige Parameter sind allgemein akzeptiert. Sogar die Lösungsformel für den europäischen Fall lässt sich ohne Einschränkungen übernehmen. Es sind lediglich die konstanten Parameter durch Integrale über die Restlaufzeit zu ersetzen. So wird zum Beispiel statt  $\exp(-r\tau)$  der Ausdruck  $\exp(\int_0^\tau r(T-u) du)$  verwendet. Die so abgewandelte Lösungsformel findet sich z.B. im Buch von Kwok [Kwo98, Kapitel 2.2].

### 1.2.3. Abhängigkeit der Parameter vom Aktienkurs

Insbesondere für den risikofreien Zinssatz  $r$  ist es per definitionem klar, dass dieser nicht von einem riskanten Wertpapier, in diesem Fall der Aktie  $S$ , abhängt.

Für die an das Wertpapier gekoppelte Dividende lässt sich hier der Einwand bringen, dass eine Zahlung durchaus vom Kurs  $S$  abhängen kann. Die Dividende wird z.B. nur dann gezahlt, falls die Bedingung  $S \geq D$  für eine Zahlung der Höhe  $D$  erfüllt ist. Dies lässt sich aber unserer Ansicht nach wesentlich besser über diskrete Dividendenzahlungen modellieren. Eine Modellierung in dieser Form führt auf die Verwendung der Sprungbedingung (1.5) an den Zahlungszeitpunkten.

Sowohl die Zinsrate als auch die Dividendenrate lässt sich aus Preisen anderer Wertpapiere ermitteln. Die Zinsrate lässt sich aus den Preisen von Zerobonds über

$$Z(t_0, t_1) := \exp\left(-\int_{t_0}^{t_1} r(u) du\right) \quad (1.6)$$

ermitteln. Danach lässt sich die Dividendenrate mit dem bekanntem  $r$  aus den Preisen für Forwards, mit Hilfe der Gleichung

$$F(t_0, t_1) := S_0 \exp\left(\int_{t_0}^{t_1} r(u) - \delta(u) du\right), \quad (1.7)$$

errechnen. Dabei werden die Gleichungen nach  $r$  bzw.  $\delta$  aufgelöst. Weitere Ausführungen hierzu findet man insbesondere in der Arbeit von Andersen und Brotherton-Ratcliffe [ABR98, Abschnitt 2].

Für die Volatilität ergibt sich das von Dupire [Dup94] sowie Derman und Kani [DK94] beschriebene Konzept des ‚Smiles‘. Hier wird ausgehend von Marktdaten eine implizite Struktur berechnet. Für das exakte Vorgehen verweisen wir nochmals auf die oben

genannte Arbeit von Andersen und Brotherton-Ratcliffe [ABR98] sowie auf die Arbeit von Dupire. In jedem Falle ist bei einem verallgemeinerten Problem die Volatilität abhängig vom aktuellen Aktienkurs  $S$ .

Natürlich bietet es sich als weitere Möglichkeit an, die Volatilität, genauso wie auch die beiden anderen Parameter, über zusätzliche Gleichungen zu simulieren.

Für die weitere Arbeit nehmen wir sowohl die Parameter

$$\begin{array}{ll} r(t) & \text{Zinssatz,} \\ \delta(t) & \text{Dividende,} \\ \text{als auch } \sigma(S, t) & \text{Volatilität} \end{array}$$

als gegeben an.

### 1.3. Transformationen der Gleichung

Ausgehend von zuerst einmal konstanten Parametern  $r$ ,  $\delta$  und  $\sigma$  existiert eine Fülle von Transformationen der verallgemeinerten Black-Scholes PDE (1.2) auf Gleichungen mit konstanten Koeffizienten. Alle nachfolgenden Gleichungen sind analytisch äquivalent zu Black-Scholes, haben jedoch numerisch entscheidend unterschiedliche Eigenschaften. Mit dieser Tatsache beschäftigen wir uns in Kapitel 3 .

#### Log-transformierte Black-Scholes PDE

Diese Transformation geht zurück auf Brennan und Schwarz [BS78]. In der hier verwendeten Form wird zusätzlich noch eine Zeitumkehr durchgeführt. Eine weitere Ergänzung ist es, die neuen Koordinaten ebenfalls noch über den Strikepreis  $K$  zu normieren (wie z.B. im Buch von Hull [Hul00]). Dies hat später nur einen Einfluss auf die Wahl der numerischen Randbedingungen. Die Transformation

$$\begin{array}{l} \tau := T - t, \\ x := \ln\left(\frac{S}{K}\right) \end{array}$$

liefert die Gleichung

$$v_\tau - \frac{\sigma^2}{2} v_{xx} - \left(r - \delta - \frac{\sigma^2}{2}\right) v_x + r v = 0. \quad (1.8)$$

#### **Bemerkung 1.3.1. Am weitesten verbreitete Transformation**

*Diese Gleichung stellt so etwas wie die ‚Drosophila‘ der Black-Scholes Transformationen dar. Sie taucht in den meisten Lehrbüchern und in sehr vielen Arbeiten zur numerischen Lösung der Black-Scholes PDE auf.*



### Transformation auf Wärmeleitungsgleichung

Wilmott et al. [WHD95] zeigen in ihrem Buch, dass die Wärmeleitungsgleichung äquivalent zur Black-Scholes PDE ist. Dazu transformieren sie in zwei Schritten. Die Definition von

$$\begin{aligned}\tau &:= (T - t) \frac{\sigma^2}{2}, \\ x &:= \ln\left(\frac{S}{K}\right), \\ \widehat{V}(x, \tau) &:= \frac{V(S, t)}{K}\end{aligned}$$

führt auf den Zwischenschritt

$$\widehat{V}_\tau - \widehat{V}_{xx} - (q_\delta - 1)\widehat{V}_x + qv = 0$$

mit

$$q := \frac{2r}{\sigma^2}, \quad q_\delta := \frac{2(r - \delta)}{\sigma^2}.$$

Der zweite Schritt führt über das Setzen von

$$\widehat{V}(x, \tau) =: e^{-\frac{1}{2}(q_\delta - 1)x - \frac{1}{4}((q_\delta - 1)^2 + 4q)\tau} v(x, \tau)$$

auf die Wärmeleitungsgleichung

$$v_\tau - v_{xx} = 0. \tag{1.9}$$

### Konvektions-Verschiebung

Eine weitere Transformation wurde zuerst bei Barraquand und Pudet [BP96] erwähnt und in ähnlicher Form auch von Smith [Smi00] verwendet. Sie stellt eine Weiterentwicklung der log-Transformation (1.8) dar. Das neue Koordinatensystem wird im Zeitablauf synchron zur Konvektion verschoben. Wir benutzen deswegen die Bezeichnung ‚Konvektions-Verschiebung‘. Der Ansatz von Barraquand und Pudet

$$\begin{aligned}\tau &:= T - t, \\ x &:= \ln\left(\frac{S}{K}\right) - Rt \quad \text{mit} \quad R := r - \delta - \frac{\sigma^2}{2}\end{aligned}$$

bzw. Smith

$$\begin{aligned}\tau &:= T - t, \\ x &:= \ln\left(\frac{S}{K}\right) + \int_t^T R(y) dy \quad \text{mit} \quad R(t) := r(t) - \delta(t) - \frac{\sigma^2(t)}{2}\end{aligned}$$

liefert in beiden Fällen das Zwischenresultat

$$\widehat{V}_\tau - \frac{\sigma^2}{2} \widehat{V}_{xx} + r \widehat{V} = 0. \tag{1.10}$$

Auch hier erhält man über  $\widehat{V}(x, \tau) =: e^{-r\tau} v(x, \tau)$  eine Form der Wärmeleitungsgleichung

$$v_\tau - \frac{\sigma^2}{2} v_{xx} = 0.$$

### Eine neue Transformation

Im Rahmen dieser Arbeit haben wir eine neue Transformation entwickelt, welche die Ideen der Wilmott Transformation (1.9) und der Konvektions-Verschiebung (1.10) kombiniert. Setzen von

$$\begin{aligned} \tau &:= T - t, \\ x &:= \ln\left(\frac{S}{K}\right) + \int_t^T R(y) dy \quad \text{mit} \quad R(t) := r(t) - \delta(t) \end{aligned}$$

liefert als Zwischenschritt

$$\widehat{V}_\tau - \frac{\sigma^2}{2} [\widehat{V}_{xx} - \widehat{V}_x] + r \widehat{V} = 0.$$

Mittels  $\widehat{V}(x, \tau) =: e^{\frac{1}{2}x - \int_0^\tau r(t-y) dy} v(x, \tau)$  können wir diesen auf die Gleichung

$$v_\tau - \frac{\sigma^2}{2} \left[ v_{xx} - \frac{1}{4}v \right] = 0 \tag{1.11}$$

transformieren. Eine detaillierte Herleitung findet sich in Kapitel 3.3.3.

## 2. Zusammenführung bekannter Arbeiten

In vorangehenden Arbeiten anderer Arbeitsgruppen werden im Wesentlichen zwei Resultate für die Bewertung von amerikanischen Optionen geliefert. Dies ist zum einen die Anwendung von (Totalvariation vermindernenden) TVD-Verfahren durch die Arbeitsgruppe Forsyth/Vetzal mit den Arbeiten von Zvan [ZFV98, ZFV01]. Hierdurch wird es möglich, oszillationsfreie Optionswerte und damit auch Hedgeparameter zu berechnen. Zum anderen sind dies Arbeiten der Arbeitsgruppe Dempster, die eine Umformulierung des diskreten Problems auf ein Lineares Programm beschreiben. Diese Formulierung macht einen direkten Zugang des Problems ohne iterative Lösungsverfahren möglich. Hierzu liegen mehrere Arbeiten [DHR98, DH99, DR00] vor.

Das Vorhaben ist es, die beiden Ideen zusammenzuführen. Dazu werden beide wiederholend analysiert und schließlich die Unmöglichkeit eines solchen Zugangs beschrieben.

Der Übersicht halber beschränken wir uns in der Darstellung auf die normierte log-transformierte Black-Scholes Gleichung (vgl. (1.8))

$$v_\tau - \frac{\sigma^2}{2} v_{xx} - \left(r - \delta - \frac{\sigma^2}{2}\right)v_x + r v = 0 \quad (2.1)$$

mit konstanten Parametern  $r$ ,  $\delta$  und  $\sigma$ .

### 2.1. TVD-Verfahren und Limiter

Bevor wir uns den eigentlichen Resultaten zuwenden können, sind einige Definitionen und Wiederholungen notwendig.

#### 2.1.1. Definitionen

Wir beginnen mit einigen Begriffen aus der Flüssigkeits- und Gasdynamik.

**Definition 2.1.1. Péclet-Zahl**

*Die Péclet-Zahl ist ein Maß für das Verhältnis von Konvektion zu Diffusion in Konvektions-Diffusions-Gleichungen. Am Beispiel der log-transformierten Black-Scholes Gleichung gilt:*

$$Pe := \frac{r - \delta - \sigma^2/2}{\sigma^2/2} = \frac{2(r - \delta)}{\sigma^2} - 1. \quad (2.2)$$

**Bemerkung 2.1.2.** *Wilmotts  $q_\delta$  ist mit der Péclet-Zahl verwandt.*

*Hier gilt mit der Definition 2.1.1 und der Definition des  $q_\delta$  in der Transformation auf die Wärmeleitungsgleichung nach Wilmott (vgl. Ausführung vor Gleichung (1.9)):*

$$Pe := q_\delta - 1.$$

In der Literatur, z.B. dem Buch von Wesseling [Wes01], wird dieses Verhältnis oft in Proportion zu einer charakteristischen Länge des gegebenen Problems gesetzt. Dies führt auf

**Definition 2.1.3.** *Gitter Péclet-Zahl*

*Die Gitter Péclet-Zahl gibt die Relation zwischen Konvektion und Diffusion für das jeweils gewählte Gitter an. Im Unterschied zur Definition 2.1.1 fließt hier insbesondere die charakteristische Zellengröße des Gitters mit ein. Im betrachteten Fall ist dies gemäß der Definition aus Wesseling [Wes01]*

$$Pe_{\Delta x} := \Delta x \left| \frac{2(r - \delta)}{\sigma^2} - 1 \right|. \quad (2.3)$$

Um das Auftreten und vor allem die Vermeidung von Oszillationen zu beschreiben, nutzt man die folgende Definition:

**Definition 2.1.4.** *Totalvariation vermindern (TVD) [Swe84]*

*Es seien mit  $u_k^{(m)}$  die Werte einer Approximation der Lösung einer PDE bezeichnet. Unter der totalen Variation der Lösung im  $m$ -ten Zeitschritt versteht man die Summe*

$$TV(u^{(m)}) = \sum_k \left| u_{k+1}^{(m)} - u_k^{(m)} \right|. \quad (2.4)$$

*Ein Finite-Differenzen-Schema heißt dann Totalvariation vermindern, wenn gilt:*

$$TV(u^{(m+1)}) \leq TV(u^{(m)}). \quad (2.5)$$

## 2.1.2. Numerische Verfahren für Konvektionsgleichungen

Im Hinblick auf die Übertragbarkeit auf den Black-Scholes Fall verwenden wir in diesem Unterabschnitt, wie schon global, die Bezeichnungen  $v$  für die kontinuierliche Lösung und  $u$  für die numerische Approximation einer Gleichung. Die Variablen bezeichnen dabei jedoch keine Optionswerte.

Anhand der Modellgleichung

$$v_\tau + a v_x = 0 \quad (2.6)$$

sollen kurz die Eigenschaften der numerischen Verfahren wiederholt werden. Um die TVD Eigenschaft eines Differenzen-Schemas zu zeigen, bedienen wir uns des folgenden Lemmas:

**Lemma 2.1.5. TVD-Eigenschaft für Finite-Differenzen**

Ein Differenzen-Schema der Form

$$u_k^{(m+1)} = \sum_k c_k u_k^{(m)} \quad (2.7)$$

ist genau dann TVD, wenn für jedes  $k$  gilt:  $c_k \geq 0$  und  $\sum_k c_k \leq 1$ .

**Beweis:**

Kapitel 9.7 im Buch von Thomas [Tho99]. Diese Aussage entspricht der dortigen Proposition 9.7.6.  $\square$

Es folgt der angekündigte kurze Überblick über häufig verwendete Finite-Differenzen-Schemata (FD-Schemata).

**Lemma 2.1.6. Eigenschaften häufig verwendeter FD-Schemata**

Für die Modellgleichung (2.6) gelten die Aussagen:

i) Eine Diskretisierung mit zentralen **Finiten-Differenzen** der Form

$$u_k^{(m+1)} = u_k^{(m)} - a \frac{\Delta\tau}{2\Delta x} (u_{k+1}^{(m)} - u_{k-1}^{(m)}) \quad (2.8)$$

ist von der Ordnung  $\mathcal{O}(\Delta x^2)$  und numerisch instabil.

ii) Eine Diskretisierung mit dem **Upwind-Schema**

$$u_k^{(m+1)} = \begin{cases} u_k^{(m)} - a \frac{\Delta\tau}{\Delta x} (u_k^{(m)} - u_{k-1}^{(m)}) & \text{für } a > 0 \\ u_k^{(m)} - a \frac{\Delta\tau}{\Delta x} (u_{k+1}^{(m)} - u_k^{(m)}) & \text{für } a < 0 \end{cases} \quad (2.9)$$

ist von der Ordnung  $\mathcal{O}(\Delta x)$ . Zusätzlich wird die Diskretisierung TVD, falls  $|a| \frac{\Delta\tau}{\Delta x} \leq 1$  erfüllt ist. In diesem Fall ist sie auch numerisch stabil.

iii) Die Verwendung des **Lax-Friedrich** Schemas

$$\begin{aligned} u_k^{(m+1)} &= \frac{1}{2} (u_{k+1}^{(m)} + u_{k-1}^{(m)}) - a \frac{\Delta\tau}{2\Delta x} (u_{k+1}^{(m)} - u_{k-1}^{(m)}) \\ &= u_k^{(m)} - a \frac{\Delta\tau}{2\Delta x} (u_{k+1}^{(m)} - u_{k-1}^{(m)}) \\ &\quad + \frac{\Delta x^2}{2\Delta\tau} \frac{\Delta\tau}{\Delta x^2} (u_{k+1}^{(m)} - 2u_k^{(m)} + u_{k-1}^{(m)}) \end{aligned} \quad (2.10)$$

erzeugt zusätzliche numerische Diffusion. Das Schema hat die Ordnung  $\mathcal{O}(\Delta x^2)$  und ist für  $|a| \frac{\Delta\tau}{\Delta x} \leq 1$  TVD.

iv) Das **Lax-Wendroff** Schema [LW60]

$$\begin{aligned}
 u_k^{(m+1)} &= u_k^{(m)} - a \frac{\Delta\tau}{\Delta x} (u_k^{(m)} - u_{k-1}^{(m)}) \\
 &\quad - \frac{a\Delta x - a^2\Delta\tau}{2} \frac{\Delta\tau}{\Delta x^2} (u_{k+1}^{(m)} - 2u_k^{(m)} + u_{k-1}^{(m)}) \\
 &= u_k^{(m)} - a \frac{\Delta\tau}{2\Delta x} (u_{k+1}^{(m)} - u_{k-1}^{(m)}) + a^2 \frac{\Delta\tau^2}{2\Delta x^2} (u_{k+1}^{(m)} - 2u_k^{(m)} + u_{k-1}^{(m)}),
 \end{aligned} \tag{2.11}$$

hier nur für den Fall  $a > 0$ , stellt eine Kombination der Ideen vom Upwind-Schema und von Lax-Friedrich dar. Es erzeugt ebenfalls zusätzliche numerische Diffusion und ist von der Ordnung  $\mathcal{O}(\Delta x^2)$ . Allerdings besteht die starke Einschränkung, dass es genau dann TVD ist, wenn  $a \frac{\Delta\tau}{\Delta x} = 1$  gilt.

v) Ein **Limitier Schema** [Swe84] der Form

$$\begin{aligned}
 u_k^{(m+1)} &= u_k^{(m)} - a \frac{\Delta\tau}{2\Delta x} (u_k^{(m)} - u_{k-1}^{(m)}) \\
 &\quad - \frac{a\Delta x - a^2\Delta\tau}{2} \frac{\Delta\tau}{\Delta x^2} \left[ \phi(\lambda_j) (u_{k+1}^{(m)} - u_k^{(m)}) - \phi(\lambda_{j-1}) (u_k^{(m)} - u_{k-1}^{(m)}) \right]
 \end{aligned} \tag{2.12}$$

mit

$$\lambda_j =: \frac{u_k^{(m)} - u_{k-1}^{(m)}}{u_{k+1}^{(m)} - u_k^{(m)}} \tag{2.13}$$

und einer nichtlinearen Funktion  $\phi(\lambda)$  zeichnet sich durch gemischte Eigenschaften aus. Durch die Wahl der Funktion  $\phi$  wird das Schema TVD. Die Ordnung wechselt durch die Wahl von  $\phi$  zwischen  $\mathcal{O}(\Delta x)$  und  $\mathcal{O}(\Delta x^2)$ . Gegebenenfalls wird zusätzliche numerische Diffusion erzeugt. Im Gegensatz zu den anderen Diskretisierungen ist dieses Schema **nichtlinear!**

**Beweis:**

Die getroffenen Ordnungsaussagen lassen sich über Taylorentwicklung beweisen. Der Beweis der TVD Bedingungen erfolgt mittels Lemma 2.1.5. Mit der von Neumann Stabilitätsanalyse werden die Aussagen zur Stabilität nachgerechnet.  $\square$

Wir haben das Resultat von Zvan et al. angesprochen. Diese benutzen den van Leer Limiter, der über die nachfolgende Definition bestimmt wird.

**Definition 2.1.7. van Leer Limiter [vL74]**

Die Flussbegrenzer (Limiter) Funktion  $\phi$  hat hier die Gestalt:

$$\phi(\lambda) = \frac{|\lambda| + \lambda}{1 + |\lambda|} = \begin{cases} 0 & \text{falls } \lambda \leq 0, \\ \frac{2\lambda}{1+|\lambda|} & \text{falls } \lambda > 0. \end{cases} \quad (2.14)$$

Die Notation ist dabei der Arbeit von Sweby [Swe84] entnommen.

**2.1.3. Anwendung der Verfahren auf die Black-Scholes PDE**

Zvan et al. haben die Anwendung einiger der vorangehenden Verfahren auf die (transformierte) Black-Scholes Gleichung für den Fall sehr niedriger Volatilität  $\sigma$  betrachtet. Es handelt sich also um eine konvektionsdominante Gleichung, deren Verhalten im Wesentlichen durch den Transport der Lösung beschrieben wird.

Ein Hauptresultat ihrer Arbeit lässt sich zusammenfassen zu

**Lemma 2.1.8. Bedingungen für die TVD-Diskretisierung der Black-Scholes PDE [ZFV98]**

- i) Für die Diskretisierung der log-transformierten Black-Scholes PDE mit zentralen Finiten-Differenzen ergeben sich die Bedingungen:

$$\frac{1}{\Delta x} > \frac{|r - \delta - \frac{\sigma^2}{2}|}{\sigma^2}, \quad (2.15a)$$

$$\frac{1}{(1 - \Theta)\Delta\tau} \geq \frac{\sigma^2}{\Delta x^2} + r. \quad (2.15b)$$

- ii) Die Diskretisierung der log-transformierten Black-Scholes PDE mit dem Upwind-Schema führt auf die Bedingung:

$$\frac{1}{(1 - \Theta)\Delta\tau} \geq \frac{\sigma^2}{\Delta x^2} + \frac{|r - \delta - \frac{\sigma^2}{2}|}{\Delta x} + r. \quad (2.16)$$

**Beweis:**

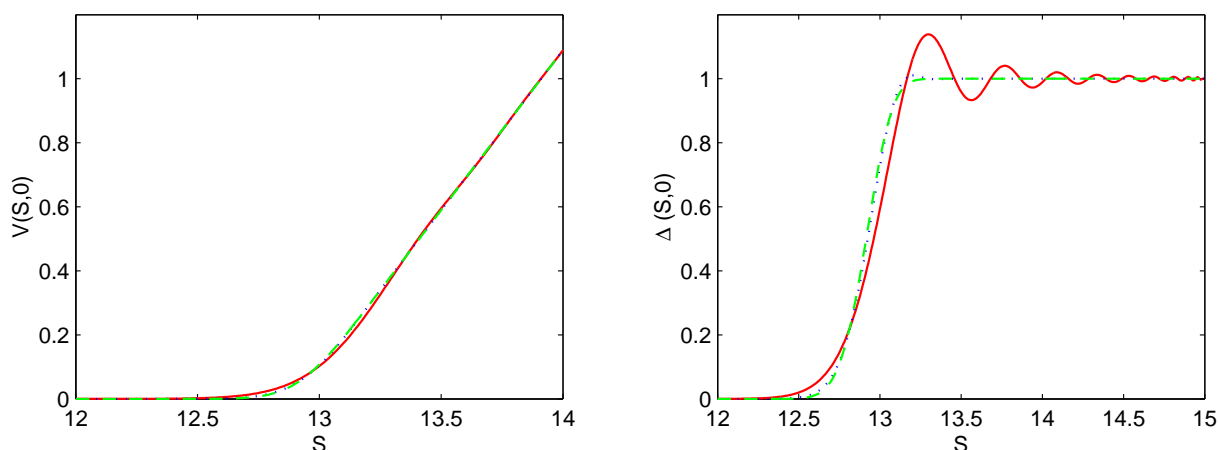
Die Resultate lassen sich aus der jeweils verwendeten Diskretisierung mittels Lemma 2.1.5 nachrechnen. Dabei ist die jeweilige Bedingung analog zu dem Resultat in [ZFV98] mit einer Abwandlung auf

- i) den Fall einer kontinuierlichen Dividende,
- ii) die log-transformierte Gleichung. □

**Bemerkung 2.1.9. Péclet-Bedingung für Black-Scholes**

In die rechte Seite der Gleichungen (2.15a) geht die Péclet-Zahl in der Form  $\frac{1}{2}|q_\delta - 1|$  ein.

Mit diesen Resultaten zeigen Zvan et al. anschaulich die Auswirkungen, falls eine der Bedingungen (2.15a) bis (2.16) verletzt wird. Die numerische Lösung der Black-Scholes Gleichung ist mit numerischen Oszillationen behaftet, weil die verwendeten Verfahren nicht mehr die TVD-Bedingungen erfüllen. Als Folge hieraus sind insbesondere die weiter



(a) Approximation des Optionswertes  $V$  für  $M = 20, 50, 100$  Zeitschritte

(b) Approximation des Optionsdelta für  $M = 20, 50, 100$  Zeitschritte

Abbildung 2.1.: Oszillationen in der Näherungslösung des Optionswertes bzw. des Optionsdelta für verschiedene Zeitschrittweiten in Anlehnung an [ZFV98, Abb. 2].

aufgerauten Approximationen der ersten und zweiten Ableitung nach dem Kurs,  $\Delta$  und  $\Gamma$ , nicht mehr für das Absichern von Aktienpositionen geeignet. Die in Abbildung 2.1 veranschaulichte Lösung ist somit betriebswirtschaftlich wertlos.

Als weiteres Ergebnis zeigen Zvan et al., dass die Verwendung des Upwind-Schemas zwar eine oszillationsfreie, allerdings sehr weit verschmierte Lösung produziert. Eine Verwendung, um die Hedgeparameter  $\Delta$  und  $\Gamma$  zu approximieren, ist hier ebenfalls nicht mehr sinnvoll. Abschließend zeigen Zvan et al. die Ermittlung einer Näherungslösung mittels des Einsatzes des van Leer Limiters. Hierbei treten weder Oszillationen auf, noch ist die zusätzliche numerische Diffusion so stark, dass die Lösung extrem verschmiert wird.

**Bemerkung 2.1.10. Lösung eines nichtlinearen Gleichungssystems**

Die Verwendung der Flussbegrenzer bringt Nichtlinearitäten in das zu lösende System hinein. Dies macht die Lösung mittels eines Verfahrens für nichtlineare Gleichungssysteme, z.B. des Newton-Verfahrens, notwendig. Im Artikel wird u. a. über “Since the flux limiter is nonlinear the solutions were obtained using full Newton iteration” [ZFV98, S. 59] darauf hingewiesen.



## 2.2. Amerikanische Optionen als Lineares Programm

Der Ansatz der Arbeitsgruppe um Dempster führt darauf, das lineare Komplementaritätsproblem, Bewertung einer Option, in ein Lineares Programm zu überführen. In einem weiteren Schritt wird dieses direkt gelöst. Bei diesem Vorgehen ist herauszustellen, dass für die numerische Übertragbarkeit zusätzliche Bedingungen an die Diskretisierung gestellt werden müssen.

### 2.2.1. Analytische Übertragbarkeit

Die Arbeit von Dempster und Hutton [DH99] beinhaltet im Wesentlichen die Äquivalenz- und Eindeutigkeitsaussage für das lineare Komplementaritätsproblem, welches der Aufgabe der Bewertung einer amerikanischen Option entspricht.

#### Satz 2.2.1. Problemäquivalenz zum Linearen Programm

Für den Hilbertraum  $H^{1,2}(\mathbb{R}^2)$  mit der Bilinearform

$$\langle f, g \rangle := \int_{\mathbb{R}^2} f(x, y) g(x, y) e^{-\mu(|x|+|y|)} dx dy$$

und einen koerziven, zeitunabhängigen Typ (Z) Operator  $(-\mathcal{L}_{\log})$  sei die Bilinearform  $a(\cdot, \cdot)$  definiert durch  $a(f, g) := \langle g, -\mathcal{L}_{\log} f \rangle$ . Dann existiert eine eindeutige Lösung der folgenden äquivalenten Probleme:

$$\begin{aligned}
 (OCP) \quad & \begin{cases} v(\cdot, T) = \tilde{\Psi} \\ v \geq \tilde{\Psi} \\ -\mathcal{L}_{\log} v - \frac{\partial v}{\partial t} \geq 0 \\ \left(-\mathcal{L}_{\log} v - \frac{\partial v}{\partial t}\right) \wedge (v - \tilde{\Psi}) = 0, \end{cases} \\
 (VI) \quad & \begin{cases} v(\cdot, T) = \tilde{\Psi} \\ v \geq \tilde{\Psi} \\ a(v, \hat{v} - v) + \langle \hat{v} - v, -\frac{\partial v}{\partial t} \rangle \geq 0 \quad \forall \hat{v} \geq \tilde{\Psi}, \end{cases} \\
 (LE) \quad & \text{finde } v = LE(F), \\
 (LP) \quad & \min_{v \in F} \langle v, c \rangle, \quad c > 0 \text{ beliebig}
 \end{aligned}$$

$$\text{mit } F := \left\{ \hat{v} \in H^{1,2}(\mathbb{R}^2) : \hat{v}(\cdot, T) = \tilde{\Psi}, \hat{v} \geq \tilde{\Psi}, -\mathcal{L}_{\log} \hat{v} - \frac{\partial \hat{v}}{\partial t} \geq 0 \right\}.$$

#### Beweis:

Dieser Satz findet sich als Theorem 3.3 in besagter Arbeit. Für den Black-Scholes Fall wird der log-transformierte Operator  $\mathcal{L}_{\log} := \frac{\sigma^2}{2} \frac{\partial^2}{\partial x^2} + (r - \delta - \frac{\sigma^2}{2}) \frac{\partial}{\partial x} - r$  verwendet. Eine detaillierte Ausführung findet sich auch in meiner Diplomarbeit [IV01, Kapitel 4].  $\square$

## 2.2.2. Numerische Übertragbarkeit

Auch dieser Punkt ist von Dempster und Hutton betrachtet worden. Es muss nun sichergestellt werden, dass für eine Diskretisierung des Problems die Voraussetzungen aus dem analytischen Fall erhalten bleiben. Zum anderen wird nach der Konvergenz der diskreten Lösung gegen die kontinuierliche Lösung gefragt.

### Satz 2.2.2. Konvergenzaussage [JLL90]

Es seien  $r$ ,  $\delta$  und  $\sigma$  beschränkte  $C^1$ -Funktionen. Die Ableitungen dieser Funktionen seien ebenfalls beschränkt. Für die Auszahlungsfunktion  $\phi$  existiere ein  $\mu > 0$ , so dass  $\phi(x) \exp(-\mu|x|)$  beschränkt ist. Dann gilt:

- i) Die kontinuierliche Lösung der Lokalisierung des Problems auf alle  $x$  aus  $\Omega_{x_{max}} := \{x \in \mathbb{R} \mid |x| \leq x_{max}\}$  konvergiert für  $x_{max} \rightarrow \infty$  gegen die globale Lösung.
- ii) Es sei  $\Theta \in [0, 1]$  der Gewichtungparameter zwischen den Zeitschritten (0 explizit; 1 implizit). Falls  $\Delta x$ ,  $\Delta \tau$  gegen Null gehen sowie für  $\Theta < 1$  zusätzlich  $\frac{\Delta \tau}{\Delta x^2}$  gegen Null geht, konvergiert die Lösung der Diskretisierung auf  $\Omega_{x_{max}}$  gegen die Lösung der Lokalisierung.

### Beweis:

Beide Aussagen befinden sich mit entsprechendem Beweis im zitierten Artikel. Die Aussage i) als Proposition 4.1 und die Aussage ii) als Theorem 4.3 .  $\square$

Die Übertragbarkeit des Satzes 2.2.1 auf das diskretisierte Problem sichert folgendes

### Lemma 2.2.3. Diskrete Äquivalenz zum LP [DH99]

Die Diskretisierungsmatrix  $A$  des Problems (OCP) ist genau dann koerziv und vom Typ (Z), wenn für die Diskretisierung der Black-Scholes PDE

- i) mittels zentraler Differenzen

$$\frac{1}{\Delta x} \geq \frac{|r - \delta - \frac{\sigma^2}{2}|}{\sigma^2} = \frac{1}{2}|q_\delta - 1| \quad (2.17)$$

gilt

- ii) und bei Verwendung des Upwind-Schemas

$$\frac{1}{\Delta x} \geq \frac{|r - \delta - \frac{\sigma^2}{2}|}{\frac{\sigma^2}{2}} = |q_\delta - 1| \quad (2.18)$$

erfüllt ist.

### Beweis:

Für die Typ (Z) Eigenschaft der Matrix  $A = (a_{i,j})$  reicht es zu zeigen, dass  $a_{i,j} \leq 0$  für  $i \neq j$ . Nach einem Ergebnis von Jaillet et al. [JLL90] reicht diese Bedingung auch, um die Koerzitivität der Matrix nachzuweisen.  $\square$

**Bemerkung 2.2.4. Typ (Z) Bedingung entspricht Péclet-Bedingung**

Die Bedingung (2.17) für die Typ (Z) Eigenschaft der Diskretisierungsmatrix entspricht genau der Péclet-Bedingung (2.15a) für die TVD Eigenschaft.

Damit wird die Möglichkeit, eine numerische Lösung zu finden, an die Parameter des Problems geknüpft. Die Maschenweite muss entsprechend eingestellt werden.

**2.2.3. Lösungsalgorithmus**

Die erste Arbeit von Dempster und Hutton [DH99] geht vor allem auf die oben beschriebene Problemäquivalenz ein, die numerische Lösung des Problems wird lediglich auf die Benutzung fertiger Optimierungsroutinen (in diesem Fall der ‚IBM<sup>®</sup> Optimization Subroutine Library‘) zurückgeführt. Jedoch bietet die schon 1998 erschienene Arbeit zusammen mit Richards [DHR98] eine weitere Möglichkeit, das Problem noch schneller zu lösen. Hier wird nicht mehr ein Standardverfahren zur Lösung des Linearen Programms eingesetzt. Statt dessen wurde die Problemstruktur analysiert und in einen direkten Löser umgesetzt. Dieser hat einen in der Anzahl der Kurs- und Zeitgitterpunkten linearen Aufwand. Die Ausführung des Simplex Algorithmus wird, durch geschickte Wahl der Basisvariablen, auf das sukzessive Lösen von linearen Gleichungssystemen zurückgeführt. Diese Gleichungssysteme sind zeitschichtweise sogar noch so voneinander abhängig, dass sie über eine Art UL-Zerlegung<sup>1</sup> berechnet werden können. Eine detaillierte Beschreibung des Algorithmus findet sich in Kapitel 4.1.

**2.3. Kombinierung der beiden Ansätze**

Die Vorstellung einer sinnvollen Lösung des Problems besteht nun aus den Forderungen, dass die Lösung möglichst gut und möglichst einfach berechnet werden kann. Die Einfachheit des Lösungsverfahrens lässt sich weiter unterteilen in die Forderungen nach einem geringen Aufwand und gleichzeitig einer fehlenden Abhängigkeit von anderen Systemen (z.B. Software). Dies ist beides für das von Dempster, Hutton und Richards vorgestellte Verfahren erfüllt, insbesondere ist die Abhängigkeit von kommerziellen Programmbibliotheken wie der ‚IBM<sup>®</sup> Optimization Subroutine Library‘ oder ‚CPLEX<sup>®</sup>‘ aufgehoben worden. Ebenso versteht man unter einer sinnvollen Lösung, dass diese den Forderungen der Problemstellung entsprechen soll und eine hinreichende Genauigkeit aufweist. Insbesondere wird in die Aufgabenstellung nun aber die Ermittlung der Hedgeparameter Delta und Gamma (evtl. implizit) eingeschlossen. Aus der Konvexität der Option im Basiswert (vgl. hierzu die Aussage von Merton in Lemma 8.1.1) folgt die Monotonie des  $\Delta$ , der ersten Ableitung nach dem Kurs. Folglich muss auch eine oszillationsfreie Lösung gefordert werden. Somit macht es Sinn, sich in diesem Zusammenhang noch einmal die Resultate von Zvan et al. anzusehen.

<sup>1</sup>Die Bezeichnung UL-Zerlegung ist korrekt. Sie unterscheidet sich von der gebräuchlichen LU-Zerlegung dadurch, dass die Zerlegung in der rechten unteren Ecke der Matrix beginnt.

Der Versuch, die beiden Ergebnisse zumindest teilweise zu kombinieren, liefert das folgende

**Korollar 2.3.1. TVD und Typ (Z) Diskretisierung von Black-Scholes**

Eine Diskretisierung, die sowohl TVD ist als auch eine Diskretisierungsmatrix vom Typ (Z) erzeugt, muss die folgenden strengeren Bedingungen erfüllen:

i) Eine Diskretisierung mittels zentraler Finiter-Differenzen muss

$$\frac{1}{\Delta x} \geq \frac{|r - \delta - \frac{\sigma^2}{2}|}{\sigma^2} = \frac{1}{2}|q_\delta - 1|, \quad (2.19a)$$

$$\frac{1}{(1 - \Theta)\Delta\tau} \geq \frac{\sigma^2}{2\Delta x^2} + r \stackrel{(2.19a)}{\geq} \frac{\sigma^2}{8}(q_\delta - 1)^2 + r \quad (2.19b)$$

erfüllen.

ii) Für die Diskretisierung mittels des Upwind-Schemas müssen

$$\frac{1}{\Delta x} \geq |q_\delta - 1|, \quad (2.20a)$$

$$\frac{1}{(1 - \Theta)\Delta\tau} \geq \frac{\sigma^2}{2\Delta x^2} + \frac{|r - \delta - \frac{\sigma^2}{2}|}{\Delta x} + r \stackrel{(2.20a)}{\geq} \sigma^2(q_\delta - 1)^2 + r \quad (2.20b)$$

erfüllt sein.

**Beweis:**

- i) Die Bedingung (2.19a) gilt nach Lemma 2.1.8 und Lemma 2.2.3 sowohl für die TVD als auch die Typ (Z) Diskretisierung (vgl. auch Bemerkung 2.2.4). Die Gleichung (2.19b) ergibt sich aus der mit der ersten Gleichung verknüpften TVD-Bedingung an das  $\Delta\tau$ .
- ii) Die Gleichung (2.20a) ist die entsprechende Typ (Z) Bedingung des Lemma 2.2.3. Die zweite Gleichung entsteht aus der TVD Bedingung für diese Diskretisierung (Lemma 2.1.8) und führt zusammen mit der Bedingung an  $\Delta x$  der ersten Gleichung zu einer Verschärfung der Forderungen an  $\Delta\tau$ .  $\square$

**Bemerkung 2.3.2. Abhängigkeit der Maschenweiten von der Péclet-Zahl**

Durch die Zusammenführung beider Ansätze werden beide Maschenweiten  $\Delta x$  und  $\Delta\tau$  von der Péclet-Zahl abhängig. Insbesondere wird auch für das Upwind-Schema wieder eine Bedingung an die Maschenweite  $\Delta x$  gestellt. In beiden Fällen verhindert die notwendig gewordene Feinheit der Diskretisierungen eine schnelle Berechnung der Lösung des Problems.

Wie schon in Lemma 2.1.6 erwähnt, ist die interessante Eigenschaft des Limiter, den numerischen Fluss so zu begrenzen, dass das resultierende Verfahren TVD Verhalten hat, mit der Notwendigkeit einer nichtlinearen Diskretisierungsgleichung verbunden. Damit ergibt sich

**Bemerkung 2.3.3. *Limiter sind mit dem LP-Löser unvereinbar***

*i) Nach der Überführung des Komplementaritätsproblems in ein Lineares Programm wird die Möglichkeit, ein Limiter-Verfahren zu verwenden, ausgeschlossen. Das LP-Lösungsverfahren, wie es von Dempster et al. vorgeschlagen wurde, erfordert zwingend lineare Gleichungen.*

*ii) Auch der alternative Ansatz über weighted essentially non-oscillatory (WENO) Schemata, wie er von Oosterlee et al. [OFG04] in diesem Zusammenhang vorgeschlagen wird, ist für die Verwendung des LP-Lösers wegen der ebenfalls nichtlinearen Gleichungen nicht geeignet.*

*Ein Differenzen-Schema gilt dabei als wesentlich oszillationsfrei, falls für ein  $p \geq 1$  gilt:*

$$TV(u^{(m+1)}) \leq TV(u^{(m)}) + \mathcal{O}(\Delta x^p). \quad (2.21)$$

Wir schließen dieses Kapitel mit der

**Beobachtung 2.3.4. *Gutartigkeit des Problems für eine kleine Péclet-Zahl***

*Auffällig ist in Korollar 2.3.1 die Gutartigkeit des Problems für kleine Werte der Péclet-Zahl  $PE = q_\delta - 1$ . Diese ist nun wiederum klein, falls der Konvektionsterm in der log-transformierten Black-Scholes Gleichung (1.8) klein ist. Wir müssen also die numerische Abhängigkeit der Lösung einer diskreten Version dieser Gleichung von der Größe des Konvektionsterms untersuchen.*



# 3. Verwenden einer geeigneten Transformation

Das bereits in Kapitel 2 beschriebene und mit Abbildung 2.1 veranschaulichte Oszillieren der Lösungsfunktion wird in der Literatur vielfach über das Auftreten von (numerischer) Dispersion und Dissipation erklärt. Wir wollen eine eingehende Untersuchung dieses Phänomens an der Black-Scholes PDE zuerst analytisch und dann für die Diskretisierung durchführen. Ein Teil hiervon ist bereits in [IV05] veröffentlicht worden.

## 3.1. Dispersion und Dissipation - Eine Analyse

Bevor wir in die Analyse einsteigen werden zunächst die Begriffe Dispersion und Dissipation präzisiert. Wir folgen dabei dem Vorgehen des Buches von Thomas [Tho95].

### 3.1.1. Einführung

Es liege eine PDE vor und die Lösung  $v$  kann als abstrakte Fourier-Reihe angegeben werden. Sei

$$v(x, t) = \sum_{n=-\infty}^{\infty} \hat{v}_n e^{i\omega_n t} e^{i\beta_n x} \tag{3.1}$$

diese Lösung, wobei mit  $\omega_n$  die Frequenz und mit  $\beta_n$  die Wellennummer gegeben sei. Die Wellenlänge ergibt sich als  $2\pi/\beta_n$ . Betrachtet man nur einen einzigen Summationsterm

$$\tilde{v}(x, t) = \hat{v} e^{i\omega t} e^{i\beta x} \tag{3.2}$$

und lässt den Index  $n$  für  $\omega$  bzw.  $\beta$  fallen, so kann man folgende Definitionen aufstellen:

**Definition 3.1.1. Dissipation**

*Die Lösung einer PDE heißt dissipativ, falls keine der Fourier-Moden verstärkt und mindestens eine Mode abgeschwächt wird.*

**Definition 3.1.2. Dispersion**

*Die Lösung einer PDE heißt dispersiv, falls sich die Fourier-Moden für verschiedene Wellennummern mit unterschiedlicher Geschwindigkeit in der Zeit fortpflanzen.*

Anhand des folgenden Beispiels sollen die Eigenschaften erklärt und gleichzeitig eine später noch wichtige Bezeichnung eingeführt werden.

**Beispiel 3.1.3. Dissipations-Dispersions-Analyse**

i) Für die Wärmeleitungsgleichung

$$v_t - a v_{xx} = 0$$

ergibt sich durch Einsetzen der Fourier-Moden aus Gleichung (3.2):  $v(x, t)$  ist nur dann eine Lösung der Gleichung, falls die so genannte **Dispersions-Relation**  $i\omega = -a\beta^2$  erfüllt ist. Die Lösungsmoden haben nun die Form

$$\tilde{v}(x, t) = \hat{v} e^{-a\beta^2 t} e^{i\beta x} \tag{3.3}$$

und zeigen dissipatives Verhalten.

ii) Für die zweite Testgleichung

$$v_t + a v_{xxx} = 0$$

ergibt sich auf ähnliche Weise die Dispersions-Relation  $\omega = a\beta^3$  und die Lösungsmoden

$$\tilde{v}(x, t) = \hat{v} e^{ia\beta^3 t} e^{i\beta x}.$$

Die Wellenbewegung erfolgt mit der Geschwindigkeit  $a\beta^3$  und ist somit dispersiv.

Man spricht in beiden Fällen von einer Dispersions-Relation, obwohl der Begriff Dissipations-Dispersions-Relation sinnvoller wäre.

### 3.1.2. Dispersion und Dissipation für die Black-Scholes PDE

Wir wollen nun eine Analyse der Gleichung von Black und Scholes vornehmen. Dazu betrachten wir die verschiedenen Transformationen der Black-Scholes PDE aus Kapitel 1.3 und untersuchen das Verhalten einzelner Moden. Wir präzisieren damit die Aussagen aus [IV05].

**Lemma 3.1.4. Dispersion und Dissipation für Black-Scholes Transformationen**  
Bei allen verwendeten Transformationen betrachten wir Fourier-Moden der Form

$$\tilde{v}(x, \tau) = \hat{v} e^{i\omega\tau} e^{i\beta x}. \tag{3.4}$$



Eine Dispersions-Dissipations-Analyse liefert die Aussagen:

i) Für die log-transformierte Black-Scholes Gleichung (vgl. Gleichung (1.8))

$$v_\tau - \frac{\sigma^2}{2} v_{xx} - \left(r - \delta - \frac{\sigma^2}{2}\right) v_x + r v = 0 \quad (3.5)$$

ergibt sich die Dispersions-Relation

$$i\omega = -r - \frac{\sigma^2}{2}\beta^2 + \left(r - \delta - \frac{\sigma^2}{2}\right)i\beta \quad (3.6)$$

mit Lösungsmoden der folgenden Form

$$\tilde{v}(x, \tau) = \hat{v} \underbrace{\exp(-r\tau)}_{=:a_1} \underbrace{\exp\left(-\frac{\sigma^2}{2}\beta^2\tau\right)}_{=:a_2} \underbrace{\exp\left(i\beta\left(x + \left(r - \delta - \frac{\sigma^2}{2}\right)\tau\right)\right)}_{=:a_3}, \quad (3.7)$$

wobei  $a_1$  den Zerfall der Lösung, d.h. in diesem Fall die Diskontierung,  $a_2$  dissipatives Verhalten und  $a_3$  den Transport der Moden gemäß  $\left(r - \delta - \frac{\sigma^2}{2}\right)$  beschreibt.

ii) Für die Wilmott Transformation der Black-Scholes PDE auf die Wärmeleitungsgleichung erhalten wir eine Dispersions-Relation  $i\omega = -\beta^2$  und Lösungsmoden

$$\tilde{v}(x, \tau) = \hat{v} \exp\left(-\beta^2\tau\right) \exp\left(i\beta x\right). \quad (3.8)$$

iii) Für die beiden das Koordinatensystem um die Konvektion verschiebenden Transformationen erhalten wir eine Dispersions-Relation  $i\omega = -\frac{\sigma^2}{2}\beta^2$ . Mit

$$\tilde{v}(x, \tau) = \hat{v} \exp\left(-\frac{\sigma^2}{2}\beta^2\tau\right) \exp\left(i\beta x\right) \quad (3.9)$$

werden in diesem Fall die Lösungsmoden beschrieben.

iv) Für die neue Transformation ergeben sich Lösungsmoden der Form

$$\tilde{v}(x, \tau) = \hat{v} \exp\left(-\frac{\sigma^2}{8}\tau\right) \exp\left(-\frac{\sigma^2}{2}\beta^2\tau\right) \exp\left(i\beta x\right). \quad (3.10)$$

Die zugehörige Dispersions-Relation lautet  $i\omega = -\frac{\sigma^2}{2}\beta^2 - \frac{\sigma^2}{8}$ .

**Beweis:**

Nachrechnen analog zum Beispiel 3.1.3. □

Die Lösungsmoden der Gleichungen (3.7)-(3.10) sind alle unterschiedlich. Die Stärke der Dissipation ist einer dieser Unterschiede. Im Falle der Wilmott Transformation ist der Vorfaktor z.B.  $-\beta^2$  statt  $-\frac{\sigma^2}{2}\beta^2$ . Ebenfalls unterschiedlich ist das restliche Verhalten der Lösung. Hervorzuheben ist dabei der Transport von Lösungsmoden im Falle der log-Transformation, auf den im späteren Verlauf der Arbeit noch eingegangen wird. Mit geringerer Bedeutung ist der (unterschiedliche) Zerfall der Lösungen zu betrachten, der in Anbetracht der Problemstellung auch als Bildung des Gegenwartswertes oder Diskontierung angesehen werden kann. Die wesentlichen Gemeinsamkeiten werden über folgende Bemerkung festgehalten:

**Bemerkung 3.1.5.** *Black-Scholes ist dissipativ, aber nicht dispersiv.*

*Die Lösungsmoden der analytischen Lösung zeigen alle ein dissipatives Verhalten. Ebenfalls allen Transformationen gemeinsam ist das Fehlen eines dispersiven Anteils!*

## 3.2. Numerisches Vorgehen

Zur Berechnung einer numerischen Lösung dieser Black-Scholes verwandten PDEs wird eine Finite-Differenzen-Diskretisierung der Gleichung behandelt. Hier scheint von vorneherein klar zu sein, dass das Dispersions-Dissipations-Verhalten nun nicht mehr allein von der Transformation der Gleichung, sondern auch noch von dem verwendeten Differenzen-Schema abhängt. Insbesondere könnte es passieren, dass sich die diskreten Moden bei analytisch äquivalenten Gleichungen unterscheiden.

Eine diskrete Dissipations-Dispersions-Analyse wird ebenso geführt wie ihr kontinuierliches Analogon. Man betrachtet Fourier-Moden der Form

$$\tilde{u}_k^{(m)} = \hat{u} e^{i\omega m \Delta \tau} e^{i\beta k \Delta x} \quad (3.11)$$

wobei die Indizes  $k$  und  $m$  auf den  $k$ -ten Knoten im Zustandsraum (der Begriff ‚Kurs‘ passt hier aufgrund der Transformation nicht mehr) sowie in der  $m$ -ten Zeitschicht hinweisen. Man beachte, dass diese Gleichung die genaue Umsetzung von Gleichung (3.2) in den diskreten Fall ist. Gesucht ist nun wieder eine Relation der Form  $\omega = f(\beta)$ .

### 3.2.1. Log-transformierte Black-Scholes PDE

Die log-transformierte Black-Scholes Gleichung (1.8) wird, wie schon in [IV05], mittels des folgenden Differenzen-Schemas auf einem noch nicht näher festgelegten äquidistanten Gitter gelöst. Hierbei wird der Einfluss von sowohl Anfangs- als auch Randbedingungen außer Acht gelassen und auch die genauen Maschenweiten werden nicht näher beschrieben. Es werden zuerst zentrale Finite-Differenzen mit einem Zeitparameter  $\Theta = 0, 1, \frac{1}{2}$  für den Wechsel zwischen explizitem Euler, implizitem Euler und Crank-Nicolson-Verfahren

verwendet. Dabei gilt:

$$v \approx \Theta u_k^{(m+1)} + (1 - \Theta) u_k^{(m)}, \quad (3.12)$$

$$v_\tau \approx \frac{u_k^{(m+1)} - u_k^{(m)}}{\Delta\tau}, \quad (3.13)$$

$$v_x \approx \Theta \frac{u_{k+1}^{(m+1)} - u_{k-1}^{(m+1)}}{2\Delta x} + (1 - \Theta) \frac{u_{k+1}^{(m)} - u_{k-1}^{(m)}}{2\Delta x}, \quad (3.14)$$

$$v_{xx} \approx \Theta \frac{u_{k+1}^{(m+1)} - 2u_k^{(m+1)} + u_{k-1}^{(m+1)}}{(\Delta x)^2} + (1 - \Theta) \frac{u_{k+1}^{(m)} - 2u_k^{(m)} + u_{k-1}^{(m)}}{(\Delta x)^2}. \quad (3.15)$$

Setzt man die Gleichungen (3.12)-(3.15) in die log-transformierte Black-Scholes Gleichung ein, so erhält man nach einer Umordnung nach den Zeitschichten:

$$\begin{aligned} & u_k^{(m+1)} - \Theta \left[ \frac{\sigma^2}{2} \frac{\Delta\tau}{(\Delta x)^2} (u_{k+1}^{(m+1)} - 2u_k^{(m+1)} + u_{k-1}^{(m+1)}) \right. \\ & \quad \left. + (r - \delta - \frac{\sigma^2}{2}) \frac{\Delta\tau}{2\Delta x} (u_{k+1}^{(m+1)} - u_{k-1}^{(m+1)}) - r\Delta\tau u_k^{(m+1)} \right] \\ = & u_k^{(m)} + (1 - \Theta) \left[ \frac{\sigma^2}{2} \frac{\Delta\tau}{(\Delta x)^2} (u_{k+1}^{(m)} - 2u_k^{(m)} + u_{k-1}^{(m)}) \right. \\ & \quad \left. + (r - \delta - \frac{\sigma^2}{2}) \frac{\Delta\tau}{2\Delta x} (u_{k+1}^{(m)} - u_{k-1}^{(m)}) - r\Delta\tau u_k^{(m)} \right]. \end{aligned} \quad (3.16)$$

Nach dem Einsetzen der diskreten Fourier Moden (3.11) in jedem Gitterpunkt ergibt sich, unter Verwendung der trigonometrischen Formeln  $e^{i\beta\Delta x} + e^{-i\beta\Delta x} = 2 \cos \beta\Delta x$  und  $e^{i\beta\Delta x} - e^{-i\beta\Delta x} = i2 \sin \beta\Delta x$ , nach abschließender Division durch  $e^{i(\omega m\Delta\tau + \beta k\Delta x)}$  aus der letzten Gleichung die Aussage

$$\begin{aligned} & e^{i\omega\Delta\tau} \left[ 1 - \Theta \left( \sigma^2 \frac{\Delta\tau}{(\Delta x)^2} (\cos(\beta\Delta x) - 1) - r\Delta\tau \right. \right. \\ & \quad \left. \left. + i2 \left( r - \delta - \frac{\sigma^2}{2} \right) \frac{\Delta\tau}{2\Delta x} \sin(\beta\Delta x) \right) \right] \\ = & 1 + (1 - \Theta) \left( \sigma^2 \frac{\Delta\tau}{(\Delta x)^2} (\cos(\beta\Delta x) - 1) - r\Delta\tau \right. \\ & \quad \left. + i2 \left( r - \delta - \frac{\sigma^2}{2} \right) \frac{\Delta\tau}{2\Delta x} \sin(\beta\Delta x) \right). \end{aligned} \quad (3.17)$$

Durch Festlegung der Abkürzungen  $G$  und  $H$  als

$$G := \sigma^2 \frac{\Delta\tau}{(\Delta x)^2} (\cos(\beta\Delta x) - 1) - r\Delta\tau,$$

$$H := 2 \left( r - \delta - \frac{\sigma^2}{2} \right) \frac{\Delta\tau}{2\Delta x} \sin(\beta\Delta x),$$

verkürzt sich diese Gleichung zu

$$e^{i\omega\Delta\tau} = \frac{1 + (1 - \Theta)(G + iH)}{1 - \Theta(G + iH)}. \quad (3.18)$$

Wir zerlegen nun die rechte Seite der Gleichung in Real- und Imaginärteil. Es folgt:

$$e^{i\omega\Delta\tau} = \frac{(1 + (1 - \Theta)G)(1 - \Theta G) - \Theta(1 - \Theta)H^2}{(1 - \Theta G)^2 + (\Theta H)^2} + i \frac{H}{(1 - \Theta G)^2 + (\Theta H)^2}. \quad (3.19)$$

Legt man ein komplexes  $\omega := \alpha + ib$  mit reellem  $\alpha$ ,  $b$  und damit auch  $e^{i\omega\Delta\tau} = e^{i\alpha\Delta\tau} e^{-b\Delta\tau}$  zugrunde, so sind die Lösungsmoden für

$$e^{-b\Delta\tau} = |e^{i\omega\Delta\tau}| \begin{cases} > 1 & \text{wachsend} \\ = 1 & \text{konstant} \\ < 1 & \text{fallend.} \end{cases} \quad (3.20)$$

Ebenso kann man den Term  $e^{i\alpha\Delta\tau}$  betrachten. Unter Verwendung der trigonometrischen Identitäten  $e^{i\alpha\Delta\tau} = \cos(\alpha\Delta\tau) + i \sin(\alpha\Delta\tau)$  und  $e^{i\alpha\Delta\tau} = \frac{e^{i\omega\Delta\tau}}{|e^{i\omega\Delta\tau}|}$  finden wir die Gleichung

$$\tan \alpha\Delta\tau = \frac{\Im(e^{i\omega\Delta\tau})}{\Re(e^{i\omega\Delta\tau})}, \quad (3.21)$$

welche für

$$\begin{aligned} \alpha &= \frac{1}{\Delta\tau} \arctan \left( \frac{\Im(e^{i\omega\Delta\tau})}{\Re(e^{i\omega\Delta\tau})} \right) \\ &= \begin{cases} 0 & \text{weder einen Modentransport noch Dispersion} \\ c \neq 0 & \text{einen Transport der Lösungsmoden} \\ f(\beta) & \text{dispersives Verhalten} \end{cases} \end{aligned} \quad (3.22)$$

beschreibt. Mit Hilfe der oben durchgeführten Zerlegung in Real- und Imaginärteil sieht man sofort, dass

$$\Im(e^{i\omega\Delta\tau}) = \frac{H}{(1 - \Theta G)^2 + (\Theta H)^2} \quad (3.23)$$

gilt. Es folgt

**Bemerkung 3.2.1. Die Diskretisierung ist dispersiv**

- i) Der Imaginärteil von  $e^{i\omega\Delta\tau}$  ist genau dann keine Funktion von  $\beta$ , wenn  $H = 0$  gilt.
- ii) Notwendig für einen Imaginärteil identisch Null ist wiederum  $r - \delta - \frac{\sigma^2}{2} = 0$ . Dies ist jedoch nur in Ausnahmefällen erfüllt. Das resultierende Verfahren ist im Allgemeinen dispersiv.
- iii) Die Bedingung  $r - \delta - \frac{\sigma^2}{2} = 0$  passt zu der Beobachtung 2.3.4. Falls der Betrag von  $r - \delta - \frac{\sigma^2}{2}$  klein ist, wird die Péclet-Zahl ebenfalls klein.

Wenden wir uns nun der in [IV05] nicht durchgeführten Dissipations-Analyse und damit dem Betrag von  $e^{i\omega\Delta\tau}$  zu, so gilt

$$|e^{i\omega\Delta\tau}| = \sqrt{\Re(e^{i\omega\Delta\tau})^2 + \Im(e^{i\omega\Delta\tau})^2} \quad (3.24)$$

und nach der Definition von  $b$  über die Gleichung (3.20) folgt:

$$b = -\frac{1}{2\Delta\tau} \ln \left( \Re(e^{i\omega\Delta\tau})^2 + \Im(e^{i\omega\Delta\tau})^2 \right). \quad (3.25)$$

Das nachfolgende Lemma liefert eine Aussage über die diskrete Dissipation:

**Lemma 3.2.2. Diskrete Dissipation**

Aus der Gleichung (3.25) kann man für die speziellen Fälle  $\Theta = 0, \frac{1}{2}, 1$  eine diskrete Dissipation

$$b \approx \begin{cases} -r - \frac{\sigma^2}{2}\beta^2 + \frac{1}{2\Delta\tau}(G^2 + H^2) & \text{für } \Theta = 0 \\ -r - \frac{\sigma^2}{2}\beta^2 - \frac{1}{2\Delta\tau}G(4G - G^2 - H^2) & \text{für } \Theta = \frac{1}{2} \\ -r - \frac{\sigma^2}{2}\beta^2 - \frac{1}{2\Delta\tau}(G^2 + H^2) & \text{für } \Theta = 1 \end{cases} \quad (3.26)$$

herleiten. In allen drei Fällen wird sowohl der Lösungszerfall als auch die analytische Dissipation nur unzureichend weitergegeben.

**Beweis:**

Unter Verwendung der Näherung  $\cos(\beta\Delta x) - 1 \doteq \frac{-\beta^2\Delta x^2}{2}$  gelten die Gleichungen

i) Für  $\Theta = 0$ :

$$\begin{aligned} b &= -\frac{1}{2\Delta\tau} \ln \left( (1 + G)^2 + H^2 \right) \\ &\stackrel{\text{Taylor}}{\approx} \frac{1}{2\Delta\tau} (2G + G^2 + H^2) \\ &\approx -r - \frac{\sigma^2}{2}\beta^2 + \frac{1}{2\Delta\tau}(G^2 + H^2). \end{aligned} \quad (3.27)$$

ii) Für  $\Theta = \frac{1}{2}$ :

$$\begin{aligned} b &= -\frac{1}{2\Delta\tau} \ln \left( \frac{\left( \left(1 + \frac{G}{2}\right)\left(1 - \frac{G}{2}\right) - \frac{H^2}{4} \right)^2 + H^2}{\left( \left(1 - \frac{G}{2}\right)^2 + \frac{H^2}{4} \right)^2} \right) \\ &\stackrel{\text{Taylor}}{\approx} \frac{1}{2\Delta\tau} \left( \left( \left(1 + \frac{G}{2}\right)\left(1 - \frac{G}{2}\right) - \frac{H^2}{4} \right)^2 + H^2 - \left( \left(1 - \frac{G}{2}\right)^2 + \frac{H^2}{4} \right)^2 \right) \\ &\approx -r - \frac{\sigma^2}{2}\beta^2 - \frac{1}{2\Delta\tau}G(4G - G^2 - H^2). \end{aligned} \quad (3.28)$$

iii) Für  $\Theta = 1$ :

$$\begin{aligned}
 b &= \frac{1}{2\Delta\tau} \ln \left( (1-G)^2 + H^2 \right) \\
 &\stackrel{\text{Taylor}}{\approx} \frac{1}{2\Delta\tau} (2G - G^2 - H^2) \\
 &\approx -r - \frac{\sigma^2}{2} \beta^2 - \frac{1}{2\Delta\tau} (G^2 + H^2).
 \end{aligned} \tag{3.29}$$

□

Für den, im Hinblick auf die numerische Dispersion, interessanten Fall eines Imaginärteils identisch gleich Null ergibt sich

**Bemerkung 3.2.3. Konvektion identisch Null liefert ‚echte‘ Dissipation**

Betrachten wir jedoch den wegen Bemerkung 3.2.1 interessanten Fall  $H = 0$ . Dann ist der Imaginärteil identisch zu Null und aus

$$b = -\frac{1}{\Delta\tau} \ln \left( \Re(e^{i\omega\Delta\tau}) \right) \tag{3.30}$$

folgt in allen drei Fällen

$$b \approx \frac{1}{\Delta\tau} G \approx -r - \frac{\sigma^2}{2} \beta^2. \tag{3.31}$$

Das Verhalten der Gleichung wird in diesem Falle qualitativ richtig wiedergegeben.

### 3.2.2. Problemanalyse für das Upwind-Schema

In Erinnerung an das Lemma 2.1.6 versuchen wir die Gleichung (1.8) mit dem Upwind-Schema zu diskretisieren. Das Verfahren gibt in jedem Falle die Richtung der Konvektion wieder, jedoch bringt die Dissipations-Dispersions-Analyse kein erfreuliches Ergebnis. Statt  $i2 \sin(\beta\Delta x)$  treten jetzt Terme der Form  $\pm(i \sin(\beta\Delta x) + \cos(\beta\Delta x) - 1)$  auf und wir erhalten

$$G_{\text{upwind}} := \left( \sigma^2 \frac{\Delta\tau}{(\Delta x)^2} - \text{sign} \left( r - \delta - \frac{\sigma^2}{2} \right) \left| r - \delta - \frac{\sigma^2}{2} \right| \frac{\Delta\tau}{2\Delta x} \right) (\cos(\beta\Delta x) - 1) - r\Delta\tau,$$

$$H_{\text{upwind}} := -\text{sign} \left( r - \delta - \frac{\sigma^2}{2} \right) \left| r - \delta - \frac{\sigma^2}{2} \right| \frac{\Delta\tau}{2\Delta x} \sin(\beta\Delta x).$$

Auch hier bleibt ein Imaginärteil ungleich Null übrig. Insbesondere wird damit das Auftreten numerischer Dispersion **nicht** verhindert.

Rufen wir uns an dieser Stelle die Form von  $H$  für die zentralen Finiten-Differenzen in Erinnerung,

$$H_{\text{zentrale FD}} := 2 \left( r - \delta - \frac{\sigma^2}{2} \right) \frac{\Delta\tau}{2\Delta x} \sin(\beta\Delta x),$$

und denken daran, dass die numerische Dispersion für  $H = 0$  komplett verschwindet, so können wir folgende Beobachtung formulieren:

**Beobachtung 3.2.4. Konvektionsterm verursacht numerische Dispersion**

Die numerische Dispersion wird durch die Diskretisierung des Konvektionsterms der PDE in die zugehörige Differenzgleichung hineingebracht.

### 3.3. Suche nach einer geeigneten Transformation

Wir haben festgestellt, dass bei der Diskretisierung der log-transformierten Black-Scholes Gleichung, unabhängig von der Wahl des Differenzen-Schemas, numerische Dispersion auftreten kann. Als Ursache dafür haben wir den Konvektionsterm der Gleichung ausgemacht. Da verschiedene Diskretisierungen den Konvektionsterm nicht hinreichend gut approximieren wollen wir nun versuchen, diesen über eine geeignete Transformation zu eliminieren.

#### 3.3.1. Wilmotts Transformation auf die Wärmeleitungsgleichung

Die am weitesten verbreitete Transformation ohne Konvektionsterm ist die von Wilmott Eingeführte. Wir erinnern uns an die resultierende Gleichung

$$v_\tau - v_{xx} = 0.$$

Die diskrete Dissipations-Dispersions-Analyse liefert uns die Relation:

$$e^{i\omega\Delta\tau} = \frac{1 + (1 - \Theta)\frac{2\Delta\tau}{(\Delta x)^2}(\cos(\beta\Delta x) - 1)}{1 - \Theta\frac{2\Delta\tau}{(\Delta x)^2}(\cos(\beta\Delta x) - 1)}. \quad (3.32)$$

Erstens fehlt hier der Imaginärteil, d.h. es tritt keine numerische Dispersion auf, und zweitens wird die kontinuierliche Dissipation hinreichend gut genähert. Damit ist ein aussichtsreicher Kandidat für die Transformation gefunden worden.

**Beispiel 3.3.1. Zvans Testproblem [ZFV98]**

Man berechne den Preis eines europäischen Vanilla Calls mit  $K = 15.0$ ,  $T = 1.0$ ,  $r = 0.15$ ,  $\delta = 0$  und  $\sigma = 0.01$ .

Versuchen wir jedoch dieses Testproblem in der transformierten Fassung zu lösen, so stoßen wir, wie bereits in [IV05] angemerkt, auf weitere Schwierigkeiten.

**Beobachtung 3.3.2. Die Anfangsbedingungen sind nicht auszuwerten.**

Beim Versuch die Anfangsbedingung bzw. Dirichlet-Randbedingungen für  $x_{max}$ ,  $x_{min}$  auszuwerten wird der numerische Algorithmus unabhängig von der Diskretisierung abbrechen.

Denn für die transformierte Auszahlungsfunktion des Calls gilt:

$$\begin{aligned} \phi(x, \tau) &= \exp\left(\frac{x}{2}(q_\delta - 1) + \frac{\tau}{4}((q_\delta - 1)^2 + 4q)\right)(e^x - 1) \\ &= \exp\left(\frac{\tau}{4}((q_\delta - 1)^2 + 4q)\right)\left(e^{\frac{x}{2}(q_\delta + 1)} - e^{\frac{x}{2}(q_\delta - 1)}\right) \\ &= \exp\left(\frac{\tau}{4}((q_\delta - 1)^2 + 4q)\right) \exp\left(\frac{x}{2}q_\delta\right) 2 \sinh\left(\frac{x}{2}\right). \end{aligned}$$

Hier lässt sich der erste Term mit  $q_\delta = q - \frac{2\delta}{\sigma^2}$  wie folgt abschätzen:

$$\exp\left(\frac{\tau}{4}((q_\delta - 1)^2 + 4q)\right) \geq \exp\left(\frac{\tau}{4}((q_\delta - 1)^2 + 4q_\delta)\right) = \exp\left(\frac{\tau}{4}(q_\delta + 1)^2\right). \quad (3.33)$$

Für die Wahl eines rechten Randwertes  $x_{max} \geq 2$  ist die Abschätzung

$$e^{\frac{x_{max}}{2}(q_\delta + 1)} > \exp(q_\delta) \quad (3.34)$$

erfüllt. Damit erhält man für  $\tau \gg 0$  oder für hinreichend großes  $x_{max}$  eine Abschätzung gegen  $\exp(q_\delta) = \exp(Pe + 1)$ . Hier spielt wieder die Péclet-Zahl eine Rolle und für das Beispiel 3.3.1 gilt  $Pe = 3000$ . Beim Versuch der numerischen Auswertung der Auszahlungsfunktion kommt es daher zu einem Exponentenüberlauf.

Auch wenn das Verfahren nun theoretisch eine dispersionsfreie Lösung liefert, so kann es in der Praxis nicht angewandt werden, da es unmöglich ist, die Iteration zu starten. Eine weitere Einschränkung besteht auch noch darin, dass die Wilmott Transformation nur für konstantes  $\sigma$  durchzuführen ist. Denn die neue Zeit  $\tau$  ist auch über den Parameter  $\sigma$  definiert worden. Die Zeit kann aber nicht unterschiedlich schnell ablaufen.

### 3.3.2. Konvektions-Verschiebung

Wir betrachten die Transformation von Barraquand und Pudet für konstante Parameter, d.h. den klassischen Black-Scholes Fall bzw. die Transformation von Smith für zeitabhängige Parameter, d.h. das von Merton erweiterte Modell. In beiden Fällen resultiert die Gleichung

$$v_\tau - \frac{\sigma(T - \tau)^2}{2} v_{xx} = 0$$

aus der Transformation. Diese Gleichung ist fast identisch zur log-Transformation für den Null Konvektion Fall. Wir haben nur zusätzlich den Diskontierungsanteil wegtransformiert. Die Dispersions-Dissipations-Analyse für die Differenzengleichung ergibt dann erwartungsgemäß die Relation

$$e^{i\omega\Delta\tau} = \frac{1 + (1 - \Theta)G}{1 - \Theta G}, \quad (3.35)$$

mit der bekannten Abkürzung  $G$ , die um den Anteil  $r\Delta\tau$  bereinigt, nun die Form

$$G := \sigma^2 \frac{\Delta\tau}{(\Delta x)^2} (\cos(\beta\Delta x) - 1)$$

hat. Die Auszahlungsfunktion für den Call genügt der Abschätzung

$$\begin{aligned} \phi(x, \tau) &= K \max \left[ \exp \left( x + \int_0^\tau \delta(T - y) + \frac{(\sigma(T - y))^2}{2} dy \right) \right. \\ &\quad \left. - \exp \left( - \int_0^\tau r(T - y) dy \right), 0 \right] \\ &\leq K \exp \left( x + \tau \left( \delta_{max} + \frac{\sigma_{max}^2}{2} \right) \right), \end{aligned} \quad (3.36)$$



wobei  $\delta_{max}$  und  $\sigma_{max}$  die jeweiligen Maxima im Intervall  $[0, \tau]$  repräsentieren. Damit ist die Auszahlungsfunktion für vernünftige Werte von  $x$  und  $\tau$  auswertbar, insbesondere wachsen ihre Werte nicht mit der Péclet-Zahl. Zusammenfassend erfüllen also beide Transformationen die Anforderung, keine numerische Dispersion zu verursachen und sind gleichzeitig so aufgebaut, dass eine numerische Auswertung der Auszahlungsfunktion auch für eine große Péclet-Zahl noch möglich ist.

Der einzig verbleibende Kritikpunkt in diesem Fall ist die Anwendbarkeit auf das Modell von Cox und Ross. In diesem Modell ist die Volatilität  $\sigma$  zusätzlich zur Zeit auch vom Kurs  $S$  abhängig. Damit sind die beiden vorangehenden Transformationen nicht mehr anwendbar. Durch die log-Transformation erhält der Konvektionsterm die Form

$$r(T - \tau) - \delta(T - \tau) - \sigma(K \exp(x), T - \tau).$$

Für das totale Differential im Transformationssatz sind dann die Ableitungen des Integrals nach  $S$  bzw.  $x$  nicht mehr Null. Daher ist es zwingend notwendig, eine weitere Transformation zu entwickeln.

### 3.3.3. Eine neue Transformation

Wir versuchen nun mit den Ideen der vorangehenden Transformationen die Verwendbarkeit für lokale Volatilitätsflächen zu ermöglichen. Die Smith Transformation scheitert genau an dieser Abhängigkeit der Volatilität  $\sigma$  vom lokalen Aktienkurs  $S$ . Es ist daher naheliegend, die Volatilität unberücksichtigt zu lassen. Die in dieser Form gemachte Einschränkung

$$\begin{aligned} \tau &:= T - t, \\ x &:= \ln\left(\frac{S}{K}\right) + \int_0^\tau r(T - y) - \delta(T - y) dy, \end{aligned} \tag{3.37}$$

führt auf die vorläufige Gleichung

$$\widehat{V}_\tau - \frac{\sigma(x, \tau)^2}{2} [\widehat{V}_{xx} - \widehat{V}_x] + r(T - \tau) \widehat{V} = 0. \tag{3.38}$$

Hier taucht allerdings immer noch ein Konvektionsterm auf, so dass man in der Lösung weiterhin numerische Dispersion erwarten muss. Jedoch kann man sich der Idee von Wilmott bemächtigen und den Ansatz

$$\widehat{V}(x, \tau) =: \exp(g(x) + h(\tau)) v(x, \tau) \tag{3.39}$$

machen. Dann gilt:

$$\begin{aligned} \widehat{V}_\tau &= \exp(g(x) + h(\tau)) (h_\tau v + v_\tau), \\ \widehat{V}_x &= \exp(g(x) + h(\tau)) (g_x v + v_x), \\ \widehat{V}_{xx} &= \exp(g(x) + h(\tau)) ((g_x)^2 v + 2g_x v_x + g_{xx} v + v_{xx}). \end{aligned}$$

Die Gleichung (3.38) wird zu

$$\begin{aligned}
 0 &= e^{(g(x)+h(\tau))} \left[ (h_\tau v + v_\tau) + r(T - \tau) v \right. \\
 &\quad \left. - \frac{\sigma(x, T - \tau)^2}{2} \left( (g_x)^2 v + 2g_x v_x + g_{xx} v + v_{xx} \right) - (g_x v + v_x) \right] \\
 &= e^{(g(x)+h(\tau))} \left[ v_\tau + \left( h_\tau + r(T - \tau) + \frac{\sigma(x, T - \tau)^2}{2} (g_x - (g_x)^2 - g_{xx}) \right) v \right. \\
 &\quad \left. + \frac{\sigma(x, T - \tau)^2}{2} (1 - 2g_x) v_x - \frac{\sigma(x, T - \tau)^2}{2} v_{xx} \right]. \tag{3.40}
 \end{aligned}$$

Dabei ist es weiterhin das Ziel, den Konvektionsterm  $v_x$  zu eliminieren. Dazu setzen wir den führenden Koeffizienten gleich Null. Somit fordern wir

$$1 - 2g_x = 0 \quad \implies \quad g(x) = \frac{1}{2} x + C_1. \tag{3.41}$$

Die Gleichung (3.40) vereinfacht sich dann zu

$$0 = e^{(\frac{1}{2}x + C_1 + h(\tau))} \left[ v_\tau + \left( h_\tau + r(T - \tau) + \frac{\sigma(x, T - \tau)^2}{2} \left( \frac{1}{2} - \frac{1}{4} \right) \right) v - \frac{\sigma(x, T - \tau)^2}{2} v_{xx} \right].$$

Zu einer weiteren Vereinfachung führt die Wahl von

$$h_\tau + r(T - \tau) = 0 \quad \implies \quad h(\tau) = - \int_0^\tau r(T - y) dy + C_2 \tag{3.42}$$

mit zugehöriger Gleichung

$$0 = \exp \left( \frac{1}{2} x - \int_0^\tau r(T - y) dy + C \right) \left[ v_\tau - \frac{\sigma(x, T - \tau)^2}{2} (v_{xx} - \frac{1}{4} v) \right].$$

Es resultiert schließlich die äquivalente Bewertungsgleichung

$$0 = v_\tau - \frac{\sigma(x, T - \tau)^2}{2} (v_{xx} - \frac{1}{4} v). \tag{3.43}$$

Zusammenfassend hat diese, in [IV05] ohne Herleitung angeführte, Transformation für die Wahl der Integrationskonstanten  $C_1 = C_2 = 0$  die Form

$$\widehat{V}(x, \tau) =: \exp \left( \frac{1}{2} x - \int_0^\tau r(T - y) dy \right) v(x, \tau). \tag{3.44}$$

Die Auszahlungsfunktionen ergeben sich mit  $S = K \exp \left( x - \int_0^\tau r(T - y) - \delta(T - y) dy \right)$  im Falle des Calls zu

$$\begin{aligned}
 \phi(x, \tau) &= K \max \left[ \exp \left( + \frac{x}{2} + \int_0^\tau \delta(T - y) dy \right) - \exp \left( - \frac{x}{2} + \int_0^\tau r(T - y) dy \right), 0 \right] \\
 &\leq K \exp \left( \frac{x}{2} + \delta_{max} \right).
 \end{aligned}$$

Die Auszahlungsfunktion ist somit ebenfalls für vernünftige Werte von  $x$  und  $\tau$  auswertbar. Für einen Vanilla Put lautet der Payoff:

$$\phi(x, \tau) = K \max \left[ \exp \left( -\frac{x}{2} + \int_0^\tau r(T-y) dy \right) - \exp \left( \frac{x}{2} + \int_0^\tau \delta(T-y) dy \right), 0 \right].$$

Somit ist es möglich, eine numerische Berechnung der Optionswerte durchzuführen, ohne dass die resultierende Näherungslösung mit künstlicher Dispersion behaftet ist.

### 3.4. Eigenschaften der Transformationen

Die zuvor erarbeiteten Transformationen wollen wir nun auf die Eignung hinsichtlich der Zielvorgaben bezüglich Oszillationsfreiheit und LP-Darstellung aus Kapitel 2.3 untersuchen.

Da wir für die Wilmott Transformation bereits den Nachteil des Exponentenüberlaufs herausgestellt haben, untersuchen wir an dieser Stelle nur die Diskretisierung der Gleichung

$$v_\tau - \frac{\sigma^2}{2} v_{xx} = 0$$

aus der Konvektions-Verschiebung bzw. der Gleichung

$$v_\tau - \frac{\sigma(x, T - \tau)^2}{2} \left( v_{xx} - \frac{1}{4} v \right) = 0$$

nach der neuen Transformation. Hierbei gilt:

**Korollar 3.4.1. Bedingungen für die Diskretisierung**

*Für eine TVD Diskretisierung (vgl. Def. 2.1.4) der beiden vorangehenden Bewertungsgleichungen muss*

*i) für die Konvektions-Verschiebung nach Smith die Bedingung*

$$\frac{1}{(1 - \Theta)\Delta\tau} \geq \frac{\sigma^2}{\Delta x^2} \tag{3.45}$$

*ii) für die neue Transformation die Bedingung*

$$\frac{1}{(1 - \Theta)\Delta\tau} \geq \sigma^2 \left( \frac{1}{\Delta x^2} + \frac{1}{8} \right) \tag{3.46}$$

*erfüllt sein. Damit das Problem in die LP-Struktur überführt werden kann, sind keine weiteren Bedingungen notwendig.*

**Beweis:**

Die TVD Bedingungen ergeben sich für beide Transformationen aus dem Lemma 2.1.5. Nach Lemma 2.2.3 reicht es, für die Übertragbarkeit des Problems in ein Lineares Programm zu zeigen, dass die Nichtdiagonaleinträge der Diskretisierungsmatrix kleiner oder gleich Null sind. Dies führt in beiden Fällen auf die Bedingung

$$-\Theta \frac{\sigma^2}{2} \frac{\Delta\tau}{\Delta x^2} \leq 0, \tag{3.47}$$

die immer erfüllt ist. □

**Bemerkung 3.4.2.** *Abhängigkeit der Diskretisierung von Péclet-Zahl eliminiert*  
*In beiden Fällen ist die, in Bedingung (2.15a) noch vorhandene, Abhängigkeit von der Péclet-Zahl verschwunden.*

Auf dem Papier haben wir das numerische Vorgehen für die Berechnung mittels der Black-Scholes Gleichung wesentlich verbessert. Die starke Einschränkung durch die Péclet-Bedingungen ist wesentlich abgeschwächt worden. Ein weiterer nicht offensichtlicher Vorteil ist die Auslagerung der Zins- und Dividendenstruktur in die Auszahlungsfunktion. Insbesondere für zeitabhängige Parameter ist hier ein Austausch der Diskretisierungsmatrizen nicht mehr notwendig. Die Zeitabhängigkeit von  $\sigma$  bleibt von dieser Aussage unberührt.

## 4. Modifizierter Algorithmus

In diesem Kapitel soll die Auswirkung der Transformationen auf den eigentlichen Lösungsalgorithmus betrachtet werden. Bisher ist lediglich die Übertragbarkeit in ein lineares Programm gesichert worden. Im Sinne von Dempster und Hutton [DH97, DH99] wäre man an dieser Stelle fertig. Das Problem wird diskretisiert und an einen verfügbaren (kommerziellen) LP-Löser übergeben. Jedoch wird so nicht auf die Eigenheiten des Problems eingegangen und damit ein möglicherweise großer Overhead in Kauf genommen. Wie schon in Kapitel 2.2.3 angesprochen, haben Dempster und Hutton in Zusammenarbeit mit Richards bereits einen Algorithmus entwickelt, der diese Feinheiten ausnutzt und damit wesentlich schneller läuft. Dieser soll zunächst beschrieben werden und danach auf seine Tauglichkeit für das modifizierte Problem getestet werden.

### 4.1. Der Algorithmus von Dempster, Hutton und Richards

Dempster, Hutton und Richards nutzen in ihrer Arbeit [DHR98] zwei wesentliche Dinge aus. Zum einen ist dies die Verwandtschaft zweier Teilprobleme in aufeinanderfolgenden Zeitschritten, zum anderen die Eigenschaften des Komplementaritätsproblems:

$$(\text{OCP}) \quad \begin{cases} V(\cdot, T) = \tilde{\Psi} \\ V \geq \tilde{\Psi} \\ -\mathcal{L}V - \frac{\partial V}{\partial t} \geq 0 \\ \left(-\mathcal{L}V - \frac{\partial V}{\partial t}\right) \wedge (V - \tilde{\Psi}) = 0, \end{cases} \quad (4.1)$$

mit dem Black-Scholes Differentialoperator

$$\mathcal{L} := \frac{\sigma^2}{2} S^2 \frac{\partial^2}{\partial S^2} + (r - \delta) S \frac{\partial}{\partial S} - r. \quad (4.2)$$

Anhand der vorletzten Gleichung sieht man sofort die zwei **implizit** definierten Bereiche:

$$\mathcal{S} := \left\{ (S, t) \in [0, \infty[ \times [0, T] \text{ mit } V(S, t) = \tilde{\Psi}(S, t) \right\}, \quad (4.3)$$

den Stoppbereich, in dem eine sofortige Ausübung sinnvoll ist. Der Optionswert entspricht hier dem Inneren Wert. Sowie

$$\mathcal{H} := \left\{ (S, t) \in [0, \infty[ \times [0, T] \text{ mit } \frac{\partial V}{\partial t} + \mathcal{L}V = 0 \right\}, \quad (4.4)$$

den Haltebereich, in dem auf die Ausübung verzichtet wird. Der Übergang zwischen den komplementären Bereichen ist die ebenfalls nur **implizit** definierte Ausübungsgrenze  $S_f(t)$ . Dempster, Hutton und Richards machen sich neben den Eigenschaften der Ausübungsgrenze zu Nutze, dass sowohl der Stoppbereich als auch der Haltebereich einfach zusammenhängend sind.

Das nach der Idee von Dempster und Hutton [DH99] in ein Lineares Programm überführte und zeitschrittweise diskretisierte Optionsbewertungsproblem hat für  $c > 0$  die Gestalt:

$$\begin{aligned} &\text{Minimiere } c^T u^{(m)}, \\ &\text{so dass } \quad u^{(m)} \geq \phi^{(m)}, \\ &\quad \quad \quad A u^{(m)} \geq B u^{(m-1)}. \end{aligned} \tag{4.5}$$

Es wird über die Verwendung des Zeitwertes der Option  $y^{(m)} := u^{(m)} - \phi^{(m)}$  (vgl. [DHR98]) zu dem Problem

$$\begin{aligned} &\text{minimiere } c^T y^{(m)}, \\ &\text{so dass } \quad y^{(m)} \geq 0, \\ &\quad \quad \quad A y^{(m)} \geq B u^{(m-1)} - A \phi^{(m)} := b^{(m)}. \end{aligned} \tag{4.6}$$

Durch die Einführung nichtnegativer Schlupfvariablen  $s^{(m)}$  erhält es die Form:

$$\begin{aligned} &\text{Minimiere } c^T y^{(m)}, \\ &\text{so dass } \quad y^{(m)} \geq 0, \\ &\quad \quad \quad s^{(m)} \geq 0, \\ &\quad \quad \quad (-I \mid A) \begin{pmatrix} s^{(m)} \\ y^{(m)} \end{pmatrix} = b^{(m)}. \end{aligned} \tag{4.7}$$

Dieses überbestimmte Gleichungssystem reduzieren Dempster, Hutton und Richards wiederum durch ‚Streichung‘ der Nichtbasisvariablen auf ein eindeutig bestimmtes Gleichungssystem. Genauer gesagt ist der Wert dieser Nichtbasisvariablen identisch Null, d.h. entweder der Zeitwert  $y_k^{(m)}$  ist identisch Null oder aber für die Schlupfvariable gilt  $s_k^{(m)} = 0$ . An dieser Stelle geht nun der einfache Zusammenhang der Gebiete und deren Komplementarität ein, denn für den Put repräsentieren die Variablen mit Wert Null gerade die mittleren Spalten der Matrix  $(-I \mid A)$ . Es gilt dann:

$$\begin{aligned} s_k^{(m)} &> 0 \quad \text{für} \quad 1 \leq k \leq nb, \\ y_k^{(m)} &= 0 \quad \text{für} \quad 1 \leq k \leq nb, \\ s_k^{(m)} &= 0 \quad \text{für} \quad nb + 1 \leq k \leq N, \\ y_k^{(m)} &> 0 \quad \text{für} \quad nb + 1 \leq k \leq N, \end{aligned} \tag{4.8}$$

mit  $nb$  als größtem Index aller Knoten, die im Stoppbereich liegen. Hilfsweise lässt sich hier ein System  $D z^{(m)} = b^{(m)}$  mit der Bedingung  $z^{(m)} \geq 0$  und Diskretisierungsmatrix

$$D := \begin{pmatrix} -1 & 0 & \cdots & 0 & A_{1,nb+1} & \cdots & A_{1,N} \\ 0 & \ddots & \ddots & \vdots & \vdots & & \vdots \\ \vdots & \ddots & \ddots & 0 & \vdots & & \vdots \\ \vdots & & \ddots & -1 & A_{nb,nb+1} & \cdots & A_{nb,N} \\ \vdots & & & 0 & \vdots & & \vdots \\ \vdots & & & \vdots & \vdots & & \vdots \\ 0 & \cdots & \cdots & 0 & A_{N,nb+1} & \cdots & A_{N,N} \end{pmatrix}$$

lösen. Die Zuordnung der Lösung erfolgt gemäß

$$s_k^{(m)} := \begin{cases} z_k^{(m)} & \text{für } k \leq nb \\ 0 & \text{sonst} \end{cases},$$

$$y_k^{(m)} := \begin{cases} 0 & \text{für } k \leq nb \\ z_k^{(m)} & \text{sonst} \end{cases}.$$

Zeitschrittweise sind die Probleme miteinander verwandt. Die monoton fallende Ausübungsgrenze bewirkt bei einem Wechsel des Ausübungsknotens einen Übergang des Wertes der Variablen  $y_{nb}^{(m)}$  von Null zu einer positiven Zahl und dementsprechend wird der Wert von  $s_{nb}^{(m)}$  Null. Dieser Knotenwechsel kann einfach durch einen Tausch der  $nb$ -ten Spalte

$$\begin{pmatrix} 0 \\ \vdots \\ 0 \\ -1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \leftarrow nb\text{-te Zeile} \quad \text{gegen die Spalte} \quad \begin{pmatrix} A_{1,nb} \\ \vdots \\ A_{nb,nb} \\ \vdots \\ A_{N,nb} \end{pmatrix} \quad (4.9)$$

innerhalb der Diskretisierungsmatrix  $D$  beschrieben werden. Dempster, Hutton und Richards verwenden aufgrund der nicht wachsenden Ausübungsgrenze eine vom Element  $A_{N,N}$  ausgehende Matrixzerlegung. Das Vorgehen entspricht bis auf die Zerlegungsrichtung einer Idee von Bartels und Golub [BG69], den Simplex Algorithmus auf eine LU-Zerlegung zurückzuführen.

Für eine tridiagonale Diskretisierungsmatrix  $D$  ergibt sich der Algorithmus für die UL-Zerlegung:

**Algorithmus 4.1.1. *Algorithmus von Dempster, Hutton und Richards***  
*(DHR-Algorithmus) Abwandlung auf variable Koeffizienten von [DHR98, Gleichungen (24), (25), S. 70 f.]*

$$L_{N,N} = A_{N,N}$$

$$U_{N-1,N} = \frac{A_{N-1,N}}{L_{N,N}}$$

*i-Schleife (1)*       $i = N - 1, \dots, nb + 1$

$$L_{i+1,i} = A_{i+1,i}$$

$$L_{i,i} = A_{i,i} - L_{i+1,i} U_{i,i+1}$$

$$U_{i-1,i} = \frac{A_{i-1,i}}{L_{i,i}}$$

*i-Schleife (2)*       $i = nb, \dots, 1$

$$L_{i+1,i} = U_{i-1,i} = 0$$

$$L_{i,i} = -1.$$

Der oben beschriebene Spaltentausch lässt sich leicht in den Algorithmus integrieren. Es ist lediglich eine Anpassung der Zerlegung über

**Algorithmus 4.1.2. *UL-Update für DHR-Algorithmus***  
*In Anlehnung an [DHR98, Gleichung (25), S. 71 ]*

$$L_{nb+1,nb} = A_{nb+1,nb}$$

$$L_{nb,nb} = A_{nb,nb} - L_{nb+1,nb} U_{nb,nb+1}$$

$$U_{nb-1,nb} = \frac{A_{nb-1,nb}}{L_{nb,nb}}$$

$$nb = nb - 1.$$

notwendig.

Ein Lösungsschritt sei mittels des Flussdiagramms in Abbildung 4.1 schematisch beschrieben.

## 4.2. Notwendige Abwandlung des Algorithmus

Wir haben im vorangegangenen Abschnitt gesehen, dass eine wesentliche Notwendigkeit für den beschleunigten Algorithmus die nicht wachsende Übungsgrenze  $S_f(\tau)$  ist. Zusätzlich hat van Moerbeke [vM76] gezeigt, dass die Übungsgrenze für  $\tau > 0$  eine stetig



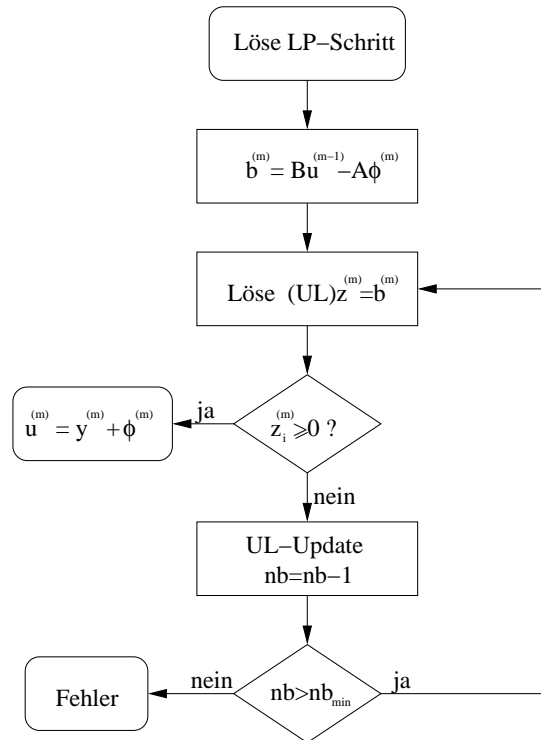


Abbildung 4.1.: Flussdiagramm für einen Zeitschritt des Algorithmus von Dempster, Hutton und Richards. Der Abbruch mit Fehler ist meist auf einen zu kleinen Diskretisierungsbereich zurückzuführen.

differenzierbare Funktion ist. Zusammen ergibt dies die Aussage

$$S'_f(\tau) := \frac{\partial S_f(\tau)}{\partial \tau} \leq 0. \quad (4.10)$$

Dies bleibt für  $x = \ln(S)$ , die von Dempster, Hutton und Richards verwendete Transformation, in den neuen Koordinaten richtig, da der Logarithmus eine streng monoton wachsende Funktion ist. Erinnern wir uns nun an die beiden als geeignet angesehenen Transformationen. Im ersten Fall ist dies die Transformation von Smith (Konvektions-Verschiebung)

$$x := \ln\left(\frac{S}{K}\right) + \int_0^\tau r(T-y) - \delta(T-y) - \frac{\sigma(T-y)^2}{2} dy.$$

Die Ausübungsgrenze in den neuen Koordinaten lautet dann:

$$x_f(\tau) := \ln\left(\frac{S_f(\tau)}{K}\right) + \int_0^\tau r(T-y) - \delta(T-y) - \frac{\sigma(T-y)^2}{2} dy.$$

Durch Differenzieren erhält man dann folgende Aussage über die Monotonie:

$$x'_f(\tau) := \underbrace{\frac{1}{S_f(\tau)}}_{\geq 0} \underbrace{S'_f(\tau)}_{\leq 0} + \underbrace{r(T-\tau) - \delta(T-\tau) - \frac{\sigma(T-\tau)^2}{2}}_{?} \stackrel{(!)}{\leq} 0.$$

Das Vorzeichen von  $r - \delta - \frac{\sigma^2}{2}$  zum Zeitpunkt  $T - \tau$  ist im Allgemeinen nicht festgelegt. Aus den Daten der Problemstellung ist es zwar bekannt, aber die optimale Ausübungsgrenze bleibt unbekannt. Insbesondere für den interessanten Fall einer konvektionsdominanten Gleichung ist das Vorzeichen jedoch höchstwahrscheinlich positiv. Hier lässt sich dann nur die Aussage treffen:

**Korollar 4.2.1. Stabilität des DHR-Algorithmus für Konvektions-Verschiebung**

i) *Hinreichend für einen sicheren Durchlauf des Algorithmus 4.1.1 ist*

$$r(T-\tau) - \delta(T-\tau) - \frac{\sigma(T-\tau)^2}{2} \leq 0 \quad (4.11)$$

für jedes  $\tau \geq 0$ .

ii) *Für  $r(T-\tau) - \delta(T-\tau) - \frac{\sigma(T-\tau)^2}{2} > 0$  kann der Algorithmus abbrechen.*

iii) *Gibt es ein  $\tau \geq 0$  mit*

$$-\frac{1}{S_f(\tau)} S'_f(\tau) < r(T-\tau) - \delta(T-\tau) - \frac{\sigma(T-\tau)^2}{2}, \quad (4.12)$$

so wird der Algorithmus abbrechen.

**Beweis:**

Wir testen, ob die Ausübungsgrenze in den transformierten Koordinaten nicht wachsend verläuft. Diese Aussage ist im ersten Fall sicher und wird im letzten Fall verletzt.  $\square$

Leider lässt sich die Aussage für die kritischen Fälle, insbesondere die Möglichkeit eines Abbruchs des Algorithmus nicht nutzen. Hier gilt:

**Bemerkung 4.2.2. Praxisrelevanz der DHR-Stabilität für die Konvektions-Verschiebung**

*Die Bedingung (4.12) ist nur von theoretischer Bedeutung. Da  $S_f(\tau)$  und auch  $S'_f(\tau)$  implizit mit dem Problem zusammen als freier Rand bestimmt werden müssen, kann diese Bedingung a priori nicht geprüft werden. Insbesondere kann man für die Praxis nur die erste Aussage des Korollars nutzen.*

Für die neue Transformation kommt man über

$$x := \ln\left(\frac{S}{K}\right) + \int_0^\tau r(T-y) - \delta(T-y) dy$$

zu dem analogen Ergebnis

$$x'_f(\tau) := \frac{1}{S_f(\tau)} S'_f(\tau) + r(T - \tau) - \delta(T - \tau) \stackrel{!}{\leq} 0.$$

Das Korollar 4.2.1 lautet nun dementsprechend:

**Korollar 4.2.3. Stabilität des DHR-Algorithmus für die neue Transformation**

i) *Hinreichend für einen sicheren Durchlauf des Algorithmus 4.1.1 ist*

$$r(T - \tau) - \delta(T - \tau) \leq 0 \tag{4.13}$$

*für jedes  $\tau \geq 0$ .*

ii) *Für  $r(T - \tau) - \delta(T - \tau) > 0$  kann der Algorithmus abbrechen.*

iii) *Gibt es ein  $\tau \geq 0$  mit*

$$-\frac{1}{S_f(\tau)} S'_f(\tau) < r(T - \tau) - \delta(T - \tau), \tag{4.14}$$

*so wird der Algorithmus abbrechen.*

**Beweis:**

Analog zu Korollar 4.2.1. □

Die Situation verbessert sich hier auch nicht, weil folgende Bemerkung gilt:

**Bemerkung 4.2.4. Praxisrelevanz der DHR-Stabilität für die neue Transformation**

*Die Bedingung (4.13) wird in der Regel für normale Parametersätze nicht erfüllt sein, da der risikofreie Zinssatz meist größer als die Dividendenrate ist. Die Bemerkung 4.2.2 gilt für die Gleichung (4.14) sinngemäß.*

Hier zeigt sich, noch stärker als bei der Smith Transformation, dass ein einfaches Verwenden des DHR-Algorithmus 4.1.1 mit Schwierigkeiten verbunden sein kann. Zur Verdeutlichung schaue man sich noch einmal die Ausübungsgrenze in den verschiedenen Koordinatensystemen an. Für konstante Parameter  $r$ ,  $\delta$  und  $\sigma$  zeigt sich in dieser schematischen Darstellung ein lokales Minimum für  $x_f(\tau)$ . Die Abbildung 4.2 zeigt ebenfalls sofort einen Weg auf, den DHR-Algorithmus zu modifizieren. Wir haben festgestellt, dass die Ausübungsgrenze  $x_f(\tau)$  in diesem Falle genau ein lokales Minimum für  $\tau^*$  besitzt. Vom Ausübungszeitpunkt an bis zum Zeitpunkt  $\tau^*$  verläuft die Ausübungsgrenze monoton fallend und danach monoton steigend. Damit muss für den Algorithmus ‚einfach‘ zum richtigen Zeitpunkt die Suchrichtung der Ausübungsgrenze geändert werden. Entsprechend kann im Fall nicht konstanter Parameter und damit möglicherweise mehrerer lokaler

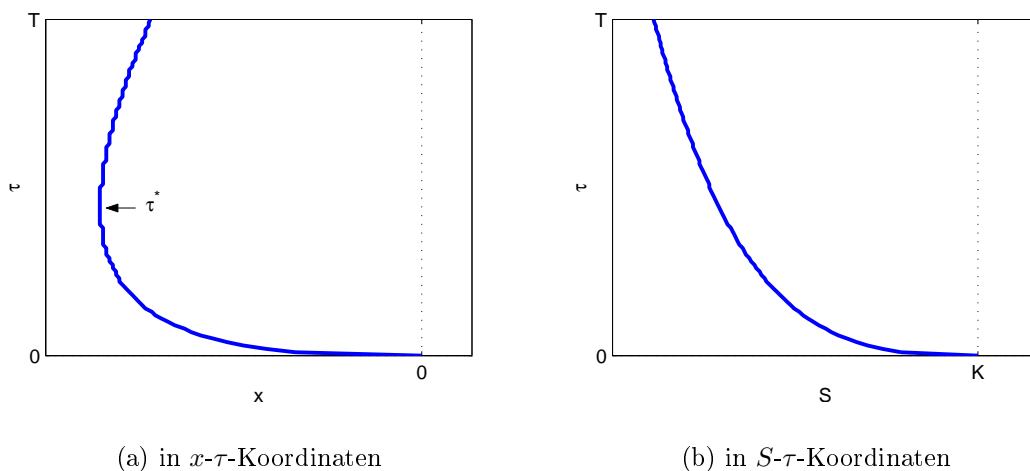


Abbildung 4.2.: Schematischer Verlauf der Ausübungsgrenze: In den  $S$ - $\tau$ -Koordinaten ist diese wie in der Theorie nicht wachsend. In diesem Beispiel ist sie sogar monoton fallend. In den  $x$ - $\tau$ -Koordinaten liegt in  $\tau^*$  ein lokales Minimum vor.

Extrema die Suchrichtung an mehreren Stellen  $\tau_i^*$  für  $i = 1, 2, \dots$  geändert werden.

Die Umkehr des Spaltentausches gemäß Gleichung (4.9) bedeutet, die Spalte

$$\begin{pmatrix} A_{1,nb} \\ \vdots \\ A_{nb,nb} \\ \vdots \\ A_{N,nb} \end{pmatrix} \quad \text{wieder gegen die Spalte} \quad \begin{pmatrix} 0 \\ \vdots \\ 0 \\ -1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \leftarrow nb\text{-te Zeile} \quad (4.15)$$

auszuwechseln. Dies ist nichts anderes als die Invertierung des Algorithmus 4.1.2. Die Bezeichnung mit der hochgestellten minus Eins für das Inverse ist daher fast zwingend.

**Algorithmus 4.2.5.  $UL$ -Update<sup>-1</sup> für  $DHR$ -Algorithmus**

$$\begin{aligned}
 L_{nb+1,nb} &= 0 \\
 L_{nb,nb} &= -1 \\
 U_{nb-1,nb} &= 0
 \end{aligned}$$

$$nb = nb + 1.$$

Der bestehende Algorithmus wird um diese Umkehr erweitert. Ein Abbruch des Verfahrens tritt nun erst dann ein, wenn auch für positive Suchrichtung, d.h. die Hilfsvariable  $nb$  wird erhöht, keine zulässige Lösung im Sinne von  $z^{(m)} \geq 0$  mehr gefunden wird. Anschaulich wird dieser Ablauf in Abbildung 4.3 beschrieben. Insbesondere sieht man dort, dass die Geschwindigkeit des Algorithmus entscheidend von der Wahl der Parameter  $nb_{min}$  bzw.  $nb_{max}$  abhängt. Unter der Voraussetzung, dass die Grenze nur ein Minimum aufweist, hängt sie sogar im Wesentlichen nur von der Wahl von  $nb_{min}$  ab. Ein Umschalten der Suchrichtung findet erst dann statt, wenn  $nb_{min}$  unterschritten wird. Liegen jedoch viele Gitterpunkte zwischen  $nb_{min}$  und dem bisherigen Index  $nb_{alt}$ , so muss in **jedem(!)** Schritt ein UL-Update und die dazugehörigen 2 Gleichungssysteme für die Prüfung, ob  $z$  zulässig ist, gerechnet werden. Bei hinreichend kleiner Schrittweite in  $\tau$ -Richtung kann man jedoch davon ausgehen, dass der Index  $nb$  nicht weit springen wird. Somit ist eine Definition als Funktion von der bisherigen Grenze, d.h.  $nb_{min}(nb_{alt})$ , sinnvoll, um den Rechenaufwand zu minimieren.

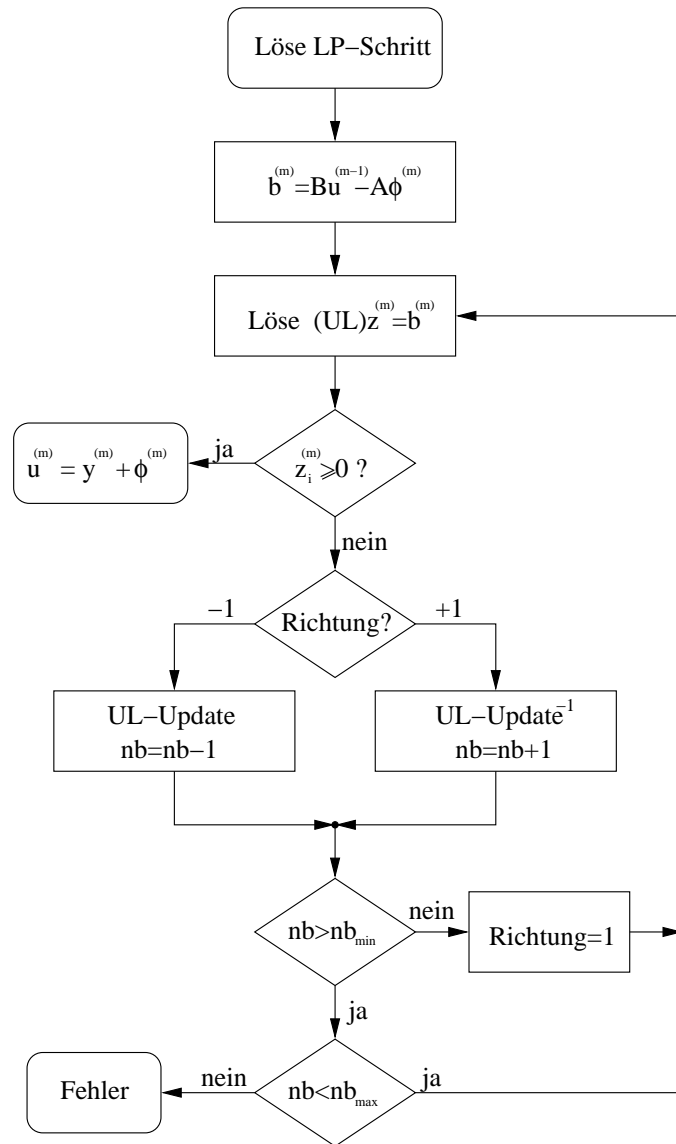


Abbildung 4.3.: Flussdiagramm für einen Zeitschritt des modifizierten Algorithmus von Dempster, Hutton und Richards. Der Basisalgorithmus ist um den inversen Spalten-tausch ergänzt worden.



## 5. Vorbereitung der Schrittweitensteuerung

Im bisherigen Teil der Arbeit ist nur wenig Rücksicht auf die Genauigkeit der Lösung bzw. die Geschwindigkeit des Lösungsvorgangs genommen worden. Bezüglich der Konvergenz ist die Aussage von Jaillet et al. [JLL90] (Satz 2.2.2) angesprochen worden. Die Resultate der Arbeiten von Dempster, Hutton und Richards zur Berechnungsdauer sind ebenfalls angeführt worden. Beide lassen sich informell auf folgende Faustregeln bringen:

### **Bemerkung 5.0.1. Faustregeln zur Geschwindigkeit und Genauigkeit**

- i) Je kleiner  $\Delta\tau$  und  $\Delta x$  gewählt werden, desto genauer wird die Näherungslösung u. Diese Aussage wird vorbehaltlich möglicher Rundungsfehler getroffen.*
- ii) Der LP-Löser ist schneller als iterative Algorithmen.*

Offen gelassen wird jedoch, wie viel Aufwand für eine Lösung mit gewünschter Genauigkeit notwendig ist. Man kann jedoch mit der zweiten informellen Aussage arbeiten. Wenn der DHR-Algorithmus in jedem Schritt schneller als ein Relaxationsverfahren ist, dann kann man den Algorithmus dadurch weiter beschleunigen, dass man die Notwendigkeit einzelner Schritte in Frage stellt.

### 5.1. Motivation

Wir betrachten nochmals das lineare Komplementaritätsproblem aus Gleichung (4.1). Die Bewertung einer amerikanischen Option entspricht der Lösung dieses Problems. Wäre die Ausübungsgrenze bekannt, so wäre damit auch die Lösung im Stoppbereich  $\mathcal{S}$  bekannt und man könnte die Lösung für  $\mathcal{H}$  mittels der PDE

$$\frac{\partial V}{\partial t} + \frac{\sigma^2}{2} S^2 \frac{\partial^2 V}{\partial S^2} + (r - \delta) S \frac{\partial V}{\partial S} - r V = 0 \quad (5.1)$$

berechnen. Dies ist **nicht** der Fall, jedoch kann man das Verhalten der Lösung auf  $\mathcal{H}$  für den Fall einer bekannten Ausübungsgrenze analysieren. Wie bereits gezeigt, lässt sich die letzte Gleichung mittels der Konvektions-Verschiebung aus Kapitel 1.3 auf eine Wärmeleitungsgleichung der Form

$$v_\tau - \frac{\sigma^2(T - \tau)}{2} v_{xx} = 0$$

reduzieren. Für diese Gleichung lassen sich nun über einen Separationsansatz Lösungen der Form

$$v(x, \tau) = \exp\left(-\frac{\sigma^2(T - \tau)}{2}\lambda^2\tau\right)\left(G \cos(\lambda x) + H \sin(\lambda x)\right)$$

bestimmen.  $G$  und  $H$  hängen dabei von den Anfangswerten und der Randbedingung, insbesondere für den freien Rand  $x_f$ , ab. Man sieht hier jedoch schon das typische Verhalten der Wärmeleitungsgleichung. Die hochfrequenten Anteile der Lösung werden sehr schnell klein, so dass die Berechnungsschrittweite vergrößert werden kann. Ebenfalls in diese Richtung weist das Verhalten der Ausübungsgrenze nach diskreter Approximation. Hier ist mit

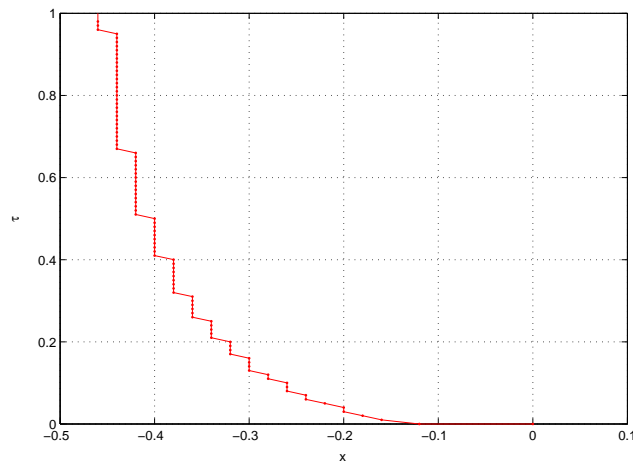


Abbildung 5.1.: Verlauf der Ausübungsgrenze:  $\Delta x = 0.02$ ,  $\Delta \tau = 0.01$

fortschreitender Zeit  $\tau$  ein immer seltener werdender Wechsel des Ausübungsknotens zu beobachten. Als Ausübungsknoten wird der Knoten bezeichnet, dessen Index mit der Variable  $nb$  übereinstimmt. Beispielhaft ist dies in der Abbildung 5.1 gezeigt. Dort verhält sich die Ausübungsgrenze für  $0.7 \leq \tau \leq 0.95$  sozusagen ‚konstant‘. Gleichzeitig springt die Ausübungsgrenze im ersten Schritt um 6 Knoten. Hier bietet sich ein Einstiegspunkt für Verfeinerungen.

## 5.2. Vorwissen

Bevor wir uns mit der Konstruktion einer Schrittweitensteuerung auseinandersetzen, sollen die Eigenschaften der Lösung betrachtet werden. Diese werden, wie sich später zeigen wird, das entstehende Verfahren entscheidend beeinflussen.



### 5.2.1. Startwerte sind nur stetig

Wenn wir uns die Auszahlungsfunktion noch einmal ansehen, dann stellen wir fest, dass diese nicht global zweimal stetig differenzierbar ist. Genauer gilt:

**Beobachtung 5.2.1. Fehlende Differenzierbarkeit am Strike**

i) Die Auszahlungsfunktion

$$\Psi(S, T) = \begin{cases} S - K & ; S < K \\ 0 & ; S \geq K \end{cases} \quad (5.2)$$

ist am Strike, d.h. für  $S = K$  nur stetig, aber nicht differenzierbar.

ii) In den transformierten Koordinaten korrespondiert der Strike mit der Stelle  $x = 0$ . Auch hier ist die transformierte Auszahlungsfunktion

$$\phi(x, 0) = \begin{cases} K(1 - \exp(x)) & ; x < 0 \\ 0 & ; x \geq 0 \end{cases} \quad (5.3)$$

nur stetig.

Setzt man ohne dies zu berücksichtigen mit einer Crank-Nicolson-Diskretisierung an, so verhält sich der Diskretisierungsfehler für den ersten Zeitschritt nicht so, wie es nach der Theorie zu erwarten wäre. Denn es gilt:

**Lemma 5.2.2. Diskretisierungsfehler in der Praxis**

*Es sei eine Transformation der Variablen mittels der Konvektions-Verschiebung durchgeführt worden. Versucht man den ersten Zeitschritt der transformierten Gleichung mittels Crank-Nicolson-Verfahren zu diskretisieren, so erhält man für  $x = 0$  einen Diskretisierungsfehler der Ordnung  $\mathcal{O}((\Delta x)^{-1})$  statt der theoretischen  $\mathcal{O}((\Delta x)^2)$ .*

**Beweis:**

Nach der Transformation mittel Konvektions-Verschiebung (vgl. Kapitel 1.10) ist die Gleichung

$$v_\tau - \frac{\sigma^2}{2} v_{xx} = 0$$

zu lösen. Für den ersten Zeitschritt, d.h.  $\tau_0 = 0$  und  $\tau_1 = \Delta\tau$  ergibt sich eine Diskretisierung

mittels des Crank-Nicolson-Schema der Form

$$\begin{aligned}
 & v_\tau(x_i, \Delta\tau) - \frac{\sigma^2}{2} v_{xx}(x_i, \Delta\tau) - \frac{v(x_i, \Delta\tau) - v(x_i, 0)}{\Delta\tau} \\
 & + \frac{\sigma^2}{4} \frac{v(x_{i+1}, \Delta\tau) - 2v(x_i, \Delta\tau) + v(x_{i-1}, \Delta\tau)}{\Delta x^2} \\
 & + \frac{\sigma^2}{4} \underbrace{\frac{v(x_{i+1}, 0) - 2v(x_i, 0) + v(x_{i-1}, 0)}{\Delta x^2}}_{=:H(x_i, 0)} \\
 = & \underbrace{\frac{\Delta\tau}{2} v_{\tau\tau}(x_i, \Delta\tau) - \frac{\sigma^2}{4} v_{xx}(x_i, \Delta\tau)}_{=:G(x_i, \Delta\tau)} + \frac{\sigma^2}{4} H(x_i, 0) + \mathcal{O}((\Delta\tau)^2) + \mathcal{O}((\Delta x)^2)
 \end{aligned}$$

Der Ausdruck  $H(x_i, 0)$  müsste dieselbe Größe wie die beiden vorangehenden Terme haben, um das theoretische Ergebnis zu erhalten. Verwendet man die Fälligkeitsbedingung des Puts

$$\phi(x, 0) = \begin{cases} K(1 - \exp(x)) & ; x < 0 \\ 0 & ; x \geq 0 \end{cases}$$

so ergibt sich auf der einen Seite:

$$H(x_i, 0) = \begin{cases} -K \exp(x) + \mathcal{O}((\Delta x)^2) & ; x_i \leq -\Delta x \\ K \left( \frac{1}{\Delta x} - \frac{1}{2} \right) + \mathcal{O}(\Delta x) & ; x_i = 0 \\ 0 & ; x_i \geq \Delta x. \end{cases}$$

Auf der anderen Seite liegt der Punkt  $(0, \Delta\tau)$  für jedes  $\Delta\tau > 0$  im Haltebereich  $\mathcal{H}$ . Die Funktion  $G$  wird sich daher nicht wie  $\frac{1}{\Delta x}$  verhalten.  $\square$

Das Verhalten der Lösung bei der Iteration lässt sich am einfachsten anhand eines Beispiels illustrieren.

### Beispiel 5.2.3. Testproblem amerikanische Standard-Option

Man berechne den Preis eines amerikanischen Vanilla Puts mit  $K = 5.0$ ,  $T = 1.0$ ,  $r = 0.1$ ,  $\delta = 0$  und  $\sigma = 0.5$ .

Für dieses Beispiel wird eine Iteration sowohl mit dem impliziten Euler-Verfahren als auch mit dem Crank-Nicolson-Verfahren durchgeführt. Die berechnete Lösung, unter der Barraquand und Pudet Transformation, wird in der Abbildung 5.2 gezeigt. Für das implizite Euler-Verfahren hat die Lösung die gewünschten Konvexitätseigenschaften (vgl. dazu Lemma 8.1.1 oder [Mer73]). Für die mit dem Crank-Nicolson-Verfahren berechnete Lösung ergibt sich jedoch ein Knick für  $x = 0$ , d.h. an der Stelle des Strikes.

Dieser Knick hat seinen Ursprung in der fehlenden Differenzierbarkeit der Anfangsbedingung an dieser Stelle. Wie sich nach Lemma 5.2.2 erwarten lässt, verstärkt sich dieser

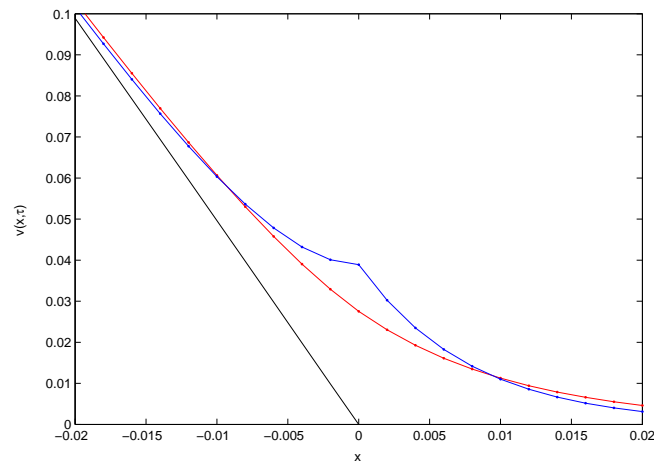


Abbildung 5.2.: Knick in der Wertfunktion für Beispiel 5.2.3 mit  $x_{min} = -1.0$ ,  $x_{max} = 1$  und  $\Delta x = 0.002$  sowie  $\Delta\tau = 0.001$ . Hierbei sind die Lösungen des **impliziten Euler-Verfahrens** und des **Crank-Nicolson-Verfahrens** nach einem Iterationsschritt d.h. für  $\tau = \Delta\tau = 0.001$  geplottet.

Effekt bei in  $x$ -Richtung verfeinertem Gitter, d.h. kleinerem  $\Delta x$ . Natürlich pflanzt sich dieser Fehler in den nachfolgenden Zeitschritten fort.

Abhilfe schafft hier die von Rannacher [Ran84] eingeführte Vorgehensweise, die auch als ‚Rannacher smoothing‘ bezeichnet wird. In dieser Arbeit untersucht Rannacher das Verhalten von Konvektions-Diffusions Gleichungen mit unstetigen Anfangswerten. Die Iteration wird mit einer endlichen Anzahl  $m_0$  von impliziten Euler-Schritten der Schrittweite  $\Delta\tau/2$  begonnen und danach mit dem Crank-Nicolson-Verfahren und einer Schrittweite  $\Delta\tau$  fortgesetzt. Bevor wir dieses Vorgehen erläutern, müssen zunächst zwei Begriffe geklärt werden:

**Definition 5.2.4. A-Stabilität/L-Stabilität**

Es sei die Testgleichung  $w' = \lambda w$  mit  $\Re(\lambda) < 0$  gegeben. Ein Einschrittverfahren habe die Vorschrift  $w^{(m+1)} = R(z)w^{(m)}$  für  $z = \Delta\tau\lambda$ . Die Funktion  $R$  heißt **Stabilitätsfunktion** und das Verfahren heißt **A-stabil**, wenn das Stabilitätsgebiet  $\{z \in \mathbb{C}; |R(z)| \leq 1\}$  in der negativen komplexen Halbebene liegt. Das Verfahren heißt **L-stabil** oder auch **streng A-stabil**, wenn zusätzlich  $\lim_{z \rightarrow \infty} R(z) = 0$  gilt.

Mit Hilfe dieser Definition kann dann die Glättung der Anfangsbedingungen durch das folgende Lemma erklärt werden.

**Lemma 5.2.5. Glättung durch L-Stabilität des Verfahrens**

- i) Ein L-stabiles Verfahren glättet die hochfrequenten Fehlerkomponenten.
- ii) Das implizite Euler-Verfahren ist L-stabil, das Crank-Nicolson-Verfahren ist nur A-stabil.

**Beweis:**

- i) Z.B. in [HW96] bei der Herleitung der Definition.
- ii) Nachrechnen mit den Stabilitätsfunktionen  $R_{impl.}(z) = \frac{1}{1-z}$  und  $R_{CN}(z) = \frac{1+z/2}{1-z/2}$ .  $\square$

Rannacher weist ausdrücklich auf die Glättung von hochfrequenten Anteilen durch das implizite Euler-Verfahren hin. Die endliche Anzahl der impliziten Euler-Schritte beeinflusst die globale Konvergenzordnung von Crank-Nicolson nicht. Es reicht sogar die Verwendung von  $m_0 = 2$  impliziten Schritten aus. Das Vorgehen lässt sich in nachfolgendem Algorithmus zusammenfassen.

**Algorithmus 5.2.6. Rannacher smoothing [Ran84, S. 311/312]**

- i) Berechnung der ersten 2 Zeitschritte mit dem impliziten Euler-Verfahren
- ii) Fortsetzung der Berechnung mit dem Crank-Nicolson-Verfahren

Im Kontext der Optionsbewertung ist dieses Verfahren bereits von Pooley et al. [PFV03] verwendet worden. Für die betrachteten Optionen zeigt sich dort eine wiedererlangte quadratische Konvergenzordnung. Allerdings sind die in dieser Arbeit berechneten Optionen nur vom *europäischen* Typ.

Die eigentliche Problematik klingt in der Arbeit von Rannacher und in der Pooley Arbeit an. Rannacher nimmt sowohl die Randbedingungen als auch den Quellterm in der PDE als glatte Funktionen in  $x$  und  $\tau$  an. Es dürfen dabei endlich viele Stellen auftreten, an denen die Funktionen unstetig in  $\tau$  ist. Dann muss nicht nur für die Anfangsbedingung, sondern auch nach jedem dieser Sprünge der Algorithmus 5.2.6 angewendet werden. In der Arbeit von Pooley et al. wird dies rigoros über “... *Rannacher time-stepping must occur at all non-smooth states, ie, whenever a discontinuity is introduced*”<sup>1</sup> [PFV03, S. 32] formuliert.

Dies lässt sich zusammenfassen zu:

**Korollar 5.2.7. Rannacher smoothing bei fehlender Glattheit**

*An jeder Stelle  $\tau$ , an der die Anfangsbedingung, Quellterme innerhalb der PDE oder aber die Randbedingung die Glattheitsanforderungen nicht erfüllen, muss Rannacher smoothing, (Algorithmus 5.2.6) verwendet werden, um die Konvergenzordnung des Crank-Nicolson-Verfahrens zu erhalten.*

---

<sup>1</sup>Fettdruck fehlt im Originaltext

Die folgende Bemerkung bezieht sich auf den Versuch, ohne Rannacher smoothing auszukommen.

**Bemerkung 5.2.8. Lage der Gitterpunkte**

Die Bemerkung aus der Arbeit von Pooley et al. zum Entfallen der Glättung ist hier gegenstandslos. Im zitierten Originaltext wird darauf verwiesen, dass für eine stetige Auszahlungsfunktion mit unstetigen Ableitungen keine Glättung notwendig ist, falls ein Kursknoten in den Ausübungspreis gelegt wird. Durch Anwendung der Transformationen, die das Koordinatensystem mit der Konvektion verschieben, kann diese Forderung für kaum einen Payoff erfüllt werden.

Im Falle eines Vanilla Put ist hiermit der Punkt  $x_0$  mit

$$K \left( 1 - \exp(x_0) \exp \left( - \left( r - \delta - \frac{\sigma^2}{2} \right) \tau \right) \right) = 0$$

gemeint. Es ist aber unmöglich, für ein konstantes Gitter den Punkt

$$x_0(\tau) = \left( r - \delta - \frac{\sigma^2}{2} \right) \tau$$

für jedes  $\tau$  in einen Knoten zu legen.

Ebenso wollen wir anmerken, dass testweise eine Aufteilung in Stopp- und Haltebereich vorgenommen wurde. Das Resultat findet sich in

**Bemerkung 5.2.9. Teilung an der Ausübungsgrenze**

Eine Anpassung des Differenzenverfahrens durch eine unabhängige Diskretisierung für den Bereich links bzw. rechts des Strikes und zusätzliche Forderung von Stetigkeitsbedingungen für die Funktion und ihre Ableitungen an der Stelle  $x = 0$  liefert eine glatte Lösung, die der des impliziten Euler-Verfahrens ähnelt. Jedoch ist dieses Vorgehen aufgrund der unbekanntenen Ausübungsgrenze und der nachfolgenden Aussagen (insbesondere Lemma 5.2.11) über die Differenzierbarkeit an der Ausübungsgrenze nur für Optionen europäischen Typs geeignet.

Die Überlegung, ob dieser Effekt vom verwendeten Lösungsverfahren abhängt, ist ebenfalls angestellt worden. Hier wurde als Vergleichsverfahren das Projected-Successive-Overrelaxation-Verfahren (PSOR-Verfahren; [WHD95, Kapitel 9.4], [Sey04, Algorithmus 4.12]) angewandt. Bei der iterativen Lösung eines Zeitschrittes finden mehrere Teilschritte des Verfahrens statt, die evtl. den beobachteten Knick glätten könnten. Hier haben wir aber die bzgl. der Glättung negative Beobachtung gemacht:

**Beobachtung 5.2.10. Knick auch in PSOR-Lösung**

- i) Nach einem Schritt des PSOR-Verfahrens weist auch die Lösung der Crank-Nicolson Diskretisierung den beschriebenen Knick auf. Das Auftreten ist dabei unabhängig von der Abbruchtoleranz ( $\epsilon$  bei Wilmott et al. bzw.  $\epsilon$  bei Seydel).
- ii) Auf dem  $S$ -Gitter ist ein Knick deutlich schlechter wahrzunehmen als auf dem  $x$ -Gitter. Hier gilt die Aussage des Lemma 5.2.2: Bei einem feineren Gitter verstärkt sich der Effekt.

### 5.2.2. Eingeschränkte Differenzierbarkeit an der Ausübungsgrenze

Schaut man sich nun die theoretischen Eigenschaften der Lösung an, so stellt man fest, dass an der Ausübungsgrenze ebenfalls nicht die erforderliche Differenzierbarkeit erreicht wird. Hier gilt:

**Lemma 5.2.11. Differenzierbarkeit an der Ausübungsgrenze  $S_f$**

Es sei mit  $S_f(\tau)$  die Ausübungsgrenze zum Zeitpunkt  $\tau$  bezeichnet. Dann gilt:

- i) Der Optionswert  $V(S, \tau)$  ist an der Stelle  $S = S_f(\tau)$  stetig.
- ii) Der Optionswert  $V(S, \tau)$  ist an der Stelle  $S = S_f(\tau)$  stetig nach  $S$  differenzierbar. Für die Ableitung gilt im Falle des amerikanischen Puts

$$\frac{\partial V(S_f(\tau), \tau)}{\partial S} = -1. \tag{5.4}$$

Die so genannte ‚smooth fit condition‘.

- iii) An der Stelle  $S = S_f(\tau)$  ist der Optionswert **nicht** zweimal nach  $S$  differenzierbar.
- iv) An der Stelle  $S = S_f(\tau)$  ist der Optionswert **nicht** nach  $\tau$  differenzierbar.

**Beweis:**

- i) ohne
- ii) Hier gibt es mehrere Ansätze. Man vgl. z.B. [WHD95, S. 110/111] bzw. [Kwo98, S. 138-140].
- iii) Folgt aus der Komplementarität. Im Haltebereich ist der Optionswert streng konvex, im Stoppbereich wird der Optionswert durch die Gerade  $S - K$  für den Call bzw.  $K - S$  für den Put beschrieben.
- iv) Folgt aus der PDE und iii). □

Diese Aussage bleibt auch nach den Transformationen erhalten:

**Lemma 5.2.12. Differenzierbarkeit an der Ausübungsgrenze  $x_f$**

Für Konvektions-Verschiebung bzw. die neue Transformation sei mit  $x_f(\tau)$  die transformierte Ausübungsgrenze zum Zeitpunkt  $\tau$  bezeichnet. Dann gilt:

- i) Der Optionswert  $v(x, \tau)$  ist an der Stelle  $x = x_f(\tau)$  stetig.
- ii) Der Optionswert  $v(x, \tau)$  ist an der Stelle  $x = x_f(\tau)$  stetig nach  $x$  differenzierbar.
- iii) An der Stelle  $x = x_f(\tau)$  ist der Optionswert **nicht** zweimal nach  $x$  differenzierbar.
- iv) An der Stelle  $x = x_f(\tau)$  ist der Optionswert **nicht** nach  $\tau$  differenzierbar.

**Beweis:**

Dazu wird das Lemma 5.2.11 zusammen mit den Aussagen  $\frac{\partial V}{\partial x} = S \frac{\partial V}{\partial S}$  bzw.  $\frac{\partial^2 V}{\partial x^2} = S \frac{\partial V}{\partial S} + S^2 \frac{\partial^2 V}{\partial S^2}$  benutzt. Weiter gilt  $v(x, \tau) = e^{+r\tau} V(x, \tau)$  für Konvektions-Verschiebung bzw.  $v(x, \tau) = e^{\int_0^\tau r(t-y) dy - \frac{1}{2}x} V(x, \tau)$  im Falle der neuen Transformation. Beide Skalierungen sind nach  $x$  beliebig oft differenzierbar.  $\square$

Aufgrund der nicht ausreichenden Differenzierbarkeit ist auch nach der Transformation mit dem Auftreten von Schwierigkeiten bei der Verwendung des Crank-Nicolson-Verfahrens zu rechnen. Auch hier sei zur Illustration wieder das Beispiel 5.2.3 herangezogen. Es werden nun zuerst zwei implizite Euler-Schritte der Länge  $\Delta\tau$  berechnet, also das Rannacher smoothing verwendet. Dann folgen zwei Crank-Nicolson-Schritte der Länge  $\Delta\tau$ . Ausgehend von der Lösung nach dem vierten Schritt  $v_{i,CN}^{(4)}$  wird dann ein weiterer Schritt der Länge  $\Delta\tau$  jeweils mit dem impliziten Euler-Verfahren und mit dem Crank-Nicolson-Verfahren berechnet. Für die beiden Lösungen ergeben sich anhand des Plots keinerlei qualitative Einschränkungen. Raut man jedoch die Lösungen durch Differenzieren auf, in diesem Fall die Approximation durch Differenzenquotienten, so zeigt sich für das Crank-Nicolson-Verfahren in der Abbildung 5.3 wieder ein Knick als Artefakt.

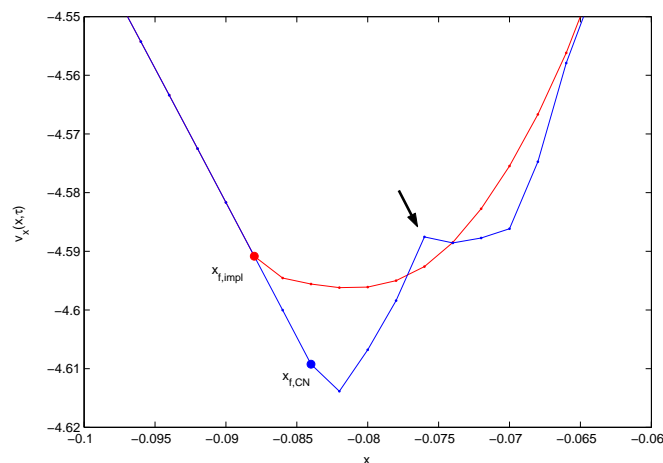


Abbildung 5.3.: Knick in der Ableitung nach  $x$  der Wertfunktion für Beispiel 5.2.3 mit  $x_{min} = -1.0$ ,  $x_{max} = 1.0$  und  $\Delta x = 0.002$  sowie  $\Delta\tau = 0.001$ . Hierbei ist eine Approximation der Ableitung nach  $x$  der Lösung des **impliziten Euler-Verfahrens** bzw. des **Crank-Nicolson-Verfahrens** nach dem fünften Iterationsschritt, d.h. für  $\tau = 5 \cdot \Delta\tau = 0.005$  geplottet. Es wurden zuvor ein Rannacher smoothing der Anfangsbedingung, d.h. zwei implizite Euler-Schritte, und dann zwei Schritte mit dem Crank-Nicolson-Verfahren durchgeführt.

**Beobachtung 5.2.13. Verschiedene Aufsprungpunkte**

Eine mögliche Konsequenz daraus ist auch das Auftreten von verschiedenen Aufsprungpunkten  $x_{f,impl}$  bzw.  $x_{f,CN}$ . Der dem Crank-Nicolson-Verfahren zugeordnete Punkt wird durch den Knick auf der Auszahlungsfunktion nach unten gedrückt.

Auch in diesem Falle liegt die Ursache des Knicks in der nicht hinreichenden Differenzierbarkeit. Vergleicht man noch einmal die Diskretisierung des Operators  $\frac{\partial^2 v}{\partial x^2}$  mittels des Crank-Nicolson-Differenzensterns und die Diskretisierung für das implizite Euler-Verfahren, so sieht man direkt einen Unterschied durch Hinzunahme der Punkte  $u_{i-1}^m$  und  $u_{i+1}^m$  (vgl. dazu Abb. 5.4). Hier wird das Verhalten der Lösung im bereits bekannten Zeitschritt  $m$  herangezogen.

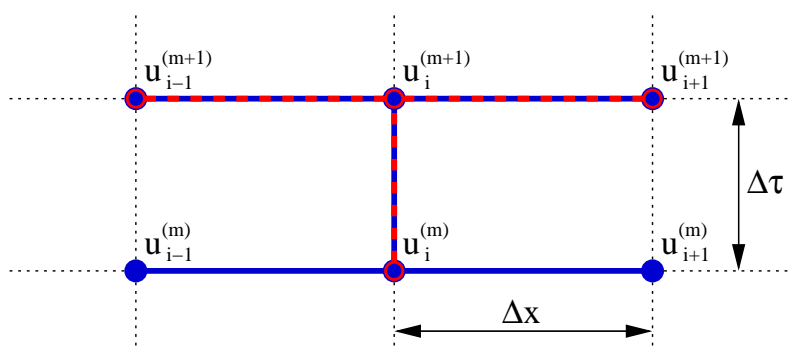


Abbildung 5.4.: Vergleich der benutzten Diskretisierungspunkte des **impliziten Euler-Verfahrens** und des **Crank-Nicolson-Verfahrens**

Liegen jedoch z.B. die Knoten  $u_{i-1}^m$  und  $u_i^m$  auf zwei verschiedenen Seiten der Ausübungsgrenze (vgl. auch Abb. 5.5), so ist nach Lemma 5.2.12 keine  $C^2$ -Differenzierbarkeit mehr gegeben.

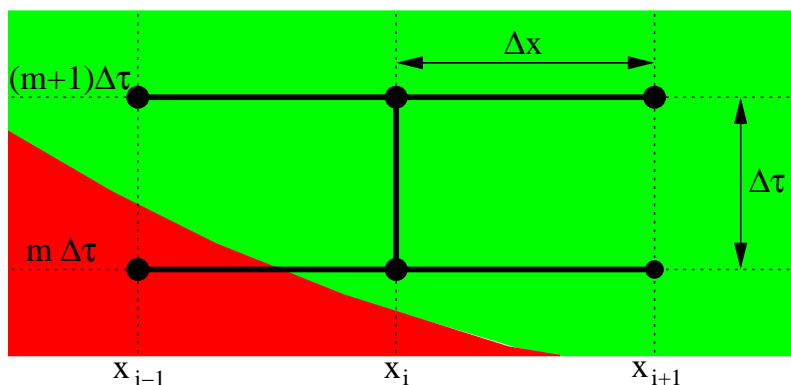


Abbildung 5.5.: Dem Crank-Nicolson-Differenzenstern werden das gestrichelten Gitter der Diskretisierung sowie der **Stoppbereich** und der **Haltebereich** unterlegt.



Insbesondere gilt:

$$\begin{aligned} u_{i-1}^m &\approx v(x_i - \Delta x, m\Delta\tau) \\ &\neq v(x_i, m\Delta\tau) - \Delta x v_x(x_i, m\Delta\tau) + \frac{\Delta x^2}{2} v_{xx}(x_i, m\Delta\tau) \\ &\quad - \frac{\Delta x^3}{6} v_{xxx}(x_i, m\Delta\tau) + \mathcal{O}(\Delta x^4) \end{aligned} \quad (5.5)$$

und damit ebenfalls

$$\frac{u_{i-1}^m - 2u_i^m + u_{i+1}^m}{\Delta x^2} \neq v_{xx}(x_i, m\Delta\tau) + \mathcal{O}(\Delta x^2). \quad (5.6)$$

Nun verläuft die Ausübungsgrenze auch im  $(m+1)$ -ten Zeitschritt zwischen zwei  $x$ -Knoten. Diese Knoten seien ebenfalls mit  $i-1$  und  $i$  indiziert. Demnach gilt dann auch hier:

$$\begin{aligned} &v(x_i - \Delta x, (m+1)\Delta\tau) \\ &\neq v(x_i, (m+1)\Delta\tau) - \Delta x v_x(x_i, (m+1)\Delta\tau) \\ &\quad + \frac{\Delta x^2}{2} v_{xx}(x_i, (m+1)\Delta\tau) - \frac{\Delta x^3}{6} v_{xxx}(x_i, (m+1)\Delta\tau) + \mathcal{O}(\Delta x^4). \end{aligned} \quad (5.7)$$

Entsprechend wird somit die Differentialungleichung auch nicht exakt, im Sinne der Fehlerordnung der Diskretisierung, wiedergegeben. Um dies zu erreichen, müsste die Diskretisierung exakt bis an die Ausübungsgrenze heran geführt werden. Der DHR-Algorithmus bestimmt, so wie andere Verfahren auch, den nächsten Knoten unterhalb der wahren Ausübungsgrenze. Für die Näherung  $u$  ist die PDE in diesem etwas zu groß bestimmten Haltebereich erfüllt. Ebenso wird die Differenzierbarkeitseigenschaft an der genäherten Ausübungsgrenze erfüllt. Zusammenfassend ergibt sich

**Bemerkung 5.2.14. Numerischer Fehler durch diskrete Ausübungsgrenze**

Weil die exakte Ausübungsgrenze nicht genau erfasst wird, liefern sowohl das implizite Euler-Verfahren als auch das Crank-Nicolson-Verfahren eine über den Abbruchfehler hinaus fehlerbehaftete Näherung des Optionswertes. Hierbei zeigen sich die glättenden Eigenschaften des impliziten Euler-Verfahrens, denn die qualitativen Eigenschaften der theoretischen Lösung (z.B. Konvexität) bleiben erhalten.

Bei fehlender Glattheit soll gemäß Korollar 5.2.7 Rannacher smoothing verwendet werden, um die Konvergenzordnung des Crank-Nicolson-Verfahrens zu erhalten. Mit den oben gemachten Beobachtungen ergibt sich dann folgendes

**Korollar 5.2.15. Rannacher smoothing ist in jedem Schritt notwendig**

- i) Aufgrund der fehlenden  $C^2$ -Differenzierbarkeit der Lösung an der Ausübungsgrenze, **muss** in jedem Iterationsschritt Rannacher smoothing angewendet werden, um die Konvergenzordnung von Crank-Nicolson zu erhalten.
- ii) Die Einhaltung von i) bedeutet aber eine Berechnung der Lösung über das **implizite Euler-Verfahren**.

**Beweis:**

- i) Nach Lemma 5.2.12 ist die Wertfunktion an der Ausübungsgrenze nur einmal stetig nach  $x$  differenzierbar. Das Korollar 5.2.7 beschreibt die Notwendigkeit des Rannacher smoothing bei nicht hinreichender Glattheit. Die Aussage folgt dann aus dem Korollar in Verbindung mit dem Lemma.
- ii) Rannacher smoothing bedeutet die Durchführung von zwei impliziten Euler-Schritten. Wobei sich nach i) dem ersten impliziten Schritt bereits ein weiteres Rannacher smoothing anschließen muss. Induktiv folgt die Behauptung.  $\square$

**Bemerkung 5.2.16. Verwendung L-stabiler Verfahren höherer Ordnung**

*Neben dem weitverbreiteten Crank-Nicolson-Verfahren finden auch Mehrschrittverfahren ihre Verwendung, um eine höhere Ordnung in der Zeitdiskretisierung zu erzeugen. Hervorzuheben sind hier insbesondere die BDF-Verfahren, wie sie in [OFG04] verwendet werden. Nach dem Lemma 5.2.5 wird für das Rannacher smoothing ein L-stabiles Verfahren benötigt. Der Einfachheit halber wurde der implizite Euler gewählt. Das BDF2-Verfahren ist ebenfalls L-stabil und damit geeignet.*

*Da jedoch die Startwerte eines n-stufigen Mehrschrittverfahrens mittels eines Einschrittverfahrens erzeugt werden müssen, soll der Fokus auf der Betrachtung von Einschrittverfahren für den, wie schon oben beschrieben, problematischen Iterationsstart liegen.*

## 5.3. Verfahrensaufbau

Bevor wir konkret das Problem der Schrittweitensteuerung betrachten, sollen einige zusätzliche Resultate hergeleitet werden, die in die späteren Überlegungen einfließen werden. Dabei wird insbesondere auf die Kosten des Verfahrens im Sinne von benötigten Rechenoperationen eingegangen. Diese beeinflussen entscheidend die Wahl des verwendeten Verfahrens.

### 5.3.1. Kosten der Berechnung

Die einzelnen Zeitschritte sollen mit dem modifizierten DHR-Algorithmus berechnet werden. Bevor wir uns mit dem Aufwand des Verfahrens auseinander setzen, blicken wir wiederholend auf den modifizierten DHR-Algorithmus zurück. Dazu eignet sich am besten das Flussdiagramm für einen Lösungsschritt aus Abbildung 4.3. Wir werden dabei den Aufwand für die Bedingungsprüfungen ignorieren. Der eigentliche Aufwand für jeden Schritt entsteht durch die Lösung des Gleichungssystems  $(UL)z^{(m)} = b^{(m)}$  sowie die evtl. Durchführung des (invertierten) Updates der UL-Zerlegung (Algorithmen 4.1.2 und 4.2.5). Der Einfachheit halber kann man diese jeweils spaltenweise durchgeführte Matrixzerlegung als eine einzige, aber vollständige Zerlegung auffassen. Ein mehrfaches Durchlaufen der Schleife, bis ein zulässiges  $z^{(m)}$  gefunden wurde, wird im nächsten Abschnitt unter dem Stichwort ‚Wechsel des Grenzknotens behandelt‘. Dass die Matrix  $A$ , für zeitabhängige Parameter sogar in jedem Schritt, neu besetzt werden muss, schlägt sich ebenfalls im Aufwand nieder.

**Bemerkung 5.3.1. Aufwand des Verfahrens**

Der Aufwand des Verfahrens besteht also im Wesentlichen aus folgenden drei Elementen:

- i) Besetzen der Iterationsmatrizen
- ii) Durchführen einer vollständigen UL-Zerlegung
- iii) Lösen des Gleichungssystems  $(UL)z^{(m)} = b^{(m)}$  in jedem Zeitschritt

Dieser Aufwand lässt sich genauer bemessen, wie es der folgende Satz angibt.

**Satz 5.3.2. Berechnungsaufwand**

Das Intervall  $[x_{min}, x_{max}]$  sei in  $N$  Teilintervalle zerlegt worden, d.h. die Diskretisierungsmatrizen haben die Größe  $N \times N$ . Es werde entweder das implizite Euler-Verfahren oder das Crank-Nicolson-Verfahren verwendet. Für die Durchführung eines Zeitschrittes der Bewertung eines amerikanischen Puts mittels des Algorithmus 4.1.1 von Dempster, Hutton und Richards ergibt sich folgender Aufwand:

- i) Für das Besetzen der Iterationsmatrizen ergeben sich

- a) bei Verwendung des impliziten Euler-Verfahrens und

- 1) einer beliebigen Zerlegung des Intervalls  $[x_{min}, x_{max}]$  insgesamt

$11N + 2$  Multiplikationen,  $5N$  Additionen

$$\implies 16N + 2 \text{ Operationen} \quad (5.8a)$$

$11N + 2$  Multiplikationen,  $6N$  Additionen

$$\implies 17N + 2 \text{ Operationen} \quad (5.8b)$$

- 2) einem äquidistanten Gitter

$8N + 4$  Multiplikationen,  $N$  Additionen

$$\implies 9N + 4 \text{ Operationen} \quad (5.9a)$$

$8N + 4$  Multiplikationen,  $2N$  Additionen

$$\implies 10N + 4 \text{ Operationen} \quad (5.9b)$$

- b) bei Verwendung des Crank-Nicolson-Verfahrens und

- 1) einer beliebigen Zerlegung des Intervalls  $[x_{min}, x_{max}]$  insgesamt

$14N + 2$  Multiplikationen,  $5N$  Additionen

$$\implies 19N + 2 \text{ Operationen} \quad (5.10a)$$

$14N + 2$  Multiplikationen,  $6N$  Additionen

$$\implies 20N + 2 \text{ Operationen} \quad (5.10b)$$

- 2) einem äquidistanten Gitter

$11N + 4$  Multiplikationen,  $N$  Additionen

$$\implies 12N + 4 \text{ Operationen} \quad (5.11a)$$

$11N + 4$  Multiplikationen,  $2N$  Additionen

$$\implies 13N + 4 \text{ Operationen} \quad (5.11b)$$

für die Konvektions-Verschiebung bzw. die neue Transformation (Nummer der Gleichung mit a) oder b)).

- ii) Unabhängig von der Transformation, der Wahl des Gitters und dem verwendeten Verfahren sind für die vollständige UL-Zerlegung der Tridiagonalmatrix  $A$

$$2N - 1 \text{ Multiplikationen, } N - 1 \text{ Additionen} \implies 3N - 2 \text{ Operationen} \quad (5.12)$$

durchzuführen. Dies entspricht dem Extremfall  $nb = 0$ .

- iii) Ebenfalls unabhängig vom gewählten Gitter und der gewählten Transformation ergibt sich für die Lösung des Gleichungssystems  $(UL)z^{(m)} = b^{(m)}$  für die Verwendung

- a) des impliziten Euler-Verfahrens ein Aufwand von

$$6N - 2 \text{ Multiplikationen, } 5N - 2 \text{ Additionen} \implies 11N - 4 \text{ Operationen.} \quad (5.13)$$

- b) des Crank-Nicolson-Verfahrens ein Aufwand von

$$9N - 2 \text{ Multiplikationen, } 7N - 2 \text{ Additionen} \implies 16N - 4 \text{ Operationen.} \quad (5.14)$$

**Beweis:**

Der Beweis befindet sich im Anhang A. □

**Bemerkung 5.3.3. Aufwand für Neubesetzung entspricht Zeitschritt**

Aus den Teilen i) und iii) des Satzes 5.3.2 ergibt sich sofort, dass eine Neubesetzung der Matrizen in etwa genauso teuer ist wie die Durchführung eines Zeitschrittes.

**5.3.2. Zusätzlicher Aufwand - Wechsel des Grenzknotens**

Unter dem Schlagwort ‚zusätzlicher Aufwand‘ laufen die Anteile der Berechnung, welche nicht in jedem Zeitschritt auftauchen bzw. die aufgrund der Zeitschrittweitensteuerung auftreten. Mit dem letzten Punkt ist im Wesentlichen die Schätzung des Diskretisierungsfehlers gemeint. Hier gilt folgende Bemerkung:

**Bemerkung 5.3.4. Aufwand für Fehlerschätzung**

Sofern die Größe des Zeitschrittes  $\Delta\tau$  gesteuert werden soll, so ist ein Instrument hierzu der Diskretisierungsfehler. Für dessen Schätzung werden zwei alternative Lösungen benötigt. Der Aufwand für die Berechnung dieser beiden Lösungen wird den jeweils verwendeten Verfahren zugeordnet. Der Aufwand, etwa  $N$  Operationen, für die eigentliche Schätzung über die absolute Differenz der beiden Lösungen wird nicht beachtet, da er bei allen Steuerungsverfahren auftritt.

Der erste Punkt, Teile des Algorithmus, die nicht in jedem Zeitschritt greifen, bezieht sich auf die nicht erfüllte Bedingungsprüfung  $z^{(m)} \geq 0$ . Diese ist eng mit dem Verlauf der Ausübungsgrenze verbunden. Wie schon mehrfach bemerkt, liegt eine der großen Schwierigkeiten, die Option zu bewerten, in der impliziten Bestimmung der Ausübungsgrenze  $x_f(\tau)$ .

**Beobachtung 5.3.5. Auffinden der numerischen Ausübungsgrenze**

Der numerisch mittels des (modifizierten) DHR-Algorithmus bestimmte Näherungswert für die Ausübungsgrenze ist der Gitterpunkt  $x_{nb}$  unmittelbar unterhalb von dieser Grenze. Für den  $m$ -ten Zeitschritt lassen sich die beiden Fälle

$$nb^{(m)} = nb^{(m-1)} \quad (5.15a)$$

und

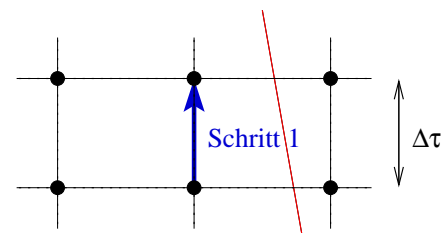
$$nb^{(m)} \neq nb^{(m-1)} \quad (5.15b)$$

unterscheiden. Im zweiten Fall muss der Wert für den neuen Zeitschritt durch den Algorithmus angepasst werden. Die Anpassung erfolgt, nachdem ein nicht zulässiges  $z^{(m)}$  berechnet wurde. Vergleicht man mit dem Flussdiagramm Abbildung 4.3, so stellt man fest, dass das Gleichungssystem  $(UL)z^{(m)} = b^{(m)}$  mindestens ein weiteres Mal gelöst werden musste.

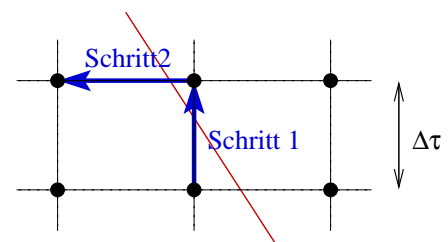
Die Abbildung 5.6 stellt die Aussage der Beobachtung 5.3.5 für beide Situationen noch einmal schematisch da.

Im Teil a) sieht man den angenommenen Normalfall. Zu beiden Iterationszeitpunkten liegt die Ausübungsgrenze zwischen den beiden gleichen Gitterpunkten. Mit Schritt 1 ist die Lösung des Systems  $(UL)z^{(m)} = b^{(m)}$  bezeichnet. Es wird ein zulässiges  $z^{(m)}$  als Ergebnis geliefert und der Zeitschritt ist beendet.

Im Teil b) hingegen führt dieser erste Schritt, weil die Ausübungsgrenze nun nicht mehr zwischen den aus dem  $(m-1)$ -ten Zeitschritt bekannten Knoten liegt, zu einer nicht zulässigen Lösung  $z^{(m,1)}$ . Es muss Schritt 2 als weiterer Schritt durchgeführt werden, um ein zulässiges  $z^{(m,2)}$  zu finden. Hier ist die Verringerung des Wertes  $nb$ , verbunden mit dem Update des DHR-Algorithmus, dargestellt. Der Aufwand für die im Update enthaltene Änderung der Matrixzerlegung ist bereits berücksichtigt worden. Allerdings muss die damit verbundene erneute Lösung des Gleichungssystems beachtet werden. Diese führt gemäß Satz 5.3.2 zu einem Mehraufwand von  $11N - 4$  Operationen für das implizite Euler-Verfahren bzw. von  $16N - 4$  Operationen für das Crank-Nicolson-Verfahren. Eine weitergehende Aussage zum Aufwand ergänzen wir mit



(a) kein Wechsel



(b) Wechsel des Grenzknoten

Abbildung 5.6.: Schematischer Ablauf des Wechsels des Gitterknotens. Dem Gitter unterlegt ist der Verlauf der **Ausübungsgrenze**.

**Bemerkung 5.3.6. Aufwand für den Wechsel des Grenzknotens**

i) Für die Anpassung des Grenzknotens muss (mindestens) ein weiteres Gleichungssystem der Form  $(UL)z^{(m)} = b^{(m)}$  gelöst werden. Der Aufwand dafür ist exakt der gleiche wie bei der Durchführung eines regulären Zeitschrittes ( Fall a). Dies sind

$$\begin{aligned} 11N - 4 & \quad \text{Operationen für das implizite Euler-Verfahren} \\ \text{bzw. } 16N - 4 & \quad \text{Operationen für das Crank-Nicolson-Verfahren.} \end{aligned}$$

ii) Bei einem in  $x$ -Richtung hinreichend feinen Gitter ist es möglich, dass der mit  $nb$  indizierte Knoten zumindest für kleines  $\tau$  mehr als einen Gitterpunkt springt. In diesen Fällen ist jedesmal ein zu Schritt 2 analoges Vorgehen erforderlich. Der Aufwand erhöht sich entsprechend.

**5.3.3. Wahl der Startschrittweite**

Die im Folgenden besprochenen Verfahren sind selbstkorrigierend im Sinne einer automatischen Schrittweitenanpassung, dennoch sollte die Wahl einer geeigneten Startschrittweite untersucht werden. Insbesondere wenn man zur Minimierung des Aufwands versucht, ein implizites Euler-Verfahren mit konstanter Schrittweite zu verwenden, so kommt man um diese Wahl nicht herum.

Verwendet man eine Semidiskretisierung in  $x$ -Richtung der mittels Konvektions-Verschiebung

$$v_\tau = \frac{\sigma^2}{2} v_{xx} \tag{5.16a}$$

bzw. neuer Transformation

$$v_\tau = \frac{\sigma^2}{2} \left[ v_{xx} - \frac{1}{4} v \right] \tag{5.16b}$$

transformierten Black-Scholes Gleichung, so erhält man die Gleichungen

$$\frac{\partial u_i}{\partial \tau} = \frac{\sigma^2}{2} \frac{1}{\Delta x^2} (u_{i-1} - 2u_i + u_{i+1}); \quad i = 1, \dots, N, \tag{5.17a}$$

$$\frac{\partial u_i}{\partial \tau} = \frac{\sigma^2}{2} \left[ \frac{1}{\Delta x^2} (u_{i-1} - 2u_i + u_{i+1}) - \frac{1}{4} u_i \right]; \quad i = 1, \dots, N. \tag{5.17b}$$

Diese lassen sich beide als ein System gewöhnlicher Differentialgleichungen der Form

$$w' = f(\tau, w) \tag{5.18}$$

auffassen. In die Funktion  $f$  gehen dabei auch die Parameter  $\sigma$  und  $\Delta x$  ein.

Auf das System 5.18 kann nun die Theorie von Gladwell, Shampine und Brankin [GSB87] zur automatischen Wahl einer geeigneten Startschrittweite angewendet werden. Die Autoren weisen dabei ausdrücklich darauf hin, auch bei steifen Differentialgleichungssystemen zufriedenstellende Resultate produzieren zu können.

Wir beziehen uns dabei auf die Ausführung des dort skizzierten Algorithmus, wie er in dem Buch von Hairer et al. [HNW91] zu finden ist. Mit  $\|a\|$  sei eine gewichtete Norm bezeichnet. In der Arbeit von Gladwell et al. wird ebenso wie in der ersten Auflage des Buches von Hairer et al. die Maximumsnorm gewählt. In ihrer zweiten Auflage gehen sie jedoch zur Euklidischen Norm über. Aussagen zur Wahl der Norm für das Komplementaritätsproblem erfolgen in der Bemerkung 7.1.1. Zuvor formulieren wir den Algorithmus für ein Verfahren erster Ordnung, ohne die Norm zu spezifizieren.

**Algorithmus 5.3.7. Automatische Startschrittweite**  
[HNW91, Kapitel II.4, S. 169]

- i) Man werte  $f(0, w_0)$  aus und setze  $d_0 := \|w_0\|$  sowie  $d_1 := \|f(0, w_0)\|$ .  
ii) Definiere die Testschrittweite

$$h_0 := \frac{1}{100} \frac{d_0}{d_1},$$

so dass der Zuwachs eines expliziten Euler-Schrittes klein im Vergleich zu den Startwerten ist. Sollte  $d_0$  oder  $d_1$  kleiner als  $10^{-5}$  sein, setze  $h_0 := 10^{-6}$ .

- iii) Man führe einen expliziten Euler-Schritt durch.  $w_1 := w_0 + h_0 f(0, w_0)$ . Danach berechne man  $f(h_0, w_1)$ .  
iv)  $d_2 := \frac{\|f(h_0, w_1) - f(0, w_0)\|}{h_0}$  liefert eine Approximation der zweiten Ableitung.  
v) Man berechne eine weitere Schrittweite

$$h_1 := \frac{1}{10} \sqrt{\frac{1}{\max(d_1, d_2)}}.$$

Für  $\max(d_1, d_2) < 10^{-15}$  setze man  $h_1 := \max(10^{-6}, h_0 \cdot 10^{-3})$ .

- vi) Die geeignete Startschrittweite ergibt sich zu

$$h_{start} := \min(100 \cdot h_0, h_1).$$

Wir wollen diesen Algorithmus auf das System (5.18) anwenden. Es ergibt sich das folgende

**Lemma 5.3.8. Approximation der Startschrittweite**

Es sei ein Put zu bewerten. Die vorgegebene Toleranz sei  $tol$  und es gelte  $2\Delta x < \sigma$ . Dann ergibt sich mittels des Algorithmus 5.3.7 die Startschrittweite:

$$i) \quad h_{start} := \frac{2\sqrt{tol}\Delta x^2}{10\sqrt{K}\sigma^2\sqrt{(1-e^{-\Delta x})(3-e^{-\Delta x})}}$$

für die mittels Konvektions-Verschiebung transformierte Black-Scholes Gleichung

$$ii) \quad h_{start} := \frac{2\sqrt{tol}\Delta x}{10\sqrt{K}\sigma^2\sqrt{\sinh(\frac{\Delta x}{2})\left(1 + \frac{4(2-\cosh(\frac{\Delta x}{2}))}{\Delta x^2}\right)}}$$

für die mittels neuer Transformation transformierte Black-Scholes Gleichung

**Beweis:**

Für die rechte Seite des Systems (5.18) gilt  $f(\tau, w) = \frac{\sigma^2}{2}Dw$ ; mit einer Tridiagonalmatrix  $D$ . Mittels des Algorithmus 5.3.7 sind nun in beiden Fällen

$$\begin{aligned} d_0 &:= \|w_0\|, \\ d_1 &:= \|f(0, w_0)\| = \frac{\sigma^2}{2} \|Dw_0\|, \\ d_2 &:= \|f(h_0, w_1) - f(0, w_0)\|/h_0 = \frac{\sigma^4}{4} \|D^2w_0\| \end{aligned}$$

zu bestimmen. Für die Norm wählen wir der Einfachheit halber die gewichtete Maximumnorm mit der vorgegebenen Toleranz  $tol$  als Gewicht.

i) Für den Put gilt unter der Konvektions-Verschiebung

$$w_0 = \phi(x, 0) = K \max(1 - \exp(x), 0).$$

Aus der Monotonie von  $\exp$  folgt sofort  $d_0 = \frac{K}{tol}(1 - \exp(x_{min}))$ . Weiter gilt:

$$(Dw_0)_i = \frac{K}{\Delta x^2} \begin{cases} 2 \exp(x_i)(1 - \cosh(\Delta x)) & ; x_i < 0 \\ 1 - r^{-\Delta x} & ; x_i = 0 \\ 0 & ; x_i > 0 \end{cases}$$

und damit  $d_1 = \frac{K}{tol} \frac{\sigma^2}{2\Delta x^2} (1 - \exp(-\Delta x))$ . Aus

$$(D^2w_0)_i = \frac{K}{\Delta x^4} \begin{cases} -4 \exp(x_i) (\cosh(\frac{\Delta x}{2}) - 1)^2 & ; x_i \leq -2\Delta x \\ (1 - e^{-\Delta x})(3 + e^{-\Delta x}(e^{-\Delta x} - 3)) & ; x_i = -\Delta x \\ (1 - e^{-\Delta x})(e^{-\Delta x} - 3) & ; x_i = 0 \\ (1 - e^{-\Delta x}) & ; x_i = \Delta x \\ 0 & ; x_i \geq 2\Delta x \end{cases}$$

folgt  $d_2 = \frac{K}{tol} \frac{\sigma^4}{4\Delta x^4} (1 - e^{-\Delta x})(3 - e^{-\Delta x})$ .



ii) Für den Put gilt unter der neuen Transformation:

$$w_0 = \phi(x, 0) = 2K \max(-\sinh(x/2), 0).$$

Aus der Monotonie von  $\sinh$  folgt sofort  $d_0 = \frac{2K}{tol} \sinh(|x_{min}|/2)$ . Weiter gilt:

$$(Dw_0)_i = 2K \begin{cases} \sinh(\frac{x_i}{2}) \left( \frac{2(1-\cosh(\frac{\Delta x}{2}))}{\Delta x^2} + \frac{1}{4} \right) & ; x_i < 0 \\ \sinh(\frac{\Delta x}{2}) \frac{1}{\Delta x^2} & ; x_i = 0 \\ 0 & ; x_i > 0 \end{cases}$$

und damit  $d_1 = \frac{2K}{tol} \frac{\sigma^2}{2\Delta x^2} \sinh(\frac{\Delta x}{2})$ . Aus

$$(D^2w_0)_i = 2K \begin{cases} -\sinh(\frac{x_i}{2}) \left( \frac{2(1-\cosh(\frac{\Delta x}{2}))}{\Delta x^2} + \frac{1}{4} \right)^2 & ; x_i \leq -2\Delta x \\ \sinh(\frac{\Delta x}{2}) \left( \frac{4(\cosh(\frac{\Delta x}{2})-1)^2+1}{\Delta x^4} + \frac{\cosh(\frac{\Delta x}{2})-1}{\Delta x^2} - \frac{1}{16} \right) & ; x_i = -\Delta x \\ \sinh(\frac{\Delta x}{2}) \left( \frac{2(\cosh(\frac{\Delta x}{2})-2)}{\Delta x^4} - \frac{1}{2\Delta x^2} \right) & ; x_i = 0 \\ \sinh(\frac{\Delta x}{2}) \frac{1}{\Delta x^4} & ; x_i = \Delta x \\ 0 & ; x_i \geq 2\Delta x \end{cases}$$

$$\text{folgt } d_2 = \frac{2K}{tol} \frac{\sigma^4}{8\Delta x^2} \sinh(\frac{\Delta x}{2}) \left( 1 + \frac{4(2-\cosh(\frac{\Delta x}{2}))}{\Delta x^2} \right).$$

In beiden Fällen sind die Werte von  $d_0$  und  $d_1$  echt größer als  $10^{-6}$ . Der Wert von  $h_{start}$  lässt sich dann mittels des Algorithmus berechnen.

$$\begin{aligned} h_{start} &:= \min(100 \cdot h_0, h_1) \\ &= \min\left(\frac{d_0}{d_1}, \frac{1}{10} \sqrt{\frac{1}{\max(d_1, d_2)}}\right) \\ &= \frac{1}{10} \min\left(\frac{10d_0}{d_1}, \sqrt{\frac{1}{d_1}}, \sqrt{\frac{1}{d_2}}\right) \end{aligned}$$

Dabei ist  $\sqrt{\frac{1}{d_2}}$  für  $2\Delta x < \sigma$  der kleinste Wert und es folgt die Behauptung.  $\square$

Auch hier lässt sich im Beweis die Abhängigkeit der Bedingung von der Knickstelle in den Anfangswerten ablesen. Die jeweiligen Maxima der Normen  $d_1$  und  $d_2$  liegen genau an der Stelle  $x = 0$ . Mit dieser Approximation lässt sich eine grobe Aussage zu der benötigten Anzahl der Zeitschritte machen:

**Bemerkung 5.3.9. Ungefähre Anzahl der benötigten Zeitschritte**

Aus dem Ratschlag für die Startschrittweite ergibt sich, wenn diese Schrittweite konstant verwendet wird, über eine grobe Approximation der obigen Werte mittels

$$h_{start} \approx \frac{\sqrt{tol}\Delta x^{3/2}}{5\sqrt{K}\sigma^2}$$

schnell die Notwendigkeit von weit **mehr** als 1000 Zeitschritten! Dabei werden große Werte von  $tol = \Delta x = \frac{1}{100}$  und moderate Parameter  $K = 1$ ,  $T = 1$ ,  $\sigma = 0.3$  zugrunde gelegt.



## 6. Verfahren zur Schrittweitensteuerung

Vor der Wahl eines geeigneten Verfahrens für die Steuerung der Zeitschrittweite sollen die Erfolg versprechenden Kandidaten beschrieben werden. Nach der Betrachtung der Vor- und Nachteile der einzelnen Methoden wird, ausgehend von einer kurzen Übersicht, das geeignetste Verfahren ausgewählt.

### 6.1. Mögliche Verfahren

Die möglichen Kandidaten lassen sich im Wesentlichen in vier Klassen einteilen. Dies sind als erstes die Methoden, für die  $\Delta\tau$  konstant bleibt. Die zweite Klasse bilden die Verfahren, die  $\Delta\tau$  über den Wertzuwachs steuern. Im nachfolgenden werden diese auch als Algorithmus von Eriksson und Johnson bzw. kurz ‚EJ-Algorithmus‘ bezeichnet. Die dritte Klasse umfasst Methoden mit einer kanonischen Steuerung über den Vergleich zweier Lösungen von Verfahren unterschiedlicher Ordnung. Den letzten Punkt der Liste stellen Verfahren mit einer Steuerung über den Vergleich zweier Lösungen mit unterschiedlicher Schrittlänge dar.

#### 6.1.1. Impliziter Euler mit konstanter Schrittweite

Die einfachste Form, die Schrittweite zu ‚steuern‘ ist sehr naheliegend. Es wird eine Startschrittweite  $\Delta\tau_0$  gewählt und für alle weiteren Schritte beibehalten. Das hierfür verwendete Verfahren sollte eine rein implizite Berechnung durchführen, denn Korollar 5.2.15 sowie Bemerkung 5.2.16 fordern ein L-stabiles implizites Verfahren. Als Einschrittverfahren bietet sich hier das implizite Euler-Verfahren an.

#### 6.1.2. Steuerung über den Wertzuwachs der Lösung

Eine weitere einfache Steuerungsmöglichkeit ist das hier als ‚EJ-Algorithmus‘ bezeichnete Vorgehen. Dieses Verfahren wurde erstmalig in einer Arbeit von Eriksson und Johnson [EJ87] vorgestellt und von Johnson, Nie und Thomée [JNT90] auf eine Finite-Differenzen-Diskretisierung einer eindimensionalen Diffusionsgleichung angewendet.

Wir skizzieren im Folgenden kurz die Idee des Algorithmus, wie er im Abschnitt 2 der Arbeit von Eriksson und Johnson [EJ87, S. 14f] formuliert wird. Dazu wird die dort benutzte Notation auf die hier für die Optionsbewertung verwendete Notation übertragen. Es sei

$$0 = \tau_0 < \tau_1 < \dots < \tau_m < \dots$$

eine (nicht notwendiger Weise äquidistante) Zerlegung der Restlaufzeit in Intervalle  $I_m := ]\tau_{m-1}, \tau_m]$ . Des Weiteren sei  $tol$  eine vorgegebene Toleranzschranke und die diskrete Approximation  $u^{(m)}$  an der Stelle  $\tau_m$  des transformierten Optionswertes  $v$  soll

$$\max_{\tau \leq \tau_M} \|v(\tau) - u^{(m)}\| \leq tol$$

erfüllen. Mithilfe einer Abschätzung für diesen Fehler zeigen Eriksson und Johnson: Die Zeitschritte  $\Delta\tau_m$  für  $m = 1, \dots, M$  müssen so gewählt werden, dass

$$C\Delta\tau_m \max_{\tau \in I_M} \|v_\tau(\tau)\| \approx tol$$

erfüllt ist. Ersetzt man nun an dieser Stelle die partielle Ableitung durch den entsprechenden Differenzenquotienten für die diskrete Lösung, so erhält man die Bedingung

$$\|u^{(m)} - u^{(m-1)}\| \approx \frac{tol}{C}.$$

Dabei bezeichnet  $C$  eine Konstante, die die kontinuierliche Lösung sowie deren Zuwächse beschränkt (vgl. [EJ87, Theorem 1]). Das Ziel wird es sein, die Zuwächse der Lösung bei den einzelnen Schritten in derselben Größenordnung zu halten.

Unter der Annahme, dass  $u^{(m-1)}$  bereits berechnet ist, ergibt sich für geeignetes  $\gamma$  der

**Algorithmus 6.1.1. Algorithmus von Eriksson und Johnson (EJ-Algorithmus, Grundform)** [EJ87, Gleichungen (2.4a)-(2.4c), S. 15]

- i) Man wähle  $\Delta\tau_m = \Delta\tau_{m-1}$ .
- ii) Mit diesem  $\Delta\tau_m$  wird über das implizite Euler-Verfahren die diskrete Lösung  $u^{(m)}$  bestimmt.
- iii) Falls

$$\frac{tol}{\gamma C} \leq \|u^{(m)} - u^{(m-1)}\| \leq \frac{tol}{C}$$

erfüllt ist, wird die Schrittweite  $\Delta\tau_m$  und die Lösung  $u^{(m)}$  akzeptiert. Ansonsten wird  $\Delta\tau_m$  um den Faktor 2 erhöht oder erniedrigt und mit Schritt ii) fortgefahren.

Eriksson und Johnson geben dabei  $\gamma$  mit Wert 2 oder 3 als geeignet an.

Für das Diffusionsproblem wandeln Johnson et al.. [JNT90] diesen Algorithmus leicht ab. Hier soll die neue Schrittweite nicht nur um Zweierpotenzen variieren, sondern direkt vom letzten Zuwachs der diskreten Lösung abhängen. Auch hier seien  $u^{(m-1)}$  und  $\Delta\tau_m$  als bekannt angenommen. Es folgt

**Algorithmus 6.1.2. EJ-Algorithmus (1-D Diffusion)**

[JNT90, Abschnitt 4, S. 287]

- i) Man bestimme über das implizite Euler-Verfahren die diskrete Lösung  $u^{(m)}$ .
- ii) Falls

$$\|u^{(m)} - u^{(m-1)}\| < tol,$$

akzeptiere man den Wert von  $u^{(m)}$  und wähle

$$\Delta\tau_{m+1} = \Delta\tau_m \frac{tol}{\|u^{(m)} - u^{(m-1)}\|}. \quad (6.1)$$

Andernfalls wird  $\Delta\tau_m$  durch  $\Delta\tau_m/2$  ersetzt und Schritt i) wiederholt.

Die Autoren haben diesen Algorithmus an gleicher Stelle für das Diffusionsproblem mit unterschiedlich glatten Anfangswerten erfolgreich getestet.

Im Finanzkontext wird dieses Verfahren bereits von Forsyth und Vetzal [FV02] angewendet. Dabei wird im Wesentlichen die Struktur des Algorithmus 6.1.2 verwendet, lediglich die Vorschrift zur Wahl des nächsten Zeitschrittes ist angepasst worden. Anstatt der Norm des punktwisen Zuwachses wird hier eine wie in der nachfolgenden Gleichung skalierte Form des Zuwachses verwendet.  $sc$  wird benutzt, um viele sehr kleine Schritte bei gleichzeitig kleinen Optionswerten zu vermeiden. Aus der Gleichung (6.1) wird dann:

$$\Delta\tau_{m+1} = \Delta\tau_m \left( \min_i \frac{tol}{\frac{|u_i^{(m)} - u_i^{(m-1)}|}{\max(sc, |u_i^{(m)}|, |u_i^{(m-1)}|)}} \right). \quad (6.2)$$

Man verhindert so ein langsames Wachstum der Zeitschrittweite  $\Delta\tau_m$  für große Kurswerte.

**Bemerkung 6.1.3. Koordinaten der Berechnung**

In unserem Fall finden sämtliche Berechnungen in dem auf  $x$ -Koordinaten transformiertem Koordinatensystem statt. Im Gegensatz zu Forsyth und Vetzal, die auf dem regulären Kursgitter, d.h. auf  $S$ -Koordinaten, rechnen. Insbesondere werden dadurch sehr große Werte für  $x$ , welche eine Abwandlung des Algorithmus 6.1.2 durch die Gleichung (6.2) notwendig machen, vermieden.

In der gleichen Arbeit liefern Forsyth und Vetzal eine Analyse des globalen Fehlers für das Crank-Nicolson-Verfahren mit und ohne Schrittweitensteuerung.

**Beobachtung 6.1.4. Tatsächliche Fehlerordnung bei Crank-Nicolson**

[FV02, Abschnitt 9, S. 2109f.]

Unter der Annahme, dass sich der globale Fehler als Summe der lokalen Fehler ergibt und dass der lokale Fehler sich wiederum global so verhält wie zur Fälligkeit der Option, gilt:

- i) Für die Iteration von  $1/\Delta\tau$  Schritten mit dem Crank-Nicolson-Verfahren und einer konstanten Schrittweite  $\Delta\tau$  hat der globale Fehler die ungefähre Ordnung  $\mathcal{O}(\Delta\tau^{3/2})$ .
- ii) Für die Iteration mittels des Crank-Nicolson-Verfahrens mit einer über die Bedingung

$$\|u^{(m)} - u^{(m-1)}\| \approx \text{tol}$$

gesteuerten Schrittweite, d.h.  $\mathcal{O}(1/\text{tol})$  Zeitschritten, hat der globale Fehler die ungefähre Ordnung  $\mathcal{O}(\text{tol}^2)$ .

Das erwartete Verhalten der Fehlerordnung für das Crank-Nicolson-Verfahren lässt sich durch numerische Berechnungen unterstützen (ebenfalls im Abschnitt 9 von [FV02]). Allerdings werden hier so viele Annahmen gemacht, dass man sich der nachfolgenden Bemerkung der Autoren anschließen muss. Diese unterteilen wir bewusst in zwei Teile.

**Bemerkung 6.1.5. Genauigkeitsanspruch** [FV02, S. 2110]

- i) “We make **no** claim that the above analysis of the time truncation error is in **any** way precise...”<sup>1</sup>; d.h. man kann sich auf die Ordnungsergebnisse aus der Beobachtung 6.1.4 nicht notwendigerweise verlassen.
- ii) “... **but** only suggestive of an appropriate timestepping strategy.”<sup>2</sup>; die Aussage reicht allerdings aus, um die Verwendbarkeit des Verfahrens zu rechtfertigen.

Bei ihren numerischen Betrachtungen stoßen Forsyth und Vetzal ebenfalls auf die für das Crank-Nicolson-Verfahren typischen Oszillationen. Wir entscheiden uns, unter Berücksichtigung des Korollars 5.2.15, dafür den Algorithmus von Johnson, Nie und Thomée (Algorithmus 6.1.2) zusammen mit dem impliziten Euler-Verfahren zu verwenden.

### 6.1.3. Vergleich mit Verfahren höherer Ordnung

Eine Standardmethode für die Schrittweitensteuerung ist der Vergleich der Lösungen zweier numerischer Verfahren mit unterschiedlicher Fehlerordnung. Für die numerische Lösung gewöhnlicher Differentialgleichungen gibt es hier als bekannteste Vertreter die auf einer Idee von Fehlberg [Feh64] beruhende Klasse von eingebetteten Runge-Kutta-Verfahren oder auch Runge-Kutta-Fehlberg-Verfahren. Es liegt zwar keine gewöhnliche Differentialgleichung vor, allerdings haben wir uns bei der Bestimmung der Startschrittweite schon einmal der Semidiskretisierung bedient, um das Problem der Steuerung der

---

<sup>1</sup>Fettdruck fehlt im Originaltext

<sup>2</sup>s.o.

Zeitschrittweiten auf ein ODE-System zurückzuführen.

Wir betrachten eine Semidiskretisierung der PDEs in der Variable  $x$ . Zugrunde gelegt wird dafür die Black-Scholes Gleichung nach Transformation über die Konvektions-Verschiebung

$$v_\tau = \frac{\sigma^2}{2} v_{xx} \quad (6.3a)$$

bzw. die neue Transformation

$$v_\tau = \frac{\sigma^2}{2} \left[ v_{xx} - \frac{1}{4} v \right]. \quad (6.3b)$$

Dies ist eine Vorgehensweise, die u.a. auch Ikonen und Toivanen [IT04] verwenden, um zwischen Raum- und Zeitdiskretisierung zu trennen. Wir unterstellen dabei wieder den Fall, dass Gleichung a) gilt. Für das Vorgehen würde sich auch im Falle b) nichts ändern, lediglich die Gestalt der Funktion  $f$  wäre eine andere. Der Differentialoperator  $v_{xx}$  wird mittels Finiter-Differenzen diskretisiert. Es resultiert das System gewöhnlicher Differentialgleichungen

$$w'(\tau) = f(\tau, w) + \mathcal{O}(\Delta x^2). \quad (6.4)$$

Vernachlässigt man die Fehlerterme der Diskretisierung nach  $x$ , so handelt es sich um ein zeitabhängiges System gewöhnlicher Differentialgleichungen der Form

$$w'(\tau) = f(\tau, w). \quad (6.5)$$

Dieses soll seinerseits numerisch gelöst werden, wobei Interesse an einer Steuerung der Zeitschrittweite  $\Delta\tau$  besteht.

Um dieses Ziel zu erreichen, schlagen wir den Weg ein, der sich für die numerische Lösung von ODEs durchgesetzt hat (vgl. z.B. [SB00, Kap. 7.2.5]). Es seien  $F_I(\tau, \Delta\tau, w^{(m)}, w^{(m+1)})$  und  $F_{II}(\tau, \Delta\tau, w^{(m)}, w^{(m+1)})$  zwei verschiedene Diskretisierungen von (6.5), so dass für  $w^{(m)} := w(\tau)$  gilt:

$$w(\tau + \Delta\tau) - [w^{(m)} + \Delta\tau F_I(\tau, \Delta\tau, w^{(m)}, w^{(m+1)})] \doteq C_{F_I} \Delta\tau^p \quad (6.6)$$

$$w(\tau + \Delta\tau) - [w^{(m)} + \Delta\tau F_{II}(\tau, \Delta\tau, w^{(m)}, w^{(m+1)})] \doteq C_{F_{II}} \Delta\tau^{p+1} \quad (6.7)$$

Weiter sei  $w_I^{(m+1)}$  die mittels der Diskretisierung  $F_I(\tau, \Delta\tau, w^{(m)}, w^{(m+1)})$  bzw.  $w_{II}^{(m+1)}$  die mittels  $F_{II}(\tau, \Delta\tau, w^{(m)}, w^{(m+1)})$  berechnete Näherung von  $w(\tau + \Delta\tau)$ . Dann folgt

$$w_I^{(m+1)} - w_{II}^{(m+1)} \doteq C_{F_I} \Delta\tau^{p+1}. \quad (6.8)$$

Das Ziel ist es,  $|w_I^{(m+1)} - w_{II}^{(m+1)}|$  unter eine geforderte Toleranzschranke  $tol$  zu drücken. Aus (6.8) gewinnt man dann eine Näherung für die Konstante  $C_{F_I}$ . Es folgt

$$|C_{F_I}^{(m)}| \approx \frac{|w_I^{(m+1)} - w_{II}^{(m+1)}|}{|\Delta\tau_m|^{p+1}}. \quad (6.9)$$

Gilt nun  $|C_{F_I}^{(m)}| \approx |C_{F_I}^{(m+1)}|$ , so lässt sich eine Näherung für die neue Schrittweite  $\Delta\tau_m$  der Form

$$\Delta\tau_{m+1} = \Delta\tau_m \left( \frac{tol}{\|w_I^{(m+1)} - w_{II}^{(m+1)}\|} \right)^{\frac{1}{p+1}} \quad (6.10)$$

formulieren.

## Augenscheinliche Kandidaten

Die Forderung nach zwei aufeinanderfolgenden Ordnungsstufen  $p$  und  $p + 1$  liefert sofort eine mögliche Kombinationen der Verfahren. Naheliegend ist der Vergleich der Lösung des impliziten Euler-Verfahrens mit einer aus dem Crank-Nicolson-Verfahren stammenden zweiten Approximation. In der gegenüber oben leicht abgewandelten Schreibweise gilt hier:

$$v(x, \tau + \Delta\tau) - [u_k^{(m)} + \Delta\tau F_{impl.}(x, \tau, \Delta\tau, \Delta x, u^{(m)}, u^{(m+1)})] \doteq C_{F_{impl.}} \Delta\tau^1 + \mathcal{O}(\Delta x^2) \quad (6.11)$$

$$v(x, \tau + \Delta\tau) - [u_k^{(m)} + \Delta\tau F_{CN}(x, \tau, \Delta\tau, \Delta x, u^{(m)}, u^{(m+1)})] \doteq C_{F_{CN}} \Delta\tau^2 + \mathcal{O}(\Delta x^2) \quad (6.12)$$

Die Vernachlässigung des Diskretisierungsfehlers in der  $x$ -Komponente lässt sich dabei als Übergang zum impliziten Euler-Verfahren bzw. zur impliziten Trapezregel für das System gewöhnlicher Differentialgleichungen interpretieren. Es lässt sich nun der folgende Algorithmus formulieren:

### Algorithmus 6.1.6. *Implizit/Crank-Nicolson Steuerung*

i) Man wähle

$$\Delta\tau_{m+1} = \Delta\tau_m \sqrt{\frac{tol}{\|u_{impl.}^{(m)} - u_{CN}^{(m)}\|}}. \quad (6.13)$$

Für  $m = 0$  sei  $\Delta\tau_1 := \Delta\tau_{start}$  vorgegeben.

ii) Mit diesem  $\Delta\tau_{m+1}$  wird sowohl über das implizite Euler-Verfahren  $u_{impl.}^{(m+1)}$  als auch mittels Crank-Nicolson  $u_{CN}^{(m+1)}$  bestimmt.

iii) Falls  $\|u_{impl.}^{(m+1)} - u_{CN}^{(m+1)}\| < tol$  erfüllt ist, weiter mit Schritt i). Andernfalls führe man die Anpassung  $\Delta\tau_{m+1} := \gamma\Delta\tau_{m+1}$  mit geeignetem  $0 < \gamma < 1$  durch. Danach wiederhole man Schritt ii).



Für die Verwendung dieses Algorithmus für das Komplementaritätsproblem ist jedoch Vorsicht geboten. Dies sei im folgenden Korollar angemerkt:

**Korollar 6.1.7. *Verlässlichkeit der Implizit/Crank-Nicolson Steuerung***

*Auch wenn sich für das implizite Euler-Verfahren ein Abbruchfehler der Ordnung  $\mathcal{O}((\Delta\tau)) + \mathcal{O}((\Delta x)^2)$  und für das Crank-Nicolson-Verfahren ein theoretischer Abbruchfehler der Ordnung  $\mathcal{O}((\Delta\tau)^2) + \mathcal{O}((\Delta x)^2)$  ergibt, so ist dieses Verfahrenspaar zu Steuerung nur bedingt geeignet. Falls die Verfahrensordnung aufgrund nicht ausreichender Glattheit der Lösung nicht exakt eingehalten wird, so ist die Steuerungsvorschrift (6.13) wirkungslos.*

### 6.1.4. Vergleich zweier Lösungen mit unterschiedlicher Schrittlänge

Als zweite Standardidee für die Steuerung der Schrittweite bei der numerischen Berechnung von gewöhnlichen Differentialgleichungen gilt der Vergleich zweier Lösungen mit unterschiedlicher Schrittweite (vgl. z.B. [SB00, Kap. 7.2.5]). Dazu betrachten wir zunächst wie schon im letzten Abschnitt die Semidiskretisierung nach  $x$ . Wieder gilt

$$w'(\tau) = f(\tau, w) + \mathcal{O}(\Delta x^2). \quad (6.14)$$

Wir vernachlässigen ebenfalls wieder die  $\mathcal{O}(\Delta x^2)$ -Terme und schauen uns zwei Diskretisierungen dieser Gleichung mittels desselben Verfahrens der Ordnung  $p$  an. Es sei  $w_I^{(m+1)}$  die erhaltene Approximation nach einem Schritt mit Länge  $\Delta\tau_m$  und  $w_{II}^{(m+1)}$  die Approximation mittels zwei Schritten der Länge  $\frac{\Delta\tau_m}{2}$ . Für den jeweiligen Diskretisierungsfehler gilt dann:

$$w(\tau + \Delta\tau) - w_I^{(m+1)} \doteq C_{FI} \Delta\tau_m^p \quad (6.15)$$

$$w(\tau + \Delta\tau) - w_{II}^{(m+1)} \doteq C_{FI} \left( \frac{\Delta\tau_m}{2} \right)^p. \quad (6.16)$$

Durch Subtraktion ergibt sich eine Approximation der Fehlerkonstante

$$C_{FI} \doteq \frac{2^p (w_{II}^{(m+1)} - w_I^{(m+1)})}{(\Delta\tau_m)^p (2^p - 1)}. \quad (6.17)$$

Über eine Taylorentwicklung der Fehlerkonstante für  $\tau + \Delta\tau$  um  $\tau$  erhält man dann eine Steuerungsvorschrift der Form

$$\Delta\tau_{m+1} = \Delta\tau_m \left( \frac{2^p - 1}{2^p} \frac{tol}{\|w_I^{(m+1)} - w_{II}^{(m+1)}\|} \right)^{\frac{1}{p+1}}. \quad (6.18)$$

### Wahl des Verfahrens

Im Allgemeinen versucht man das verwendete Verfahren so auszuwählen, dass die Ordnung möglichst hoch ist. Für das Crank-Nicolson-Verfahren ist nach unserem bisherigen Wissen das Ergebnis des ersten Iterationsschrittes unbrauchbar; man vergleiche dazu nochmal die Ausführungen in Kapitel 5.2.1. Hier ist in jedem Falle Rannacher smoothing anzuwenden. Darüber hinaus haben wir mehrfach die Notwendigkeit des Rannacher smoothing für jeden Zeitschritt mit dem Resultat einer rein impliziten Berechnung betont (vgl. Korollar 5.2.15). Als geeignetes Einschrittverfahren bleibt das implizite Euler-Verfahren mit der Ordnung  $p = 1$  übrig. Es ergibt sich der

#### Algorithmus 6.1.8. Implizit/Implizit-Halbe Steuerung

i) Man wähle

$$\Delta\tau_{m+1} = \Delta\tau_m \sqrt{\frac{tol}{2 \|u_{impl.}^{(m)} - u_{\frac{impl.}{2}}^{(m)}\|}}. \quad (6.19)$$

Für  $m = 0$  sei  $\Delta\tau_1 := \Delta\tau_{start}$  vorgegeben.

ii) Man berechne über einen Schritt des impliziten Euler-Verfahrens mit  $\Delta\tau_{m+1}$  als Schrittlänge  $u_{impl.}^{(m+1)}$  sowie mittels zwei impliziten Euler-Schritten der Länge  $\frac{\Delta\tau_{m+1}}{2}$  den Näherungswert  $u_{\frac{impl.}{2}}^{(m+1)}$ .

iii) Falls  $\|u_{impl.}^{(m+1)} - u_{\frac{impl.}{2}}^{(m+1)}\| < tol$  erfüllt ist, weiter mit Schritt i). Andernfalls führe man die Anpassung  $\Delta\tau_{m+1} := \gamma\Delta\tau_{m+1}$  mit geeignetem  $0 < \gamma < 1$  durch. Danach wiederhole man Schritt ii).

## 6.2. Theoretischer Vergleich der Verfahren

In diesem Abschnitt werden der Reihe nach die vorangehenden Verfahren auf ihre theoretischen Eigenschaften untersucht. Die Punkte Aufwand und Verlässlichkeit spielen dabei eine wichtige Rolle. Eine Zusammenfassung der Ergebnisse in Kurzform bildet den letzten Teil dieses Kapitels.

### 6.2.1. Impliziter Euler mit konstanter Schrittweite

Diese Verwendung des impliziten Euler-Verfahrens gibt eigentlich keine Steuerungsmöglichkeit wieder. Die Zeitschrittweite soll per Definitionem konstant bleiben, jedoch stellt dieses Vorgehen die gradlinigste Implementierung eines Näherungsverfahrens dar. Es ist unkompliziert, den (modifizierten) Algorithmus von Dempster, Hutton und Richards mit diesem Verfahren umzusetzen. Die nach Korollar 5.2.15 erforderliche rein implizite

Berechnung zur Erzeugung von Rannacher Glattheit in jedem Schritt wird mit diesem Algorithmus automatisch durchgeführt. Offensichtliche Nachteile sind hier die fehlende Schrittweitenvariation sowie die fehlende Fehlerschätzung. Nach der einführenden Motivation in Kapitel 5.1 ist anzunehmen, dass eine Vergrößerung der Schrittweite gegen Ende der Berechnung durchaus sinnvoll sein kann. Gleichzeitig besteht ein Interesse daran, dass die Lösung lokal hinreichend genau ist, insbesondere ob ein gewünschtes Ergebnis mit einem geringeren numerischen Aufwand erzielt werden kann. Mittels des Satzes 5.3.2 erhält man für ein äquidistantes Gitter in  $x$ -Richtung mit  $N$  Gitterpunkten einen Aufwand von ungefähr  $10N$  Operationen für das Aufstellen der Iterationsmatrix. Diese Matrix zu zerlegen stellt einen Aufwand von  $3N$  Operationen dar. Schließlich muss in jedem der  $M_1 = \frac{1}{\Delta\tau}$  Zeitschritte ein Gleichungssystem gelöst werden, mit dem Aufwand von je  $11N$  Operationen.

### 6.2.2. Steuerung über den Wertzuwachs der Lösung

Der EJ-Algorithmus greift im Wesentlichen die beiden größten Nachteile des impliziten Euler-Verfahrens mit konstanter Schrittweite auf. Es wird für dieses Verfahren zugelassen, dass sich die Schrittweite ändern kann. Die Änderung der Schrittweite beruht auf einer Schätzung des Approximationsfehlers über die Zeitableitung, die wiederum über den Lösungszuwachs approximiert wird. Dieses Verfahren ist einfach, vor allem aber benötigt es keine zusätzliche Vergleichslösung, um den Fehler schätzen zu können. Es entsteht also nur ein zusätzlicher Aufwand für die Neubesetzung und Zerlegung einer Iterationsmatrix, die für den Wechsel der Zeitschrittweite notwendig wird.

Ein Nachteil ist die weiterhin relativ hohe Anzahl von Zeitschritten. Es handelt sich um die Größenordnung  $\mathcal{O}(1/tol)$ . Dazu vergleiche man auch die Ausführungen von Forsyth und Vetzal [FV02, Abschnitt 9]. Die Implementierung unterscheidet sich vom impliziten Euler-Verfahren ohne Steuerung nur durch den einfach umzusetzenden Algorithmus 6.1.2.

### 6.2.3. Vergleich mit Verfahren höherer Ordnung

Aus den Vorbemerkungen geht hervor, dass sich hinter dieser Klasse von Verfahren im Wesentlichen der Vergleich eines impliziten Euler-Schrittes mit einem Crank-Nicolson-Schritt verbirgt. Für diesen Vergleich muss einige Arbeit geleistet werden. Zum einen ist, wie bei den beiden vorangehenden Verfahren, eine Lösung mit dem impliziten Euler-Verfahren zu berechnen. Des Weiteren wird ebenfalls ein neues Gleichungssystem für das Crank-Nicolson-Verfahren notwendig. Die erforderlichen Matrizen lassen sich mit Kenntnis des impliziten Systems relativ leicht füllen. Jedoch muss diese Matrix für jeden Schrittweitenwechsel nach der Neubesetzung auch zerlegt werden. Dabei verdoppelt sich der für Matrixzerlegungen benötigte Aufwand. Für die Lösung des Gleichungssystems kommen zu den  $11N$  Operationen, die für einen impliziten Schritt notwendig sind, noch zusätzlich etwa  $16N$  Operationen für die Lösung des Crank-Nicolson-Systems im selben Zeitschritt hinzu. Auch wenn hier über den Algorithmus 6.1.6 eine Schrittweitensteuerung vorgenommen werden kann, so beinhaltet dieser doch einen wesentlichen Nachteil. Wie schon in Korol-

lar 6.1.7 angemerkt, ist die verwendete Steuerung nicht verlässlich, weil für das Crank-Nicolson-System kein Rannacher smoothing stattfindet. Dabei treten die in Kapitel 5.2.1 und 5.2.2 beschriebenen Effekte für die Crank-Nicolson-Lösung auf. Eine vernünftige Auswertung der entsprechenden Fehlernorm ist dadurch evtl. nicht mehr möglich. Insbesondere sollte in jedem Falle nur mit der Lösung des impliziten Euler-Verfahrens weiter gerechnet werden, um diese Effekte durch aktives Rannacher smoothing zu reduzieren.

Die Implementierung des Verfahrens wird deutlich komplizierter, denn es müssen mehrere Gleichungssysteme gelöst werden, deren Struktur ähnlich, aber keinesfalls gleich ist. Der entsprechende Lösungsalgorithmus muss zwischen einem Gleichungssystem für den impliziten Euler und einem System für Crank-Nicolson unterscheiden.

### 6.2.4. Vergleich zweier Lösungen mit unterschiedlicher Schrittlänge

Um für beide Schrittlängen ein Rannacher smoothing zu erhalten, wird ausschließlich das implizite Euler-Verfahren verwendet. Verwendet man den Algorithmus 6.1.8, so ist der Mehraufwand für das System mit der halben Schrittweite zu beachten. Es sind in jedem Zeitschritt zusätzlich zum normalen impliziten Euler-Schritt zwei weitere Schritte der halben Länge durchzuführen, die einen Mehraufwand von  $2 \cdot 11N$  Operationen bewirken. Für das Aufstellen und Zerlegen der Iterationsmatrizen bei jeder Schrittweitenänderung ist jeweils der doppelte Aufwand im Vergleich zum impliziten Euler-Verfahren mit konstanter Schrittweite notwendig. Eindeutiger Vorteil bleibt allerdings das Rannacher smoothing für beide Lösungen.

Eine Implementierung des Algorithmus wird durch die Steuerungsprozedur aufwendiger als die ersten beiden Verfahren, jedoch reicht ein Gleichungssystemlöser zur Lösung beider Systeme aus. Weil die Schrittweite von vorne herein als variabel angenommen werden muss, bereitet die Reduzierung auf die Hälfte keine zusätzlichen Schwierigkeiten bei der Umsetzung in ein Programm.

### 6.2.5. Zusammenfassung der Eigenschaften

Die ermittelten Eigenschaften werden zum Überblick noch einmal in der Tabelle 6.1 zusammengefasst. Dabei sind die für uns wichtigsten Entscheidungskriterien die variable Schrittlänge und das Rannacher smoothing. Beide sollten in jedem Falle vorhanden sein. Betrachtet man den Gesamtaufwand für die Verfahren, so ergibt sich folgendes Resultat:

#### **Lemma 6.2.1. Gesamtaufwand der Verfahren**

*Es sei mit  $M_i$  die Anzahl der Zeitschritte für das Verfahren bezeichnet. Weiter sei  $C_i$  die Anzahl der durchgeführten Schrittweitenwechsel beim entsprechenden Verfahren. Es sei nun angenommen, dass die Schrittweite jeweils optimal angepasst wird. Dann gelten folgende Aussagen:*

- i) Die Schrittweite  $\Delta\tau_m$  wird in **jedem** Schritt angepasst, um stets die optimale Schrittweite zu haben. Es folgt sofort  $C_i = M_i$  für  $i = 2, 3, 4$ . Für  $i = 1$ , d.h. implizites Euler-Verfahren ohne Steuerung, bleibt natürlich  $C_1 = 1$ .
- ii) Als Gesamtaufwand der Verfahren ergeben sich damit
- a)  $\sim (M_1 + 1) \cdot 11N$  Operationen für das implizite Euler-Verfahren.
  - b)  $\sim M_2 \cdot 24N$  Operationen für die Steuerung über Wertzuwachs.
  - c)  $\sim M_3 \cdot 35N$  Operationen für den Vergleich vom implizitem Euler-Verfahren mit dem Crank-Nicolson-Verfahren.
  - d)  $\sim M_4 \cdot 59N$  Operationen für den Vergleich von ganzer und halber Schrittweite mit dem implizitem Euler-Verfahren.
- iii) Ebenso muss bei optimaler Steuerung
- a) der EJ-Algorithmus mehr als die Hälfte der Schritte gegenüber dem impliziten Euler-Verfahren ohne Steuerung einsparen,
  - b) die implizit/Crank-Nicolson Steuerung, d.h. Vergleich des Lösung des impliziten Eulers mit der Lösung des Crank-Nicolson-Verfahrens, mehr als zwei Drittel gegenüber dem impliziten Euler-Verfahren ohne Steuerung einsparen,
  - c) die implizit/implizit Halbe Steuerung, d.h. Vergleich zweier Lösungen des impliziten Eulers mit einem ganzen und zwei halben Schritten, mehr als vier Fünftel gegenüber dem impliziten Euler-Verfahren ohne Steuerung einsparen,
- um gemäß dem Gesamtaufwand ähnlich effektiv zu sein.

**Beweis:**

- i) Folgt aus der Anpassung in jedem Schritt.
- ii) Folgt mittels i) und Satz 5.3.2.
- iii) Folgt überschlagsweise aus ii). □

Die Zahlen sprechen eine deutliche Sprache für die Verwendung einer Steuerung über den Lösungszuwachs. Dieses Verfahren wird schon bei einer relativ geringen Einsparung in der Anzahl der Schritten bezüglich des Aufwands effektiv. Bei den vergleichenden Verfahren ist der Ordnungsvergleich mit 60 Prozent des Aufwands, bei gleicher Anzahl von Zeitschritten, zu bevorzugen. Hier weisen wir jedoch erneut auf das fehlende Ranacher smoothing hin, welches die Crank-Nicolson Vergleichslösung ineffektiv werden lässt.

Für alle drei Verfahren mit Steuerungsfunktion lässt sich folgende Beobachtung festhalten:

**Beobachtung 6.2.2. Trade-off: Optimalität – Effektivität:**

*Der Vergleich des Aufwands für die Änderung der Zeitschrittweite, also die Neubesetzung und Zerlegung der Iterationsmatrizen, mit dem Aufwand zur Durchführung eines Iterationsschrittes liefert jeweils ungefähr ein eins zu eins Verhältnis.*

*Verzichtet man an dieser Stelle auf die optimale Anpassung der Zeitschrittweite in jedem Schritt, so sind die Abfolgen*

*Iterationsschritt → Anpassung            → Iterationsschritt → Anpassung*

*sowie*

*Iterationsschritt → Iterationsschritt → Iterationsschritt → Anpassung*

*bezüglich ihres Aufwands gleichwertig. Der Verzicht auf die optimale Anpassung der Schrittweite  $\Delta\tau_m$  bei geringer Änderung ermöglicht die Durchführung von etwa drei Schritten statt bisher nur zwei Schritten mit dem gleichen Aufwand.*

Die Reduzierung des Aufwands durch den Verzicht auf die Verwendung der optimalen Schrittweite wird insbesondere im Kapitel 7.1.3 untersucht.

Eigenschaft	impliziter Euler mit konstanter Schrittweite	Steuerung über Zuwachs	Vergleich implizit mit Crank-Nicolson	Vergleich implizit mit implizit Halbe
Schrittlänge variabel	nein	ja	ja	ja
Anzahl der Zeitschritte	$M_1 = \frac{1}{\Delta\tau}$	$M_2 = \mathcal{O}(1/tol)$	$M_3$	$M_4$
Anzahl der Schrittweitenwechsel	$C_1 = 1$	$C_2$	$C_3$	$C_4$
Rannacher smoothing	ja	ja	nur für das implizite System	ja
Fehlerschätzung	nein	ja	ja	ja
Mehraufwand gegenüber implizitem Euler je Zeitschritt	—	nur Fehlerbestimmung	Lösung eines CN-Systems	2-fache Lösung des impl/2-Systems
Mehraufwand in Operationen gegenüber implizitem Euler je Zeitschritt	—	—	$16N$	$2 \cdot 11N$
zusätzliche Matrizen	—	—	$A_{CN}, B_{CN}$	$A_{impl./2}$
Anzahl zu zerlegender Matrizen	1	1	2	2
Aufwand Matrizen besetzen	$\sim 10N$	$\sim C_2 \cdot 10N$	$\sim C_3 \cdot 13N$	$\sim 2 \cdot C_4 \cdot 10N$
Aufwand Matrixzerlegungen	$\sim 3N$	$\sim C_2 \cdot 3N$	$\sim 2 \cdot C_3 \cdot 3N$	$\sim 2 \cdot C_4 \cdot 3N$
Aufwand Gleichungssystem(e) lösen	$\sim M_1 \cdot 11N$	$\sim M_2 \cdot 11N$	$\sim M_3 \cdot (11 + 16)N$	$\sim 3 \cdot M_4 \cdot 11N$
Gesamtaufwand für $C_i = M_i$	$\sim (M_1 + 1) \cdot 11N$	$\sim M_2 \cdot 24N$	$\sim M_3 \cdot 35N$	$\sim M_4 \cdot 59N$
Implementierung	einfach	einfach	kompliziert	kompliziert

Tabelle 6.1.: Zusammenfassung der Eigenschaften aller vier Verfahren





# 7. Numerische Umsetzung

Nachdem die theoretischen Voraussetzungen geschaffen worden sind, sollen die Verfahren auf ihre Praxistauglichkeit getestet werden. Außerdem muss eine Anpassung der Grundalgorithmen zur Schrittweitensteuerung an den modifizierten Algorithmus von Dempster, Hutton und Richards vorgenommen werden.

## 7.1. Ein erster Praxistest

Wir starten damit, einen Iterationslauf für ein Testproblem durchzuführen. Zuvor muss die Norm aus den vorangehenden Kapitel spezifiziert werden.

In den Steuerungsalgorithmen taucht jeweils eine Norm der Form  $\|u_I^{(m+1)} - u_{II}^{(m+1)}\|$  auf, wobei die Indizes zwei alternative Näherungslösungen bezeichnen. Üblicherweise wird an dieser Stelle entweder die Euklidische Norm oder die Maximumnorm verwendet. Wir zeigen, dass die Maximumnorm hier die bessere Wahl ist. Dazu fordern wir eine von der Diskretisierung unabhängige Zeitschrittweite  $\Delta\tau$ , d.h. bei gleicher Maschenweite  $\Delta x$  ist die neue Zeitschrittweite weder von der Anzahl der Diskretisierungspunkte noch von der genauen Lage der Bereichsgrenzen  $x_{min}$  bzw.  $x_{max}$  abhängig. Diese Forderung bereitet bei der Euklidischen Norm jedoch aufgrund der Komplementarität Probleme. Wir machen dazu folgende

### **Bemerkung 7.1.1. Wahl der Norm**

*Beide Näherungslösungen unterscheiden sich im Stoppbereich  $\mathcal{S}$  nicht. Unter der Annahme, dass die Lösung im Haltebereich  $\mathcal{H}$  in jedem Knoten einen durchschnittlichen Unterschied der Größe  $\alpha$  aufweist, betrachten wir das Verhalten der Euklidischen Norm der Lösungsdifferenz. Diese hat den Wert*

$$\|u_I^{(m+1)} - u_{II}^{(m+1)}\|_2 = \frac{1}{N} \sqrt{N_{\mathcal{H}} \alpha^2}.$$

*Dabei bezeichnet  $N$  die Anzahl der Diskretisierungspunkte in  $x$ -Richtung. Entsprechend bezeichnet  $N_{\mathcal{H}}$  die Anzahl dieser Punkte, die in  $\mathcal{H}$  liegen. Es zeigt sich, dass die Norm mit wachsendem  $N_{\mathcal{H}}$  wächst bzw. entsprechend fällt. Zum Beispiel wächst der Wert von  $N_{\mathcal{H}}$ , wenn der komplette Gitterausschnitt um einen Knoten nach rechts verschoben wird. Ebenso ändert sich der Wert der Norm, wenn der Gitterausschnitt verkleinert oder vergrößert wird. Damit ist der Wert der Norm nicht mehr unabhängig vom Gitter.*

Es ergibt sich das

**Korollar 7.1.2. Wahl der Norm**

*Aufgrund der Bemerkung 7.1.1 bietet es sich nicht an, die Euklidische Norm zu verwenden. Statt dessen werden wir im Folgenden immer die Maximumsnorm verwenden.*

**7.1.1. Ein Beispiel zum Einstieg**

Nachdem wir die Norm zur Ermittlung der Fehlerschätzung festgelegt haben, wenden wir uns einem ersten Test zu. Hierzu soll eine Option mit, im Vergleich zum extremen Beispiel 3.3.1, gewöhnlichen Parametern berechnet werden. Hierfür wird das

**Beispiel 7.1.3. Einstiegsproblem**

*Man berechne den Preis eines amerikanischen Vanilla Puts mit  $K = 1.0$ ,  $T = 1.0$ ,  $r = 0.1$ ,  $\delta = 0$  und  $\sigma = 0.3$ .*

betrachtet. Die Parameter der Diskretisierung werden als  $x_{min} = -2$ ,  $x_{max} = 2$ ,  $N = 401$ , d.h.  $\Delta x = \frac{1}{100}$  gewählt. Mit der durch die Konvektions-Verschiebung transformierten Gleichung wird ein erster Iterationslauf mit konstanter Schrittweite  $\Delta\tau$  vorgenommen. Dabei wird gleichzeitig eine Näherung für das Maximum über alle  $x$  für festes  $\tau$  von  $v_\tau(x, \tau)$  gewonnen. Hierzu wird eine Finite-Differenzen-Näherung

$$v_\tau(x, \tau) \approx \frac{u_k^{(m+1)} - u_k^{(m)}}{\Delta\tau}$$

verwendet. Es ergibt sich das aufgrund der Motivation aus Kapitel 5.1 erwartete Bild. Im Verlauf der Iteration, also für  $\tau \rightarrow T = 1$ , nimmt der Wert von  $v_\tau(x, \tau)$  deutlich ab. Man vergleiche dazu auch die Abbildung 7.1.

Nachdem die Motivation für eine Schrittweitensteuerung an einem Beispiel ‚gerechtfertigt‘ wurde, soll diese Steuerung in einer ersten Form angewandt werden. Es zeigt sich empirisch, dass das Beispiel 7.1.3 die Bedingung iii) des Korollars 4.2.1 nicht erfüllt. Der ursprüngliche DHR-Algorithmus wird ohne Unterbrechung durchlaufen. Somit lässt sich eine Schrittweitensteuerung einfach zwischen die einzelnen Zeitschritte integrieren. Für den Algorithmus von Eriksson und Johnson wird dies direkt umgesetzt. Wir erinnern uns dabei noch einmal an die Beobachtung 6.2.2. Dort ist die Aussage gemacht worden, dass es zweckmäßig sein kann, die Schrittweite nicht in jedem Schritt anzupassen, sondern dies nur sporadisch zuzulassen. Eine realisierbare Idee ist es, nur ganze Zehnerpotenzen als mögliche Schrittweiten zuzulassen und damit gleichzeitig die Anzahl der Schrittweitenwechsel herabzusetzen. Statt der bisherigen Steuerungsvorschrift

$$\Delta\tau_{m+1} = \Delta\tau_m \frac{tol}{\|u^{(m)} - u^{(m-1)}\|} \tag{7.1}$$

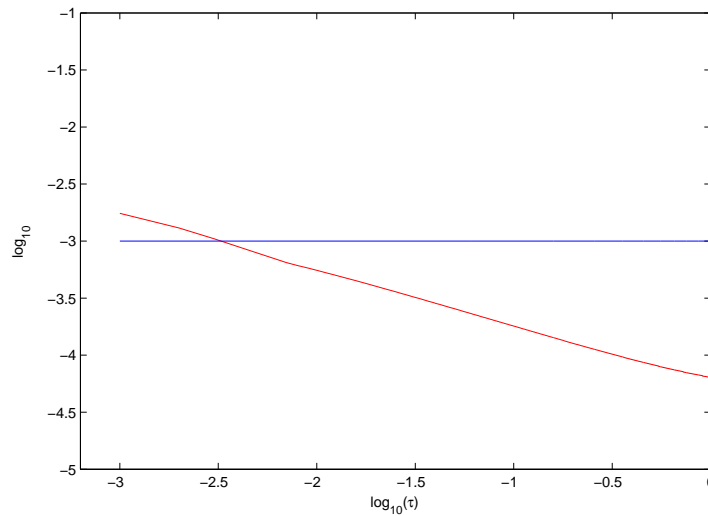


Abbildung 7.1.: Maximum über alle  $x$  für festes  $\tau$  der **Approximation von  $v_\tau(x, \tau)$**  im Zeitablauf im Vergleich zur **Schrittweite** für das Beispiel 7.1.3 mit  $x_{min} = -2$ ,  $x_{max} = 2$ ,  $\Delta x = \frac{1}{100}$  und  $\Delta \tau = \frac{1}{1000}$ .

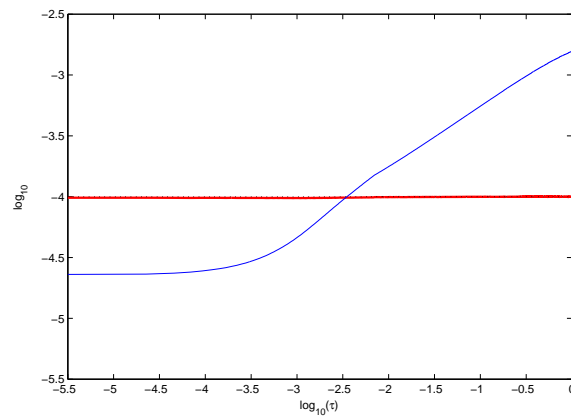
(vgl. hierzu Gleichung (6.1) im Algorithmus 6.1.2) wird nun die Vorschrift

$$\Delta \tau_{m+1} = 10^{\text{round}\left(\log_{10}\left(\Delta \tau_m \frac{\text{tol}}{\|u^{(m)} - u^{(m-1)}\|}\right) - 0.5\right)} \quad (7.2)$$

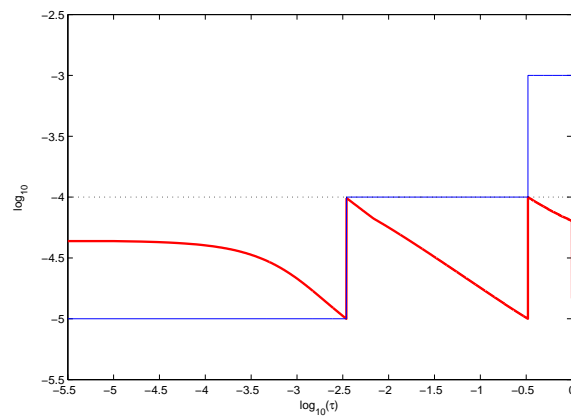
verwendet.

Der Unterschied zwischen den beiden Vorschriften zeigt sich in Abbildung 7.2. In beiden Fällen wurde das Beispiel 7.1.3 mit  $x_{min} = -2$ ,  $x_{max} = 2$ ,  $\Delta x = \frac{1}{100}$  diskretisiert. Die lokale Toleranz für den verwendeten EJ-Algorithmus wurde mit  $\text{tol} = 10^{-4}$  vorgegeben. Für die erste der beiden Berechnungen ist ein stufenloser Schrittweitenwechsel in jedem Schritt möglich. Es werden insgesamt 1170 Iterationen benötigt. Sind jedoch nur stufenweise Wechsel der Schrittweite auf die nächste Zehnerpotenz möglich, so sieht man deutlich den sich verringernden Fehler und beim Unterschreiten der Schranke  $\text{tol} \cdot 10^{-1}$  den Wechsel auf die nächste zulässige Schrittweite. Hier ist nur für jeden dieser Sprünge eine Neubesetzung der Matrizen notwendig, allerdings werden insgesamt 4291 Iterationen durchgeführt. Der erhoffte Performancegewinn bleibt (noch) aus. Der Grund hierfür ist die geringe Startschrittweite  $\Delta \tau = 10^{-5}$ , die für die stufenweise Steuerung sehr lange beibehalten wird. Die Auswahlprozedur über Zehnerpotenzen der Schrittweite ist noch nicht vollständig ausgefeilt. Dazu wird noch eine genauere Untersuchung stattfinden.

Für die beiden anderen echten Steuerungsverfahren ergeben sich qualitativ ähnliche Ergebnisse.



(a) stufenloser Schrittweitenwechsel (1170 Iterationen)



(b) Schrittweitenwechsel sind nur um Zehnerpotenz zulässig (4291 Iterationen)

Abbildung 7.2.: **Lokaler Fehler** im Zeitablauf im Vergleich zur **Schrittweite** für das Beispiel 7.1.3 mit  $x_{min} = -2$ ,  $x_{max} = 2$ ,  $\Delta x = \frac{1}{100}$ . Für den EJ-Algorithmus ist die lokale Toleranz  $tol = 10^{-4}$  vorgegeben worden.

### 7.1.2. Schrittweitenabsturz bei Implizit/Crank-Nicolson Vergleich

Weil wir bereits mehrfach auf die Problematik, die Lösung des Crank-Nicolson-Verfahrens zu verwenden, hingewiesen haben (vgl. Korollar 6.1.7 und Kapitel 5.2.2), soll diese Warnung mit einem Beispiel belegt werden. Wir verwenden erneut das Beispiel 7.1.3 mit

der gleichen Diskretisierung wie bereits zuvor. Es seien wieder  $x_{min} = -2$ ,  $x_{max} = 2$ ,  $\Delta x = \frac{1}{100}$ . Lediglich die Toleranzschränke wird auf  $tol = 10^{-7}$  verringert. Dies ist nicht ganz angemessen für die Wahl von  $\Delta x$ , allerdings lässt sich damit ein durch die Lösung des Crank-Nicolson-Verfahrens hervorgerufener ‚Schrittweitenabsturz‘ provozieren. Mit diesem Begriff bezeichnen wir die Reduzierung der aktuellen Zeitschrittweite um mehrere Zehnerpotenzen bei gleichzeitigem Abbruch des Algorithmus.

Wir betonen die ausschließliche Verwendung der Lösung des impliziten Euler-Verfahrens für die weitergehende Iteration. Die über Crank-Nicolson berechnete zweite Lösung wird einzig und allein zur Schätzung des Iterationsfehlers mitberechnet. Trotzdem wird die Schätzung des Fehlers so sehr durch den Knick in der Crank-Nicolson-Lösung beeinflusst, dass eine verlässliche Ermittlung des Iterationsfehlers nicht mehr möglich ist. Abbildung 7.3 gibt ein Protokoll des Iterationsverhaltens wieder. Es ist deutlich ein Flattern des Fehlers für  $10^{-2} \leq \tau \leq 10^{-0.5}$  zu beobachten. Unmittelbar damit verbunden ist die kurzfristige Reduktion der Schrittweite nach einem solchen Fehlerpeak. Letztendlich kommt es nach 1082 Iterationsschritten für  $\tau \approx 10^{-0.5}$  zu einer anormalen Reduktion der Schrittweite, in deren Folge der Algorithmus abbricht.

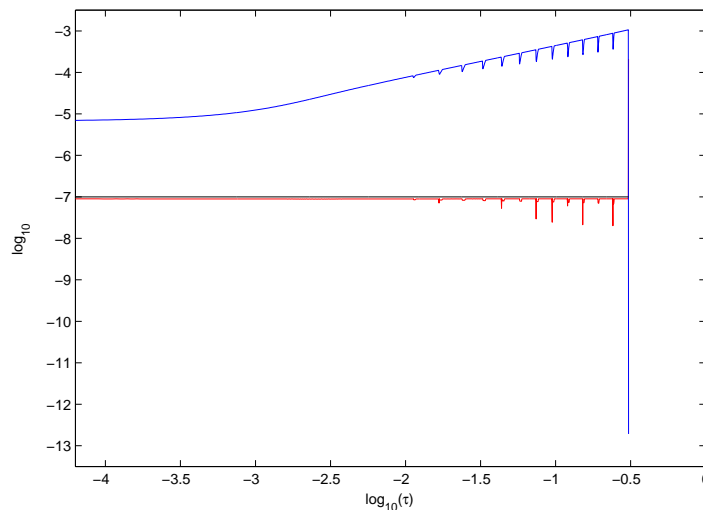


Abbildung 7.3.: **Schrittweite** und **Fehlerschätzung** im Zeitablauf für Steuerung über Implizit/Crank-Nicolson Vergleich am Beispiel 7.1.3 mit  $x_{min} = -2$ ,  $x_{max} = 2$ ,  $\Delta x = \frac{1}{100}$  mit  $tol = 10^{-7}$

Für diesen Vorgang gilt:

**Bemerkung 7.1.4. Auftreten des Schrittweitenabsturzes**

- i) Verwendet man die mittels Crank-Nicolson-Verfahren berechnete Näherung zur Fortsetzung der Iteration, obwohl sie per Konstruktion des Verfahrens nur zur Fehlerschätzung berechnet wird, so treten solche Effekte wesentlich häufiger auf. Die notwendige Glättung durch Rannacher smoothing geht in die Lösung nicht mehr ein.*
- ii) Der Effekt ist abhängig von der Diskretisierung. Für das betrachtete Beispiel reicht es aus, das komplette Gitter um  $\frac{\Delta x}{2}$  nach rechts zu verschieben, um einen Abbruch bereits nach 308 statt den oben beobachteten 1082 Iterationen zu erreichen.*
- iii) Durch die in ii) beschriebene Abhängigkeit ist es schwierig, exakte Kriterien für einen Absturz anzugeben. Als Faustregel ist jedoch festzuhalten:*
  - a) Die Verwendung des Crank-Nicolson-Verfahrens zur Fehlerschätzung ist mit Vorbehalt zu behandeln.*
  - b) Ein möglicher Abbruch des Algorithmus wird umso wahrscheinlicher, desto größer das Missverhältnis von  $\Delta x$  zu  $tol$  ist.*

**7.1.3. Optimierung des Schrittweitenwechsels**

Im Einstiegsabschnitt hatten wir uns bereits mit einer Idee befasst, die die Anzahl der Schrittweitenwechsel und den damit verbundenen Aufwand reduzieren kann. Dort wurde vorgeschlagen, einen Wechsel der Schrittweite nur um Zehnerpotenzen zuzulassen. Die Umsetzung war problemlos, denn die Vorschrift zur Anpassung der Schrittweite musste lediglich gegen Gleichung (7.3) ausgetauscht werden. Das Ergebnis für das Einstiegsbeispiel 7.1.3 war eine Reduktion der Anzahl der Schrittweitenwechsel, verbunden mit der vierfachen Gesamtzahl von Iterationen. Die Feinheit einer Änderung von  $v_\tau(x, \tau)$  ist ebenfalls nur noch stufenweise und damit nicht mehr hinreichend genau erfasst worden.

Eine verbesserte Idee ist es, Bruchteile von Zehnerpotenzen zuzulassen, z.B. statt nur einer zwei oder drei Schrittweitenstufen pro Zehnerpotenz zu erlauben. Dazu wird die Auswahlgleichung durch

$$\Delta\tau_{m+1} = 10^{\left( \text{round} \left( \alpha \cdot \log_{10} \left( \Delta\tau_m \frac{tol}{\|u^{(m)} - u^{(m-1)}\|} \right) - 0.5 \right) / \alpha \right)} \tag{7.3}$$

ersetzt. Die natürliche Zahl  $\alpha$  gibt hier die Anzahl der Schrittweitenstufen je Zehnerpotenz an. Ein Wechsel auf die nächste Stufe erfolgt nun beim Unterschreiten der Schranke  $tol \cdot 10^{-1/\alpha}$ . Der Fall des stufenlosen Wechsels wird dabei für  $\alpha$  gegen Unendlich eingeschlossen. Variiert man  $\alpha$ , so kann man die unterschiedlichen Iterationsverläufe mit verschiedenen Anzahlen von Schrittweitenwechseln und unterschiedlichen Gesamtiterationsanzahlen beobachten. Diese Variation ist in der Tabelle 7.1 zusammengefasst.

$\alpha$	Anzahl Iterationen	Anzahl Wechsel	Laufzeit CPU-Sek.	$\alpha$	Anzahl Iterationen	Anzahl Wechsel	Laufzeit CPU-Sek.
1	4291	5	434	10	1320	21	38
2	2046	7	60	20	1239	39	36
3	1706	8	48	100	1183	186	38
4	1584	10	45	1000	1170	799	60
5	1520	12	43	10000	1170	1170	88

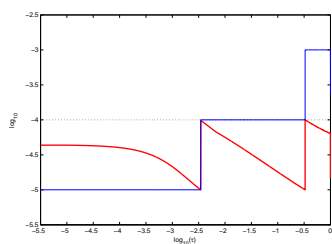
Tabelle 7.1.: Auswirkungen der Variation von  $\alpha$  auf den Berechnungsaufwand, d.h. Iterationen und Schrittweitenwechsel, sowie die Laufzeit für das mittels EJ-Algorithmus gesteuerte Beispiel 7.1.3

Der Gewinn in Form weniger Schrittweitenwechsel durch weniger mögliche Schrittweiten ist hier deutlich zu sehen. Auf der anderen Seite ist aber ebenso deutlich die hohe Gesamtzahl von Iterationen durch zu wenige zulässige Schrittweiten zu erkennen. Im Fall  $\alpha = 1$  arbeitet der Algorithmus fast so, als würde keine Variation der Schrittweite stattfinden. Es werden viel zu viele Iterationen durchgeführt. Um einen weiteren Vergleich zu haben, werden die Iterationsverläufe für verschiedene  $\alpha$  in Abbildung 7.4 wiedergegeben. Diese Abbildung zeigt im Vergleich der Teilbilder a) bis e) mit dem Teilbild f) deutlich, wie sehr die Schrittweite von der optimalen Schrittweite bei stufenloser Steuerungsmöglichkeit abweicht. Besonders bei Erhöhung der zulässigen Schrittweitenstufen pro Zehnerpotenz von 1 auf 2 ist eine Verbesserung durch diese Wahl sichtbar. Pro zusätzlicher Stufe werden hier, nach dem Wechsel in die eingefügte Zwischenstufe,  $10^{0.5}$ -mal die Anzahl der gemachten Schritte eingespart. Für die im Intervall  $10^{-3.5} \leq \tau \leq 10^{-2.5}$  überschlagsweise gemachten 90 Schritte der Länge  $10^{-4.5}$  wurden zuvor etwa 270 Schritte der Länge  $10^{-5}$  gemacht. Der Zugewinn nimmt mit weiterer Erhöhung von  $\alpha$  ab. Man sieht für  $\alpha = 10$  eine relativ gute Approximation der Schrittweitenfunktion  $\Delta\tau(\tau, tol)$  durch eine Treppenfunktion mit einer geringen Stufenhöhe und einer mäßigen Anzahl unterschiedlicher Stufen.

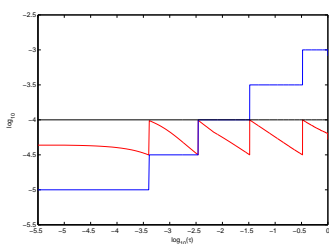
Zusammenfassend ergibt sich

**Bemerkung 7.1.5. Wahl von  $\alpha$**

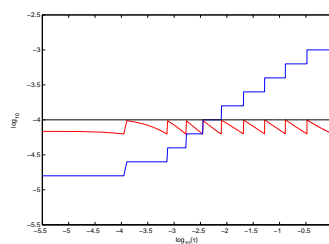
- i) Natürlich hängt die optimal zu verwendende Schrittweite  $\Delta\tau(\tau, tol)$  nicht nur von der Zeit  $\tau$  und der lokalen Toleranzgrenze  $tol$  ab. Es gehen insbesondere auch die Maschenweite  $\Delta x$  und die Parameter der Option ein. Ein großes  $\sigma$  bedeutet eine schnelle Diffusion und damit auch den schnellen Zerfall hochfrequenter Lösungsanteile. Der Algorithmus behält eine gewählte Schrittweite in diesem Fall nicht sehr lange bei. Eine deutlich zu klein gewählte Schrittweite wird demnach sehr schnell korrigiert.  $\alpha$  sollte daher in Abhängigkeit des Problems gewählt werden.
- ii) Weiterhin ist die Betrachtung ausschließlich für die Verwendung einer Steuerung über den Zuwachs der Lösung gemacht worden. Für eine genaue Wahl bei Verwendung der



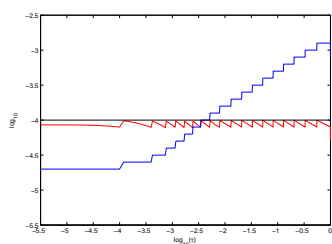
(a)  $\alpha = 1$ , 4291 Iterationen



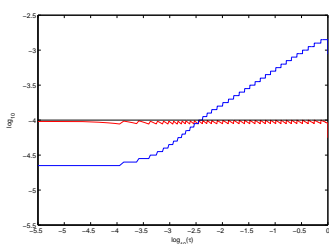
(b)  $\alpha = 2$ , 2046 Iterationen



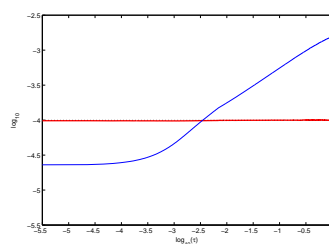
(c)  $\alpha = 5$ , 1520 Iterationen



(d)  $\alpha = 10$ , 1320 Iterationen



(e)  $\alpha = 20$ , 1239 Iterationen



(f)  $\alpha = 10000$ , 1170 Iterationen

Abbildung 7.4.: Variation von  $\alpha$  für das mittels EJ-Algorithmus gesteuerte Beispiel 7.1.3 und eine lokale Toleranz von  $tol = 10^{-4}$ . Dabei wird der lokale Fehler im Zeitablauf im Vergleich zur Schrittweite geplottet.

beiden anderen Methoden ist, ebenso wie hier, ein wenig Erfahrung notwendig.

iii) Eine geschickte Wahl des  $\alpha$  in Gleichung (7.3) kann durchaus eine Halbierung der Laufzeit gegenüber stufenloser Schrittweitenwahl bewirken. Allerdings führt eine schlechte Wahl von  $\alpha$  zu einer Verfünffachung der Laufzeit, wie in Tabelle 7.1 zu sehen ist.

Als Faustregel ist das nachfolgende Korollar zu sehen:

**Korollar 7.1.6. Faustregel zur Wahl von  $\alpha$**

In der Steuerungsvorschrift

$$\Delta\tau_{m+1} = 10 \left( \text{round} \left( \alpha \cdot \log_{10} \left( \Delta\tau_m \frac{tol}{\|u^{(m)} - u^{(m-1)}\|} \right) - 0.5 \right) / \alpha \right) \quad (7.4)$$

ist  $\alpha$  zwischen 5 und 10 eine geeignete Wahl. Liegt die verwendete Schrittweite auf der nächst niedrigeren Stufe, so ist in diesem Falle die 1.58-fache bzw. 1.25-fache Zahl von Schritten gegenüber der optimalen Wahl durchzuführen.



## 7.2. Verbinden der Algorithmen

Für den ersten Test hatten wir ein Beispiel ausgewählt, für das die Aussage iii) des Korollars 4.2.1 nicht erfüllt war. Trotz der Transformation über die Konvektions-Verschiebung war die Ausübungsgrenze in den  $(x, \tau)$ -Koordinaten weiterhin nicht wachsend. Die Frage ist nun, welche Auswirkungen ein Beispiel erzeugt, in dem diese Aussage erfüllt wird. Insbesondere wird dies im Fall der neuen Transformation und der angepassten Aussage iii) im Korollar 4.2.3 problematisch. Wir können bisher den groben Algorithmus festhalten:

**Algorithmus 7.2.1. Schrittweitensteuerung und DHR-Algorithmus (Grundform):**

- i) Man lege die Testschrittweite  $\Delta\tau_m^{TEST}$  fest.
- ii) Bestimmung der Lösung  $u_I^{(m+1)} := u_{impl.}^{(m+1)}$ , ausgehend von  $u^{(m)}$  unter Verwendung der Schrittweite  $\Delta\tau_m^{TEST}$  mit dem impliziten Euler-Verfahren.
- iii) Von  $u^{(m)}$  ausgehende Bestimmung der alternativen Lösung  $u_{II}^{(m+1)}$ . Dies erfolgt mittels des Crank-Nicolson-Verfahrens bzw. zwei impliziten Euler-Schritten der halben Schrittweite. Für die Steuerung über den Lösungszuwachs ist die alternative Lösung als  $u_{II}^{(m+1)} := u_{impl.}^{(m)}$  definiert.
- iv) Bestimmung von  $norm := \|u_I^{(m+1)} - u_{II}^{(m+1)}\|$ .
- v) Falls  $norm < tol$  erfüllt ist, akzeptiere man den Schritt, setze  $u^{(m+1)} := u_I^{(m+1)}$  und beginne einen neuen Schritt mit i). Ansonsten wird  $\Delta\tau_m^{TEST}$  auf  $\Delta\tau_m^{TEST}/2$  reduziert und mit Schritt ii) fortgefahren.

Innerhalb dieses Algorithmus sind insbesondere die Schritte ii) und iii) abhängig vom Verlauf der Ausübungsgrenze. An diesen beiden Stellen wird der Algorithmus von Dempster, Hutton und Richards eingesetzt, um den jeweiligen Schritt auszuführen. Dies ist zunächst unproblematisch. Wenn allerdings die oben genannte Aussage iii) der Korollare erfüllt ist, findet möglicherweise ein Wechsel der Suchrichtung statt. Interessant wird es besonders dann, wenn der Wechsel der Suchrichtung in einen verworfenen Schritt fällt. Wir illustrieren dies an der Abbildung 7.5. Ausgehend von der bekannten Lösung  $u^{(m)}$  nach  $m$  Schritten, wird mit der Schrittweite  $\Delta\tau_{m+1,I}$  ein erster Berechnungsversuch unternommen. Wir überschreiten das  $\tau^*$ , für das die Suchrichtung von fallend auf steigend wechselt. D.h. statt mit dem Algorithmus 4.1.2 (UL-update) wird nun mit dem Algorithmus 4.2.5 (UL-update<sup>-1</sup>) gearbeitet. Der Schritt wird jedoch verworfen und mit der Schrittweite  $\Delta\tau_{m+1,II}$  wiederholt. Gilt jetzt aber  $\tau_m + \Delta\tau_{m+1,II} < \tau^*$ , so ist die Suchrichtung noch fallend. Im verworfenen Schritt ist allerdings bereits auf eine steigende Suche nach dem Punkt  $x_{II}$  rechts von  $x_f(m)$  umgeschaltet worden. Der modifizierte DHR-Algorithmus wird nun rechts von der grün gepunkteten Linie nach dem Ausübungspunkt  $x_{II}$  suchen. Der gesuchte Punkt liegt jedoch links von dieser Linie und wird daher nicht gefunden. Der Algorithmus bricht folglich mit einem Fehler ab. Wir tragen dem Rechnung und modifizieren den Grundalgorithmus 7.2.1 durch Hinzufügen zweier Teilschritte.

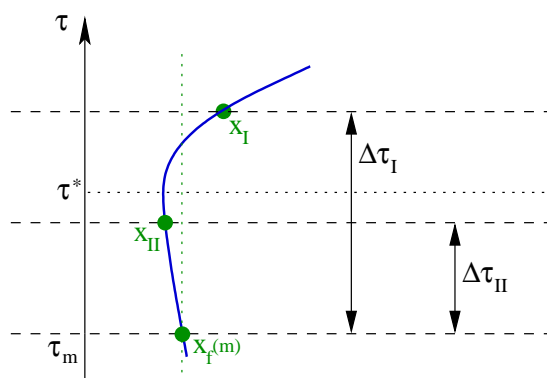


Abbildung 7.5.: Schematisch: Verschiedene Schrittweiten und Suchrichtungen beim modifiziertem DHR-Algorithmus. Der Schrittindex  $m$  für die Zeitschrittweiten ist der Übersichtlichkeit halber fallen gelassen worden. Im Bild sind zusätzlich die **Ausübungsgrenze**  $x_f$  und der Zeitpunkt für den Wechsel der Suchrichtung  $\tau^*$  sowie die möglichen Ausübungspunkte  $x_f(m)$ ,  $x_I$  und  $x_{II}$  wiedergegeben.

**Algorithmus 7.2.2. Schrittweitensteuerung und DHR-Algorithmus (lauffähig):**

- i) a) Man lege die Testschrittweite  $\Delta\tau_m^{TEST}$  fest.  
 b) *Speichern der Spaltenausrichtung.*
- ii) *Bestimmung der Lösung  $u_I^{(m+1)} := u_{impl.}^{(m+1)}$ , ausgehend von  $u^{(m)}$  unter Verwendung der Schrittweite  $\Delta\tau_m^{TEST}$  mit dem impliziten Euler-Verfahren.*
- iii) *Von  $u^{(m)}$  ausgehende Bestimmung der alternativen Lösung  $u_{II}^{(m+1)}$ . Dies erfolgt mittels des Crank-Nicolson-Verfahrens bzw. zwei impliziten Euler-Schritten der halben Schrittweite. Für die Steuerung über den Lösungszuwachs ist die alternative Lösung als  $u_{II}^{(m+1)} := u_{impl.}^{(m)}$  definiert.*
- iv) *Bestimmung von  $norm := \|u_I^{(m+1)} - u_{II}^{(m+1)}\|$ .*
- v) *Falls  $norm < tol$  erfüllt ist, akzeptiere man den Schritt, setze  $u^{(m+1)} := u_I^{(m+1)}$  und beginne einen neuen Schritt mit i). Ansonsten wird  $\Delta\tau_m^{TEST}$  auf  $\Delta\tau_m^{TEST}/2$  reduziert und*
  - a) *im Falle einer zu i) b) identischen Spaltenausrichtung wird mit Schritt ii) fortgefahren.*
  - b) *anderenfalls wird der Zustand aus i) b) wiederhergestellt und mit Schritt ii) fortgefahren.*

Die beiden Ergänzungen zeigen Wirkung. In den bisher kritischen Fällen funktioniert die Schrittweitensteuerung nun auch über den Punkt  $\tau^*$  hinaus. Abschließend wird die folgende Bemerkung gemacht:

**Bemerkung 7.2.3. Notwendigkeit der Ergänzungen**

- i) Die beiden Ergänzungen sind nicht immer notwendig. Wenn die Aussagen i) aus den Korollaren 4.2.1 bzw. 4.2.3 erfüllt sind, dann ist die Ausübungsgrenze  $x_f$  nicht wachsend und damit kein Wechsel auf die  $UL\text{-update}^{-1}$  Iteration notwendig.*
- ii) Auch für die Verwendung des inversen Updates greifen die Anpassungen nicht zwingend. Wenn der über  $\tau^*$  hinausgehende Schritt nicht verworfen wird, wird die Prozedur v) b) niemals abgearbeitet.*
- iii) Die Idee bis genau an die Stelle  $\tau^*$  mit der Schrittweite zu rechnen, also den  $(m + 1)$ -ten Schritt jeweils auf die Länge*

$$\Delta\tau_{m+1} := \min(\tau^* - \tau_m, \Delta\tau_{m+1})$$

*zu beschränken, kann nicht verwirklicht werden. Der Zeitpunkt  $\tau^*$  hängt von  $x_f(\tau)$  ab. Diese aus  $S_f(\tau)$  transformierte Ausübungsgrenze ist a priori unbekannt und muss deshalb implizit zusammen mit der Lösung bestimmt werden.*

## 7.3. Iteration im Detail

Bisher wurde die Klasse von Algorithmen an die Transformationen angepasst. Im Algorithmus 7.2.2 ist sowohl die genaue Steuerungsvorschrift als auch die verwendete Alternativlösung offen gelassen worden. Nun kann eine Detailbetrachtung der Iteration stattfinden. Wir haben in Kapitel 5.3 mit dem Satz 5.3.2 eine genaue Angabe der benötigten Rechenoperationen geliefert, außerdem haben wir uns in Kapitel 5.3.2 Gedanken über den nicht direkt messbaren Rechenaufwand für die Ermittlung der Ausübungsgrenze gemacht. Die Frage bleibt, ob sich weitere Aussagen über den Ablauf der Iteration treffen lassen.

Wir führen erneut eine Berechnung unseres Einstiegsbeispiels durch. Zur Wiederholung:

**Beispiel 7.3.1. Einstiegsproblem**

*Man berechne den Preis eines amerikanischen Vanilla Puts mit  $K = 1.0$ ,  $T = 1.0$ ,  $r = 0.1$ ,  $\delta = 0$  und  $\sigma = 0.3$ .*

Auch hier wird wieder die Diskretisierung der Form  $x_{min} = -2$ ,  $x_{max} = 2$ ,  $N = 401$ , d.h.  $\Delta x = \frac{1}{100}$  verwendet. Der einzige Unterschied ist nun, dass die Steuerung über den Vergleich Implizit, d.h. impliziter Euler, mit Crank-Nicolson statt der bisher verwendeten Steuerung über den Lösungszuwachs erfolgt. Wir setzen die lokale Toleranz  $tol$  wieder auf

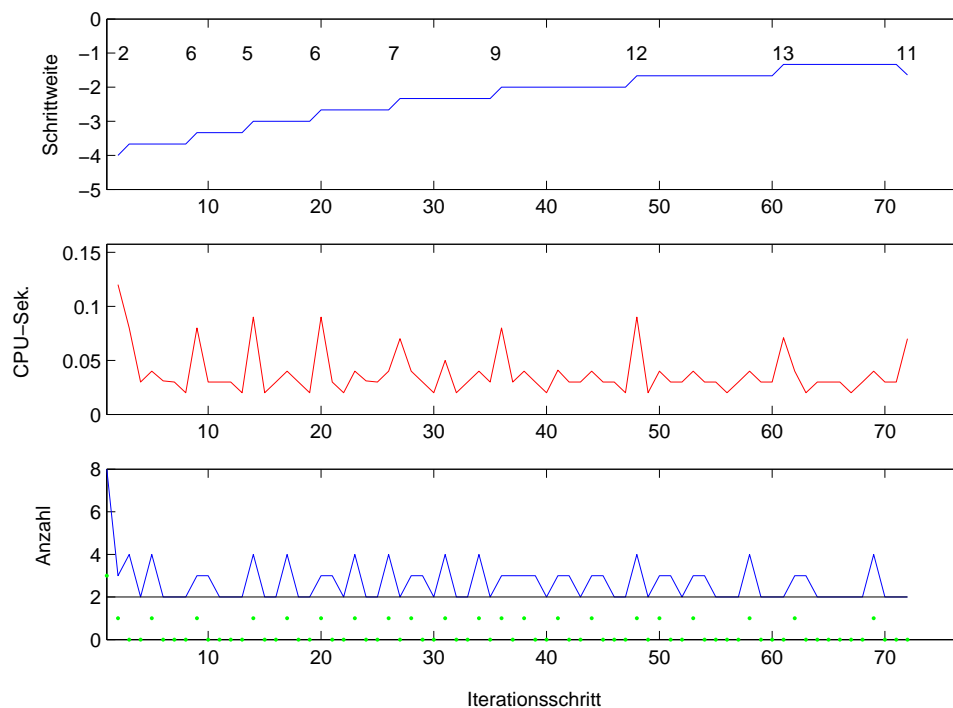


Abbildung 7.6.: Details für die Iteration bei einer Schrittweitensteuerung über den Vergleich der Lösungen von implizitem Euler und Crank-Nicolson:  
 oben: Schrittweite  $\Delta\tau_m$  für jeweiligen Iterationsschritt  $m$ ; Anzahl der mit dieser Länge durchgeführten Schritte  
 Mitte: CPU-Sekunden im  $m$ -ten Schritt  
 unten: Anzahl der gelösten Gleichungssysteme und der Knotenwechsel im  $m$ -ten Schritt

$10^{-4}$ , lassen allerdings nur noch 3 Schrittstufen je Zehnerpotenz zu.

Die Abbildung 7.6 gibt den Iterationsverlauf wieder. Die x-Achse gibt in allen drei Diagrammen die Nummer des jeweiligen Iterationsschrittes an. Im obersten Bild wird die für den Iterationsschritt aktuelle Schrittweite blau geplottet. Die an den Schrittweitenwechseln hochgestellten Zahlen geben die Anzahl der mit dieser Schrittweite durchgeführten Iterationsschritte wieder. Dabei ist deutlich das schnelle Abklingen der hochfrequenten Anteile der Lösung zu beobachten. Denn mit den zu Anfang verwendeten kleinen Schritten werden deutlich weniger Iterationen ohne Wechsel durchgeführt als gegen Ende der Iteration bei einer vergleichsweise großen Schrittweite.

Im zweiten Diagramm ist die auf den aktuellen Schritt verwendete CPU-Zeit, als Maß

für den Aufwand, in Sekunden wiedergegeben. Die Lage der in diesem Zusammenhang auftretenden Peaks an Stellen, an denen ein Schrittweitenwechsel stattfindet, fällt sofort ins Auge.

Der dritte Plot gibt u.a. die Anzahl der gelösten Gleichungssysteme wieder. Hierbei tritt pro Iterationsschritt eine Grundlast von 2 Systemen pro Schritt auf. Es muss in jedem Fall ein implizites und ein Crank-Nicolson-System gelöst werden. Kommt es nun zu einem Wechsel des Ausübungsknotens, im Plot durch die grünen Punkte markiert, so ist hierfür eigentlich für jede Diskretisierung ein zusätzliches Gleichungssystem zu lösen. Wenn man sich aber z.B. den neunten oder den zwanzigsten Schritt herausgreift, so wird in diesem Schritt nur ein zusätzliches Gleichungssystem gelöst. Der Knotenwechsel für das Crank-Nicolson-Verfahren und die damit verbundene Lösung des zweiten Gleichungssystems findet erst verspätet im nachfolgenden Schritt statt. Weiterhin gibt dieser Plot auch das Anlaufen der Iteration wieder. Als Startwert für  $nb$  wird immer der Index des Minimums aller  $x_i > 0$  gewählt. Wir wissen bereits, dass  $x_f(0) = 0$  ist. Genauer gilt  $x_f(0^+) = 0$ , falls  $r \geq \delta$  erfüllt ist und  $x_f(0^+) = \ln\left(\frac{r}{\delta}\right) < 0$  für  $r < \delta$ . Daher findet im ersten Iterationsschritt eine Suche nach dem  $nb_1$  statt, die mehrere Wechsel notwendig macht.

Anhand des gleichen Beispiels mit denselben Parametern betrachten wir insbesondere das letzte Diagramm für eine Steuerung über den Vergleich der Schrittlängen. Für den Vergleich der Lösung des impliziten Eulers mit der Lösung des impliziten Eulers bei halber Schrittweite ergibt sich das Bild 7.7. Als Grundlast sind nun drei Gleichungssysteme je

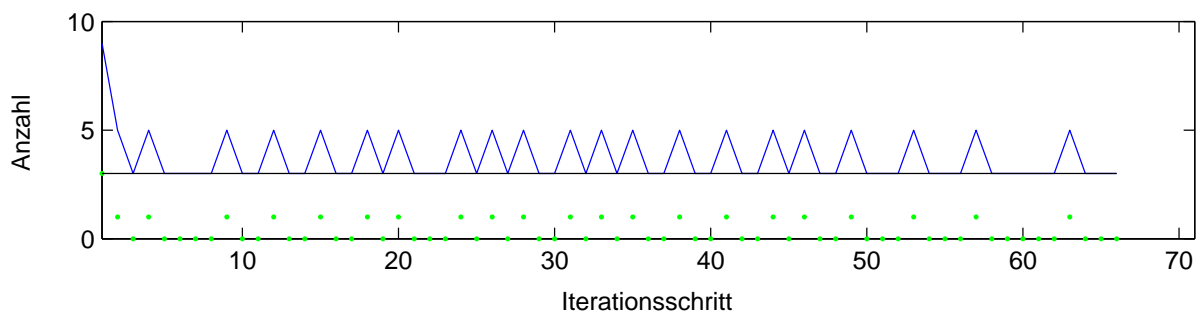


Abbildung 7.7.: Details für die Iteration bei einer Schrittweitensteuerung über den Vergleich der Lösungen von implizitem Euler mit ganzer und halber Schrittweite (vgl. auch Abb. 7.6):

Anzahl der gelösten Gleichungssysteme und der Knotenwechsel im  $m$ -ten Schritt.

Iterationsschritt zu lösen. Findet ein Wechsel des Grenzknotens statt, so müssen zwei zusätzliche Gleichungssysteme gelöst werden; wieder ein System bei der Lösung des impliziten Eulers und ein(!) weiteres System für den ersten oder zweiten Schritt des impliziten Eulers mit halber Schrittweite. Wie man sieht, finden die Wechsel bei beiden Verfahren simultan

statt. Als weitere Bemerkung ergibt sich:

**Bemerkung 7.3.2. Wechsel des Grenzknotens bei Crank-Nicolson**

*Den Details des Iterationsablaufes kann man entnehmen, dass die numerisch ermittelte Ausübungsgrenze  $x_{nb}$  für das implizite Euler-Verfahren nicht mit der des Crank-Nicolson-Verfahrens übereinstimmt. Beim Vergleich der beiden Lösungen ergibt sich daher für den Knoten  $x_{nb_{impl}+1}$  in jedem Fall ein numerischer Fehler. Dieser führt bei kleinen Werten der lokalen Toleranz  $tol$  zu dem in Kapitel 7.1.2 beschriebenen Absturz der gewählten Schrittweite und damit auch zu einer Beendigung des Algorithmus.*

## 8. Leistungsangabe und Vergleich

In diesem Kapitel soll nun der Leistungsgewinn, im Sinne einer Beschleunigung der Berechnung, dokumentiert werden, der durch die Schrittweisensteuerung erzielt wird. Zuvor jedoch sollen Kriterien formuliert werden, die die Güte der Lösung quantifizieren.

### 8.1. Kriterien für die Genauigkeit der Lösung

#### 8.1.1. Genauigkeit des Preises

Wie schon im Kapitel 2.3 angekündigt, greifen wir nochmals auf die Resultate von Merton zurück. Wir benutzen das qualitative Resultat:

**Lemma 8.1.1. *Konvexität des Preises [Mer73]***

*Der Preis  $V(S, t, K)$  einer amerikanischen Vanilla Option ist sowohl konvex im Kurs  $S$  als auch konvex im Basispreis  $K$ .*

**Beweis:**

Beweise zu Theorem 4 und Theorem 10 der zitierten Arbeit. □

Aus den Vorbereitungen zum Beweis dieses Theorems ergeben sich jedoch weitere Aussagen, die sich quantitativ ausnutzen lassen.

**Lemma 8.1.2. *Homogenität des Preises in Strike und Kurs [Mer73]***

*Der Optionswert  $V(S, t, K)$  einer amerikanischen Vanilla Option ist simultan homogen im Strike  $K$  und Aktienkurs  $S$ . Dies bedeutet, dass für  $\lambda > 0$  gilt:*

$$V(\lambda S, t, \lambda K) = \lambda V(S, t, K). \tag{8.1}$$

**Beweis:**

Vgl. Theorem 9 der zitierten Arbeit. □

Aus diesem Lemma lässt sich direkt das nachfolgende Korollar ableiten:

**Korollar 8.1.3. *Homogenität des Preises im Strike nach der Transformation***

*Sowohl für die Konvektions-Verschiebung als auch für die neue Transformation gilt: Nach vorgenommener Transformation ist der Optionswert  $v(x, \tau)$  homogen im Strike  $K$ . Für  $\lambda > 0$  gilt dann:*

$$v(x, \tau, \lambda K) = \lambda v(x, \tau, K). \tag{8.2}$$

**Beweis:**

Zunächst kann die Invarianz von  $x$  gegenüber  $\lambda$ -Vielfachen von  $S$  und  $K$  gezeigt werden. Diese folgt durch Einsetzen von  $\lambda S$  und  $\lambda K$  aus der Definition der Transformationen über

$$x := \ln \left( \frac{S}{K} \right) + f(t, T, r, \delta, \sigma).$$

Des Weiteren gilt  $v(x, t, K) = e^{+\int_0^\tau r(t-y) dy} V(S, \tau, K)$  für die Konvektions-Verschiebung bzw.  $v(x, t, K) = e^{\int_0^\tau r(t-y) dy - \frac{1}{2}x} V(S, \tau, K)$  für die neue Transformation. Mit dem Lemma 8.1.2 folgt die Behauptung.  $\square$

Das Korollar 8.1.3 lässt sich auf zweierlei Weise ausnutzen:

**Bemerkung 8.1.4. Möglichkeiten, die Homogenität auszunutzen**

i) Durch die Wahl von  $\lambda = \frac{1}{K}$  lässt sich eine Klasse von Optionen mit unterschiedlichen Preisen aus einer Option mit Strike 1 berechnen. Denn es gilt:

$$v(x, \tau, K) = K v(x, \tau, 1). \tag{8.3}$$

ii) Die Homogenität lässt sich ebenso für eine Konsistenzprüfung der Schrittweisensteuerung nutzen. Berechnet man sowohl  $v_1 := v(x, \tau, \lambda K)$  als auch  $v_2 := v(x, \tau, K)$  mittels einem der Algorithmen, so sollten sich die Lösungen punktweise um den Faktor  $\lambda$  unterscheiden. Anders gesagt, das  $\lambda$ -fache der zweiten Lösung sollte sich nicht gravierend von der ersten Lösung unterscheiden.

### 8.1.2. Genauigkeit der Ausübungsgrenze

Die Ausübungsgrenze lässt sich nach dem Buch von Kwok über eine Integralgleichung darstellen. Für einen Put gilt [Kwo98, S. 161]:

$$\begin{aligned} S_f^P(\tau) - K &= K e^{-r\tau} \mathcal{N}(-d_2) - S_f^P(\tau) e^{-\delta\tau} \mathcal{N}(-d_1) \\ &\quad + \int_0^\tau r K e^{-ry} \mathcal{N}(-\hat{d}_2(y, \tau)) - \delta S_f^P(\tau) e^{-\delta y} \mathcal{N}(-\hat{d}_1(y, \tau)) dy. \end{aligned} \tag{8.4}$$

Analog gilt nach Kwok [Kwo98, S. 163] die Gleichung

$$\begin{aligned} K - S_f^C(\tau) &= S_f^C(\tau) e^{-\delta\tau} \mathcal{N}(d_1) - K e^{-r\tau} \mathcal{N}(d_2) \\ &\quad + \int_0^\tau \delta S_f^C(\tau) e^{-\delta y} \mathcal{N}(\hat{d}_1(y, \tau)) - r K e^{-ry} \mathcal{N}(\hat{d}_2(y, \tau)) dy \end{aligned} \tag{8.5}$$

für den Call. In beiden Fällen sind die Parameter für passendes  $S_f$  über

$$\begin{aligned} d_1 &:= \frac{\ln \left( \frac{S_f(\tau)}{K} \right) + (r - \delta)\tau}{\sigma\sqrt{\tau}} + \frac{\sigma}{2}\sqrt{\tau}, & d_2 &:= d_1 - \sigma\sqrt{\tau}, \\ \hat{d}_1 &:= \frac{\ln \left( \frac{S_f(\tau)}{S_f(\tau-y)} \right) + (r - \delta)y}{\sigma\sqrt{y}} + \frac{\sigma}{2}\sqrt{y}, & \hat{d}_2 &:= \hat{d}_1 - \sigma\sqrt{y} \end{aligned}$$

definiert. Damit lässt sich eine ähnliche Aussage für die Ausübungsgrenze formulieren:



**Lemma 8.1.5. Homogenität der Ausübungsgrenze**

Die Ausübungsgrenze  $S_f$  der amerikanischen Vanilla Option erfüllt sowohl für den Put als auch für den Call für  $\lambda > 0$  die Homogenitätsaussage

$$S_f(\tau, \lambda K) = \lambda S_f(\tau, K). \quad (8.6)$$

**Beweis:**

Für den Beweis ist die Formulierung der Zeit  $t$  oder  $\tau$  unerheblich. Das Einsetzen von  $\lambda S_f$  und  $\lambda K$  in die Integralgleichung (8.4) für den Put bzw. (8.5) für den Call liefert die Aussage.  $\square$

Auch hier lässt sich nach der Transformation eine stärkere Aussage ableiten:

**Korollar 8.1.6. Unabhängigkeit von  $x_f$  vom Strike**

Nach der Transformation hängt die Ausübungsgrenze nicht mehr vom Strike  $K$  der Option ab. Statt dessen ist  $x_f$  in den  $(x, \tau)$ -Koordinaten nur noch eine Funktion von  $\tau$ ,

$$x_f = x_f(\tau). \quad (8.7)$$

**Beweis:**

Es gilt für beide Transformationen

$$x := \ln\left(\frac{S}{K}\right) + f(\tau, T, r, \delta, \sigma).$$

Damit gilt für die Ausübungsgrenze

$$x_f(\tau, K) = \ln\left(\frac{S_f(\tau, K)}{K}\right) + f(\tau, T, r, \delta, \sigma).$$

Sei nun  $\hat{K} > 0$  ein beliebiger zweiter Strikepreis. Dann existiert ein  $\lambda > 0$  mit  $\hat{K} = \lambda K$ . Für die entsprechende Ausübungsgrenze gilt

$$\begin{aligned} \hat{x}_f(\tau, \hat{K}) &= \ln\left(\frac{\hat{S}_f(\tau, \hat{K})}{\hat{K}}\right) + f(\tau, T, r, \delta, \sigma) \\ &\stackrel{\text{Lemma 8.1.5}}{=} \ln\left(\frac{\lambda S_f(\tau, K)}{\lambda K}\right) + f(\tau, T, r, \delta, \sigma) \\ &= \ln\left(\frac{S_f(\tau, K)}{K}\right) + f(\tau, T, r, \delta, \sigma) = x_f(\tau, K). \end{aligned}$$

Die Aufsprungpunkte sind also identisch. Damit ist der Aufsprungpunkt für alle Strikepreise  $K$  gleich.  $\square$

Auch hier lässt sich eine entsprechende Bemerkung ergänzen. Wie Korollar 8.1.3 ein Konsistenzkriterium für den Preis angibt, so liefert das Korollar 8.1.6 ein Konsistenzkriterium für die Ausübungsgrenze:

**Bemerkung 8.1.7. Auszunutzen der Unabhängigkeit vom Strike**

Berechnet man eine Option für verschiedene Strikepreise  $K$ , so sollten die Approximationen der Ausübungsgrenzen in den  $(x, \tau)$ -Koordinaten um weniger als die Maschenweite  $\Delta x$  voneinander abweichen.

## 8.2. Konsistenztest

Bisher ist lediglich das Verhalten der Algorithmen untersucht worden. Über die Qualität der ermittelten Lösung ist bisher kaum ein Wort verloren worden. Nur zu der Konvexitätsaussage aus Lemma 8.1.1 ist Stellung bezogen worden. Allerdings ist hier das Ergebnis eher qualitativ. Die Transformationen bewirken, dass keine numerischen Oszillationen durch Dispersion auftreten. Mittels des Rannacher smoothing werden aufgrund der geringeren Glattheit an der Ausübungsgrenze  $S_f$  auftretende Oszillationen ausgeschaltet. Nun soll untersucht werden, wie gut die Qualität der Lösung ist, insbesondere inwiefern der Fehler der numerischen Näherung vom gewählten Beispiel abhängt. Der Begriff ‚Konsistenz‘ wird hier für die Widerspruchsfreiheit der Lösung verwendet. Bei gleicher gewünschter Anzahl korrekter Dezimalstellen und unterschiedlicher Skalierung des Problems soll die ‚gleiche‘ Lösung geliefert werden.

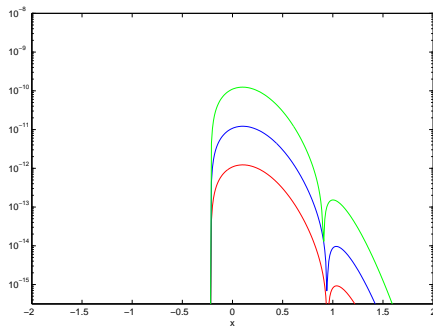
### 8.2.1. Optionswert für verschiedene Strikepreise

Für Optionen mit identischen Parametern, jedoch unterschiedlichen Strikepreisen  $K$ , sollte die Aussage des Korollars 8.1.3 erfüllt sein. Wir nutzen die Idee aus Bemerkung 8.1.4 und berechnen ein Beispiel für unterschiedliche Strikes  $K$  mit den verschiedenen Algorithmen. Damit sollen die Algorithmen auf ihre Widerspruchsfreiheit getestet werden.

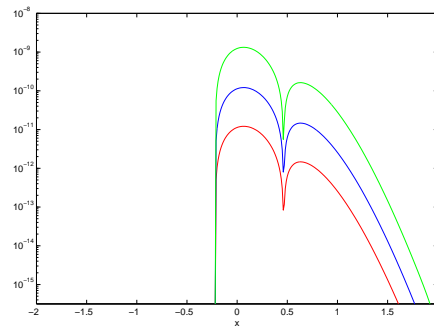
Für das implizite Euler-Verfahren ohne Schrittweitensteuerung fällt dieser Test positiv aus. Die Näherungen bzw. die entsprechenden Vielfachen zum Zeitpunkt  $\tau = T$  unterscheiden sich zumeist nur im Rahmen der Maschinengenauigkeit, höchstens jedoch um die Größenordnung  $10^{-12}$ . Sie erfüllen also in jedem Fall die gemachten Anforderungen.

Für die drei Steuerungsverfahren berechnen wir nochmals das Einstiegsbeispiel 7.1.3. Zur Erinnerung: Man berechne den Preis eines amerikanischen Vanilla Puts mit  $T = 1.0$ ,  $r = 0.1$ ,  $\delta = 0$  und  $\sigma = 0.3$  in der Diskretisierung  $x_{min} = -2$ ,  $x_{max} = 2$ ,  $\Delta x = \frac{1}{100}$ . Allerdings wird nun  $K$  und der Wert von  $tol$  so variiert, dass der Quotient  $\frac{tol}{K}$  konstant  $10^{-4}$  beträgt. Es wird die absolute lokale Toleranz über  $\|u_I - u_{II}\|$  gemessen. Die Anzahl der auf diese Weise korrekt bestimmter Dezimalstellen soll in allen Fällen gleich sein. Wir wählen  $K$  als 0.1, 1, 10 und 100 und führen die Berechnung durch. Die berechneten Werte werden nun für die Lösung bei  $\tau = T$  miteinander verglichen und die Gültigkeit des Korollars 8.1.3 geprüft. Dabei sei  $K = 1$  der Normalfall und die abweichenden Werte werden über  $\frac{1}{K}$  skaliert.

Die Abbildung 8.1 zeigt das Ergebnis dieser Berechnung. Eine exakte Erfüllung der



(a) EJ-Algorithmus



(b) Implizit/Halbe Vergleich

Abbildung 8.1.: Konsistenzprüfung für den Optionswert mit Korollar 8.1.3 am Beispiel 7.1.3 mit  $x_{min} = -2$ ,  $x_{max} = 2$ ,  $\Delta x = \frac{1}{100}$ : Vergleich von  $K = 1$  mit  $K = 0.1$ ,  $K = 10$  und  $K = 100$  bei konstantem Verhältnis  $\frac{tol}{K} = 10^{-4}$ .

Gleichung aus dem Korollar tritt nicht ein. Dies wurde allerdings auch nicht erwartet, denn die Aussage des Korollars bezieht sich auf die exakte Lösung. Hier handelt es sich jedoch um Näherungslösungen. In allen Fällen bleibt der Unterschied unter  $10^{-8}$ , der Algorithmus liefert aufgrund der Einstellungen jedoch lediglich 5 richtige Dezimalstellen der Lösung. Somit bleibt die Anwendung des Algorithmus frei von Widersprüchen.

Betrachtet man die Unterschiede im Detail, so sieht man, dass diese für die Steuerung über den Lösungszuwachs, EJ-Algorithmus, geringer ausfallen als bei den beiden anderen Verfahren. Diese sind untereinander in etwa vergleichbar, weshalb der Vergleich zwischen den Lösungen des impliziten Euler-Verfahrens mit ganzer und halber Schrittweite, dem so genannten ‚implizit/Halbe Vergleich‘, als Stellvertreter herangezogen wird. Grund für den geringeren Unterschied beim EJ-Algorithmus ist die höhere Anzahl von Iterationsschritten bei diesem Verfahren, denn es findet in jedem Schritt eine weitere Glättung des numerischen Fehlers durch das L-stabile Verfahren statt. Insbesondere wird der Fehler aus den ersten Iterationsschritten, Übergang der stetigen auf eine differenzierbare Lösung, weiter geglättet. Eine Beobachtung, die durch die größere Breite des Bereiches um das Kurvenmaximum unterstrichen wird. Für die Zuwachsteuerung reicht dieser Bereich bis etwa  $x = 1.0$ , bei den beiden anderen Verfahren reicht er nur bis  $x = 0.5$ . Für  $x < 0$  ist die linke Seite des Bereiches bei allen drei Verfahren durch die Ausübungsgrenze fixiert.

Verändert man die Parameter, so ändern sich auch die Resultate dieser Konsistenzprüfung entsprechend. Wird der Quotient von  $\frac{K}{tol}$  erhöht, so vergrößert sich ebenfalls der Unterschied zwischen den skalierten Näherungslösungen. Für eine Änderung der Maschenweite  $\Delta x$  bleiben die Werte konstant. Die Näherungslösung  $u$  ist also für die Verwendung eines Algorithmus konsistent.

### 8.2.2. Ausübungsgrenze für verschiedene Strikepreise

Bisher haben wir gesehen, dass die verschiedenen Algorithmen hinsichtlich des Optionswertes konsistent im Sinne der Anwendung von Korollar 8.1.3 sind. Wir wollen dies nun für die Ausübungsgrenze prüfen. Die entsprechende Aussage für die theoretische Ausübungsgrenze ist bereits in Korollar 8.1.6 hergeleitet worden. Wir testen, ob sich die Aussage einer in  $K$  invarianten Ausübungsgrenze  $x_f$  halten lässt.

Wir wandeln dazu das Einstiegsbeispiel durch eine höhere Volatilität ab, in der Hoffnung, dadurch neue Effekte sehen zu können. Der Berechnung liegt nun das nachfolgende Beispiel in der Diskretisierung  $x_{min} = -2$ ,  $x_{max} = 2$  und  $\Delta x = \frac{1}{100}$  bzw.  $\frac{1}{1000}$  sowie einem Verhältnis  $\frac{tol}{K} = 10^{-4}$  zu Grunde.

#### Beispiel 8.2.1.

*Man berechne den Preis eines amerikanischen Vanilla Puts mit  $K = 0.1, 1.0, 10.0, 100.0$ ,  $T = 1.0$ ,  $r = 0.1$ ,  $\delta = 0$  und  $\sigma = 0.8$ .*

Für das implizite Euler-Verfahren ohne Steuerung ist die Aussage des Korollars uneingeschränkt erfüllt. Die Ausübungsgrenze stimmt für jedes  $\tau$  zwischen den Näherungen für verschiedene  $K$  überein.

Verwendet man die Algorithmen mit Schrittweitensteuerung so gilt diese Aussage mit Einschränkungen. Es wird meist die gleiche Anzahl von Zeitschritten für alle  $K$  durchgeführt, evtl. tritt eine Abweichung von einem Schritt in der Gesamtzahl auf, jedoch entsprechen sich bei gleicher Anzahl nicht alle Zeitschritte. Für die gleiche Schrittnummer sind die Schrittweiten für verschiedene  $K$  unterschiedlich lang. Folge davon ist, dass die Verschiebung um den Konvektionsterm unterschiedlich stark greift (man vergleiche nochmals die Definition der Transformationen). Die auftretenden Unregelmäßigkeiten sind jedoch so gering, dass sie bei einem Plot der verschiedenen Approximationen der Ausübungsgrenze nicht offensichtlich sind. Aus diesem Grund zeigen wir auch keine Grafik. Quantitativ bedeutet dies einen Unterschied der Größenordnung  $10^{-4}$ , also genau der geforderten Dezimalstellenzahl. D.h. bei gleicher Anzahl korrekt zu bestimmender Stellen liefern die Algorithmen dieselbe Approximation der Ausübungsgrenze.

Macht man einen ergänzenden Test mit dem Verhältnis  $\frac{tol}{K} = 10^{-3}$  so treten allerdings in der Nähe von  $\tau = 0$  Abweichungen für große  $K$  auf. Hier springt die Ausübungsgrenze an einigen Stellen auf den nächsten Knoten.

### 8.3. Genauigkeitsvergleich

Im vorangehenden Abschnitt wurde gezeigt, dass die Methoden in sich stimmig sind. Nun soll dargelegt werden, dass die Methoden mit Schrittweitensteuerung sowohl untereinander als auch mit dem impliziten Euler-Verfahren ohne Steuerung vergleichbare Lösungen generieren. Dazu nutzen wir das bereits zitierte Konvergenzresultat von Jaillet, Lamberton und Lapeyre aus Satz 2.2.2. Wir wählen relativ kleine Werte von  $\Delta x$  und  $\Delta \tau$  und berechnen

eine Näherungslösung  $u_{JLL}$ . Nach der Aussage dieses Satzes und unter Vernachlässigung des Rundungsfehlereinflusses kann diese Lösung dann als ‚genau‘ angesehen werden. Der Begriff ‚genau‘ ist nicht sehr präzise, aber weil es keine analytische Vergleichslösung gibt, muss in jedem Falle eine numerische Approximation für den Vergleich benutzt werden.

**Bemerkung 8.3.1. Quervergleich mit europäischer Option**

- i) Auch wenn der Vergleich seine Schwächen hat und das Problem durch die fehlende Komplementarität deutlich einfacher wird, so kann doch für das gleiche Vorgehen bei einer europäischen Option eine Genauigkeitsangabe gemacht werden. Eine Approximation im Rahmen von  $\Delta x = 10^{-5}$  und  $\Delta \tau = 10^{-5}$  liefert etwa 5 richtige Dezimalstellen.*
- ii) Die Extrapolation der Ergebnisse für die amerikanische Option liefert für ein Gitter der gleichen Feinheit eine ähnliche Genauigkeit.*

Somit berechnen wir die Lösung  $u_{JLL}$  mit  $\Delta x = 10^{-5}$  und  $\Delta \tau = 10^{-5}$  in den Intervallgrenzen  $x_{min} = -2$  und  $x_{max} = 2$ . Dies erfolgt erneut für das

**Beispiel 8.3.2.**

*Man berechne den Preis eines amerikanischen Vanilla Puts mit  $K = 1.0$ ,  $T = 1.0$ ,  $r = 0.1$ ,  $\delta = 0$  und  $\sigma = 0.3$ .*

Die Wahl von  $K = 1$  ist dabei aufgrund der vorher durchgeführten Konsistenzuntersuchung repräsentativ. Diese Lösung vergleichen wir nun mit den Ergebnissen verschiedener anderer Diskretisierungen mit und ohne Schrittweitensteuerung.

Für den Vergleich mit  $u_{JLL}$  werden Lösungen verwendet, die mittels der drei Verfahren die Zeitschrittweite auf einem Gitter der Feinheit  $\Delta x = 10^{-4}$  und einer gleich großen lokalen Genauigkeit  $tol$  berechnet wurden. Bei den Ergebnissen unterscheiden wir zwischen dem Optionswert und der Approximation der Ausübungsgrenze.

Wir verwenden an dieser Stelle die folgenden Bezeichnungen:

- $u_{halb}$  für die Lösung, die über die Steuerung mittels implizitem Euler-Verfahren bei Vergleich von ganzer und halber Schrittweite berechnet wurde.
- $u_{CN}$  für die Lösung, deren Schrittweite über den Vergleich von implizitem Euler-Verfahren und Crank-Nicolson-Verfahren gesteuert wurde.
- $u_{EJ}$  für die Lösung, deren Schrittweite über den Lösungszuwachs gesteuert wurde.

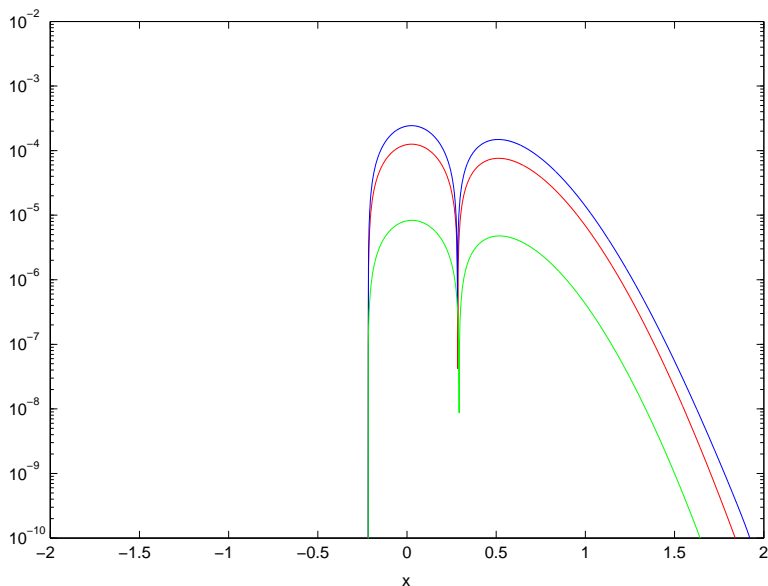


Abbildung 8.2.: Vergleich der Optionswerte für verschiedene Steuerungen für das Beispiel 8.3.2 mit  $x_{min} = -2$ ,  $x_{max} = 2$ ,  $\Delta x = \frac{1}{1000}$ : Vergleich des absoluten Unterschieds von  $u_{JLL}$  zu  $u_{halb}$ ,  $u_{CN}$  und  $u_{EJ}$ .

### 8.3.1. Optionswert für verschiedene Steuerungen

Zunächst vergleichen wir den Optionswert für die drei alternativen Steuerungsverfahren, um zu sehen, ob die ‚exakte‘ Lösung  $u_{JLL}$  hinreichend genau approximiert wurde. Dies wird in der Abbildung 8.2 wiedergeben.

Dort sieht man, dass  $u_{EJ}$  die beste Näherung abgibt. Dies erklärt sich, wenn man die Anzahl der durchgeführten Schritte betrachtet. Man vergleiche dazu die Tabelle 8.1.

Lösung	CPU-Sekunden	Zeitschritte
$u_{halb}$	107.28	71
$u_{CN}$	86.88	74
$u_{EJ}$	300.24	1532

Tabelle 8.1.: Kurzvergleich der Steuerungsverfahren am Beispiel 8.3.2

Die Anzahl der verwendeten Schritte ist deutlich höher als bei den beiden anderen Algorithmen. Da aber in jedem Schritt mit der Berechnung des impliziten Euler-Schrittes auch eine Glättung der Lösung stattfindet (das implizite Euler-Verfahren ist  $L$ -stabil), wirkt sich die zusätzliche Zahl der Schritte positiv aus. Aus demselben Grund ist die Lösung  $u_{halb}$  besser als die Lösung  $u_{CN}$ . Hier wird mit der Näherung, die mit der halben Schrittweite bestimmt wurde, weitergerechnet. Es werden daher doppelt so viele Glättungsschritte vorgenommen. Absolut liegt diese Lösung knapp über der lokalen Toleranzschwelle von  $10^{-4}$ .

### 8.3.2. Ausübungsgrenze für verschiedene Steuerungen

Mit gleichbleibenden Bezeichnungen wird für die Ausübungsgrenze eine analoge Betrachtung vorgenommen. Bei der Lösungsberechnung mit den drei Steuerungsalgorithmen im vorangegangenen Abschnitt wurde ebenfalls die Approximation der Ausübungsgrenze festgehalten. Alle vier Approximationen werden in der Abbildung 8.3 geplottet. Die beiden Detailvergrößerungen zeigen als erstes die gute Wiedergabe der optimalen Ausübungsgrenze der Lösung  $u_{JLL}$  durch die Approximation des EJ-Algorithmus. Dabei ist zu beachten, dass es sich bei der ‚exakten‘ Grenze nur um eine genauere Approximation handelt. Außerdem ist das Gitter hier in  $x$ -Richtung um den Faktor 10 gröber gewählt worden. Für die feinen Schritte zu Anfang der Iteration ( $\tau < 0.02$ ) ist keine Differenz zu sehen und auch am Ende der Iteration ( $\tau > 0.8$ ) ist die Differenz zwischen der schwarzen und grünen Kurve auf das in  $x$ -Richtung gröbere Gitter zurückzuführen. In der Totalen ist die schwarze, die Ausübungsgrenze von  $u_{JLL}$  repräsentierende, Kurve daher weggelassen worden.

Für die beiden anderen, zu  $u_{halb}$  bzw.  $u_{CN}$ , gehörenden Ausübungsgrenzen ergibt sich ein nicht so exakter Verlauf. Dieser ist in der Totalen kaum zu erkennen. Die Detailvergrößerung für kleine  $\tau$  zeigt, dass diese Abweichung durch die im Vergleich zum EJ-Algorithmus wesentlich größeren Zeitschritte auftritt. Ferner ist anhand dieser Vergrößerung ebenfalls die unterschiedliche Anzahl von Zeitschritten ersichtlich. Die roten und blauen Stützstellen liegen nicht auf den gleichen  $\tau$ -Werten. Diese Beobachtung lässt sich auch bei der zweiten Detailvergrößerung für  $\tau > 0.8$  machen. Dort zeigt sich ebenso ein bzgl.  $\tau$  verzögertes Nachlaufen der Ausübungsgrenze der Lösung, die über die Steuerung mittels Vergleich zwischen implizitem Euler und Crank-Nicolson gewonnen wurde, zur Grenze der Lösung aus dem Schrittweitenvergleich. Wir notieren:

#### **Bemerkung 8.3.3. Verwendbarkeit der Ausübungsgrenze**

*Die mittels Schrittweitenvergleich bzw. Ordnungsvergleich gewonnene Ausübungsgrenze ist nur eingeschränkt verwendbar. Sie stellt aufgrund der langen Schrittweiten oft nur eine grobe Approximation dar.*

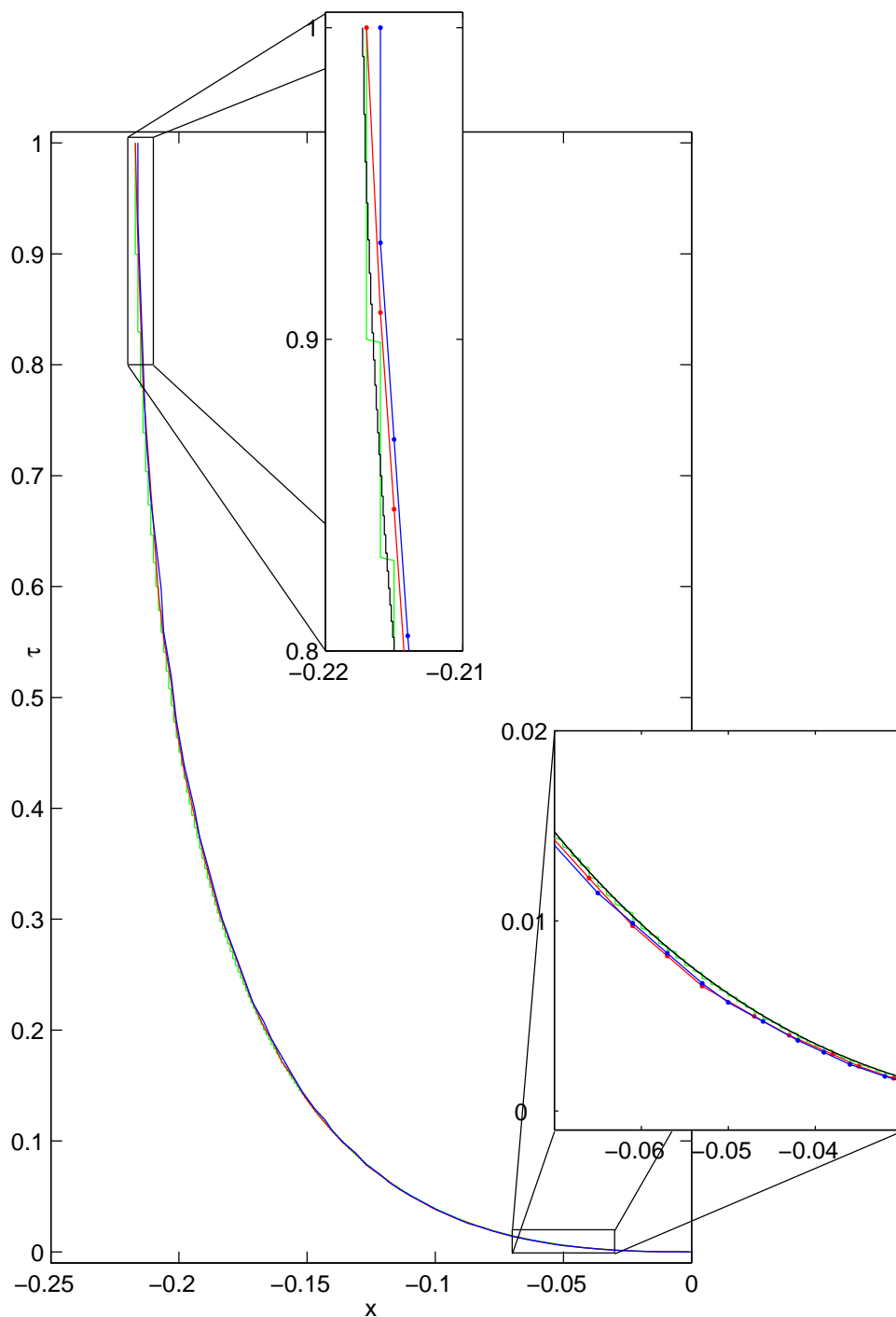


Abbildung 8.3.: Ausübungsgrenze für verschiedene Steuerungen für das Beispiel 8.3.2 mit  $x_{min} = -2$ ,  $x_{max} = 2$ ,  $\Delta x = \frac{1}{1000}$ : In der Totalen ist der Verlauf für  $u_{halb}$ ,  $u_{CN}$  und  $u_{EJ}$  angegeben. Die beiden Ausschnitte geben die markierten Bereiche vergrößert wieder. Dort kommt zusätzlich die Approximation für  $u_{JLL}$  als ‚exakte‘ Grenze hinzu. Ebenfalls zusätzlich sind dort für die Approximationen  $u_{halb}$  und  $u_{CN}$  Punktemarker an den Stützstellen gesetzt worden.



## 8.4. Leistungszuwachs

Nachdem gezeigt wurde, dass mittels der Algorithmen, die eine Steuerung der Zeitschrittweite vornehmen, hinreichend genaue Lösungen berechnet werden, wenden wir uns nun dem numerischen Aufwand zu. Dieser gibt Aufschluss darüber, ob ein praktischer Einsatz der Verfahren sinnvoll ist. Wir verwenden erneut das Beispiel 8.3.2, um uns die Laufzeiten und den Aufwand der Algorithmen anzusehen. Dabei verwenden wir, jeweils für  $x_{min} = -2$  und  $x_{max} = 2$ , die Diskretisierungen:

$$\text{Diskretisierung a) } \Delta x = \frac{1}{100}, \Delta \tau_{max} = 10^{-1.0} ,$$

$$\text{Diskretisierung b) } \Delta x = \frac{1}{1000}, \Delta \tau_{max} = 10^{-1.5} .$$

### 8.4.1. Laufzeiten

Für die Berechnung des Beispiels 8.3.2 mit Diskretisierung a) ergeben sich die Laufzeiten in CPU-Sekunden<sup>1</sup> gemäß der Tabelle 8.2. Entsprechend gibt Tabelle 8.3 die Resultate für die Diskretisierung b) wieder. Neben den drei Algorithmen und deren Laufzeiten für verschiedene Werte der lokalen Toleranz  $tol$  werden auch die Laufzeiten für 100 bzw. 1000 Schritte des impliziten Euler-Verfahrens mit konstanter Schrittweite angegeben.

Für das implizite Euler-Verfahren ist  $\Delta \tau$  dann konstant  $\frac{1}{100}$  bzw.  $\frac{1}{1000}$ . Eine Aussage über die Genauigkeit der Lösung ist für dieses Verfahren nicht möglich. Allerdings stellt der Aufwand des impliziten Eulers eine Untergrenze für die Programmierung eines Lösungsverfahrens dar. Es wird genau eine Diskretisierung mittels Finiter-Differenzen erstellt und berechnet. Das Verfahren geht, bis auf die zeitschrittweise Abfolge der Lösung, nicht auf die Eigenheiten des Problems ein.

Die beiden Tabellen zeigen, dass sich sowohl das Verfahren mit Schrittweitenvergleich und zugehöriger Lösung  $u_{halb}$  als auch das Verfahren mit Vergleich der Diskretisierungsordnung und Lösung  $u_{CN}$  im Bezug auf die Laufzeit für beide Beispiele deutlich unterhalb der Laufzeit von 1000 äquidistanten impliziten Euler-Schritten bewegen. Dabei liegen beide Verfahren nahe an der für 100 implizite Euler-Schritte benötigten Zeit, im ersten Beispiel sogar teilweise darunter.

Ebenso deutlich zeigt sich ein erheblicher Anstieg der Laufzeit für eine Schrittweitensteuerung über den Lösungszuwachs bei niedrigerer lokaler Toleranz  $tol$ . Ein Anstieg der Laufzeit ist auch bei den beiden anderen Verfahren zu beobachten, jedoch fällt er wesentlich schwächer aus. Bei dem verwendeten feineren Gitter in  $x$ -Richtung schneidet der EJ-Algorithmus für  $tol = 10^{-2}$  erstaunlicherweise wesentlich besser ab als die beiden anderen Verfahren. Diese qualitativen Resultate sollen auch auf der Ebene der durchgeführten Operationen untersucht werden, um sie genauer erklären zu können. Die bisherigen Ergebnisse lassen sich zur nachfolgenden Beobachtung zusammenfassen:

<sup>1</sup>Bemerkungen zur Vergleichbarkeit finden sich in Anhang B

$tol$	$u_{halb}$	$u_{CN}$	$u_{EJ}$
$10^{-2}$	2.34	1.77	1.56
$10^{-2.5}$	2.50	1.95	2.33
$10^{-3}$	2.85	2.09	4.69
$10^{-3.5}$	3.73	3.15	12.04
$10^{-4}$	4.70	3.40	40.51
	100 Schritte impliziter Euler		3.43
	1000 Schritte impliziter Euler		26.44

Tabelle 8.2.: Vergleich der Laufzeiten in CPU-Sekunden für die Algorithmen am Beispiel 8.3.2 mit der Diskretisierung a)

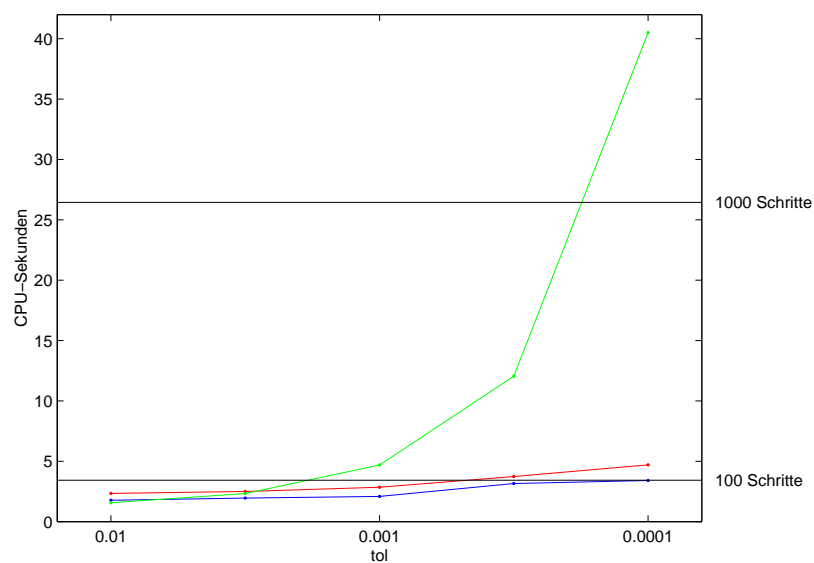


Abbildung 8.4.: Visualisierung der Laufzeiten aus Tabelle 8.2: Die Laufzeiten sind für  $u_{halb}$ ,  $u_{CN}$  und  $u_{EJ}$  in CPU-Sekunden angegeben. Zur Orientierung sind die Laufzeiten für das implizite Euler-Verfahren mit 100 bzw. 1000 äquidistanten Schritten eingezeichnet.

$tol$	$u_{halb}$	$u_{CN}$	$u_{EJ}$
$10^{-2}$	84.17	54.85	40.19
$10^{-2.5}$	85.36	<i>Error</i>	45.14
$10^{-3}$	91.24	<i>Error</i>	62.50
$10^{-3.5}$	98.11	75.88	112.17
$10^{-4}$	107.98	86.55	300.24
	100 Schritte impliziter Euler		46.26
	1000 Schritte impliziter Euler		234.41

Tabelle 8.3.: Vergleich der Laufzeiten in CPU-Sekunden für die Algorithmen am Beispiel 8.3.2 mit der Diskretisierung b). Für die mit *Error* bezeichneten Einträge brach der Algorithmus ab.

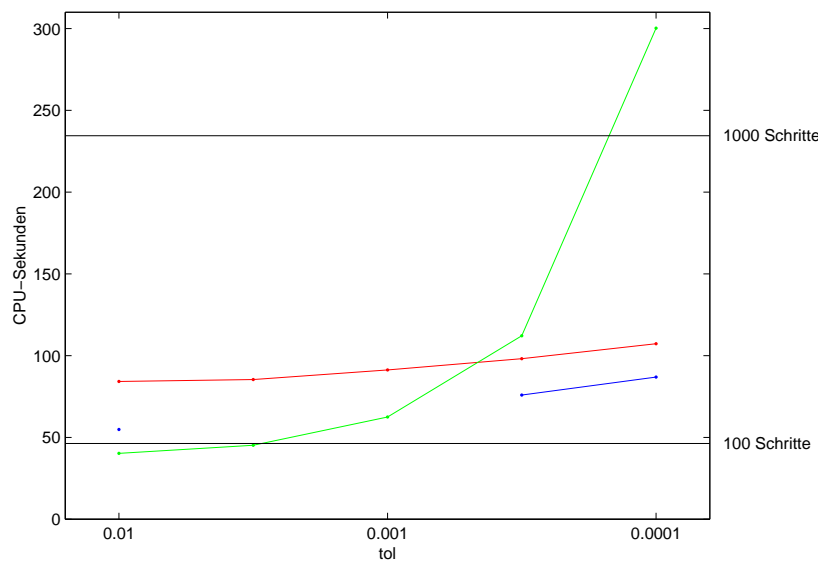


Abbildung 8.5.: Visualisierung der Laufzeiten aus Tabelle 8.3: Die Laufzeiten sind für  $u_{halb}$ ,  $u_{CN}$  und  $u_{EJ}$  in CPU-Sekunden angegeben. Zur Orientierung sind die Laufzeiten für das implizite Euler-Verfahren mit 100 bzw. 1000 äquidistanten Schritten eingezeichnet.

**Beobachtung 8.4.1. Laufzeitvergleich:**

- i) Die Steuerung der Zeitschrittweite ermöglicht eine schnellere Berechnung des Optionswertes, als mit den üblichen 1000 äquidistanten Schritten und Berechnung über das implizite Euler-Verfahren.*
- ii) Der Zugewinn der Verfahren fällt für ein grobes Gitter deutlich größer aus; in unserem Beispiel etwa um einen Faktor 6 für die Diskretisierung mit  $\Delta x = \frac{1}{100}$  gegenüber dem Faktor 3 für  $\Delta x = \frac{1}{1000}$ .*
- iii) Bezüglich der benötigten Laufzeit schneidet der Ordnungsvergleich am besten ab. Allerdings wird diese Position mit dem Nachteil möglicher Ausfälle durch die Verwendung der Crank-Nicolson-Vergleichslösung, wie im zweiten Beispiel, erkauft.*
- iv) Die Steuerung über den Lösungszuwachs wird mit geringerer Toleranz wesentlich langsamer. Allerdings schneidet der EJ-Algorithmus für feines  $\Delta x$  und niedrige lokale Toleranz *tol* gut ab.*

**8.4.2. Durchgeführte Operationen**

Aufgrund der Messergebnisse ist es sinnvoll, die Schrittweitensteuerung anzuwenden. Wir wollen die Beobachtungen 8.4.1 ii) klären und schauen uns für beide Diskretisierungen weitere Details an. Dazu werden die Anzahlen der

- i) Schrittweitenwechsel (SW-Wechsel)
- ii) durchgeführten Zeitschritte
- iii) Wechsel des Ausübungsknotens (NB-Wechsel)

für verringerte Toleranz und verfeinertes Gitter ermittelt und der Aufwand mittels des Satzes 5.3.2 überschlagen. Wir erinnern daran, dass für einen Wechsel der Schrittweite Matrizen besetzt und zerlegt werden müssen. Ebenso erfordert sowohl jeder Zeitschritt als auch jeder NB-Wechsel das Lösen von Gleichungssystemen. Im günstigsten Fall ist dies ein System, im ungünstigsten bis zu drei Systeme. Zu den überschlagsweise benötigten Operationen vergleiche man nochmals die Tabelle 6.1 (S. 87). Insbesondere sei erneut darauf hingewiesen, dass sich die Zahlen auf vollbesetzte Tridiagonalmatrizen beziehen. Die Matrix  $D$  des diskretisierten Komplementaritätsproblems besteht aber nur ab der Spalte  $nb+1$  aus einer Tridiagonalmatrix. In den Spalten 1 bis  $nb$  hingegen stehen die Einträge aus der negativen  $N \times N$ -Einheitsmatrix. Der Aufwand für die Lösung der Gleichungssysteme wird somit überschätzt. Dieser Tatsache und evtl. auch dem Zugriff auf Unterprogramme wird am Beispiel  $tol = 10^{-2}$ ,  $\Delta x = 10^{-2}$  dadurch Rechnung getragen, dass die Umsetzung des Ordnungsvergleichs zwar rechnerisch mehr Operationen benötigt als der Schrittweitenvergleich, auf dem Rechner jedoch weniger CPU-Sekunden erfordert.

Lösung	Aktion	Anzahl	Aufwand
$u_{halb}$	SW-Wechsel	7	$7 \cdot 20 \cdot N = 140 \cdot N$ $7 \cdot 6 \cdot N = 42 \cdot N$
	Schritte	15	$15 \cdot 33 \cdot N = 495 \cdot N$
	NB-Wechsel	24	$24 \cdot 22 \cdot N = 528 \cdot N$
	Summe		$1205 \cdot N$
$u_{CN}$	SW-Wechsel	8	$8 \cdot 13 \cdot N = 104 \cdot N$ $8 \cdot 6 \cdot N = 48 \cdot N$
	Schritte	16	$16 \cdot 27 \cdot N = 432 \cdot N$
	NB-Wechsel	24	$24 \cdot 27 \cdot N = 648 \cdot N$
	Summe		$1232 \cdot N$
$u_{EJ}$	SW-Wechsel	10	$10 \cdot 10 \cdot N = 100 \cdot N$ $10 \cdot 3 \cdot N = 30 \cdot N$
	Schritte	19	$19 \cdot 11 \cdot N = 209 \cdot N$
	NB-Wechsel	24	$24 \cdot 11 \cdot N = 264 \cdot N$
	Summe		$603 \cdot N$

Tabelle 8.4.: Übersichtsweise Anzahl von Operationen für die Algorithmen am Beispiel 8.3.2. Dabei wird die Diskretisierung a), d.h.  $\Delta x = 10^{-2}$  mit  $N = 401$ , für eine lokale Toleranz  $tol = 10^{-2}$  verwendet.

Wenn wir nun die Ergebnisse vergleichen, so versuchen wir Effekte zu isolieren, die hauptsächlich an der Verschärfung der Toleranzforderung oder aber an der Verfeinerung des Gitters festzumachen sind.

Der Vergleich der drei Tabellen zeigt zunächst eine Tatsache, die eigentlich sofort aus der Aufgabenstellung hervorgeht:

**Bemerkung 8.4.2. Unabhängigkeit der NB-Wechsel vom Verfahren**

- i) Die Anzahl der Wechsel des Ausübungsknotens ist unabhängig vom gewählten Verfahren zur Steuerung der Zeitschrittweite. Dies wird sofort klar, wenn man bedenkt, dass durch diese Wechsel im Wesentlichen das Auffinden der implizit in der Problemstellung enthaltenen Ausübungsgrenze wiedergegeben wird.
- ii) Als Kennzahl für die durchgeführten NB-Wechsel lässt sich die Anzahl der Gitterpunkte  $x_i$  angeben, die während der Laufzeit der Option einen Ausübungsknoten wiedergeben. Bei einer monotonen verlaufenden Grenze ist dies die Mächtigkeit der Menge

$$\{x_i\} \cap \{x_f(\tau) : 0 < \tau \leq T\}.$$

Man sieht sofort, dass bei einem äquidistanten Gitter die zehnfache Verfeinerung des Gitters eine in etwa Verzehnfachung dieser Anzahl bedeutet.

Lösung	Aktion	Anzahl	Aufwand
$u_{halb}$	SW-Wechsel	14	$14 \cdot 20 \cdot N = 280 \cdot N$ $14 \cdot 6 \cdot N = 84 \cdot N$
	Schritte	23	$23 \cdot 33 \cdot N = 759 \cdot N$
	NB-Wechsel	24	$24 \cdot 22 \cdot N = 528 \cdot N$
	Summe		$1651 \cdot N$
$u_{CN}$	SW-Wechsel	14	$14 \cdot 13 \cdot N = 182 \cdot N$ $14 \cdot 6 \cdot N = 84 \cdot N$
	Schritte	23	$23 \cdot 27 \cdot N = 621 \cdot N$
	NB-Wechsel	24	$24 \cdot 27 \cdot N = 648 \cdot N$
	Summe		$1535 \cdot N$
$u_{EJ}$	SW-Wechsel	12	$12 \cdot 10 \cdot N = 120 \cdot N$ $12 \cdot 3 \cdot N = 36 \cdot N$
	Schritte	156	$156 \cdot 11 \cdot N = 1716 \cdot N$
	NB-Wechsel	24	$24 \cdot 11 \cdot N = 264 \cdot N$
	Summe		$2136 \cdot N$

Tabelle 8.5.: Übersichtsweise Anzahl von Operationen für die Algorithmen am Beispiel 8.3.2. Dabei wird die Diskretisierung a), d.h.  $\Delta x = 10^{-2}$  mit  $N = 401$ , für eine lokale Toleranz  $tol = 10^{-3}$  verwendet.

Für ein weiter verfeinertes Gitter stellen diese Wechsel des Ausübungsknotens, im Gegensatz zu der Bezeichnung ‚zusätzlicher Aufwand‘ in Kapitel 5.3.2, den Hauptaufwand des Lösungsvorgangs dar. Laut Tabelle 8.6 werden hierfür bei jedem der drei Verfahren mehr als 80 Prozent der Operationen aufgewandt. Je feiner die Maschenweite in  $x$ -Richtung gewählt wird, desto vorteilhafter zeigt sich hier die Steuerung über den Wertzuwachs (EJ-Algorithmus), denn dort ist für einen solchen Wechsel nur ein Gleichungssystem zu lösen.

Anhand des Schrittweitenvergleichs wird dies nochmals in der Abbildung 8.6 dargestellt. Dort ist deutlich die Durchführung von sehr vielen NB-Wechseln am Beginn der Iteration zu sehen, die letztendlich auch für den Hauptanteil der gelösten Gleichungssysteme verantwortlich sind. Zum Teil werden je Zeitschritt mehr als dreißigmal so viele Gleichungssysteme wie normal bearbeitet. Die Ausübungsgrenze verändert sich in diesem Bereich sehr schnell und muss ‚verfolgt‘ werden.

**Bemerkung 8.4.3. Aufwand für das Verfolgen der Ausübungsgrenze**

Für ein sehr feines Gitter stellt das Verfolgen der Ausübungsgrenze bei allen drei Verfahren den Hauptaufwand des Algorithmus dar. Mit einer evtl. auch nur teilweisen Kenntnis des Verlaufes der Ausübungsgrenze könnte die Anzahl der benötigten Operationen beträchtlich reduziert werden. In diesem Falle könnte der NB-Wechsel in den auszuführenden Zeitschrit-

Lösung	Aktion	Anzahl	Aufwand
$u_{halb}$	SW-Wechsel	8	$8 \cdot 20 \cdot N = 160 \cdot N$ $8 \cdot 6 \cdot N = 48 \cdot N$
	Schritte	37	$37 \cdot 33 \cdot N = 1221 \cdot N$
	NB-Wechsel	218	$218 \cdot 22 \cdot N = 4796 \cdot N$
	Summe		$6225 \cdot N$
$u_{CN}$	SW-Wechsel	9	$9 \cdot 13 \cdot N = 117 \cdot N$ $9 \cdot 6 \cdot N = 54 \cdot N$
	Schritte	38	$38 \cdot 27 \cdot N = 1026 \cdot N$
	NB-Wechsel	218	$218 \cdot 27 \cdot N = 5886 \cdot N$
	Summe		$7083 \cdot N$
$u_{EJ}$	SW-Wechsel	10	$10 \cdot 10 \cdot N = 100 \cdot N$ $10 \cdot 3 \cdot N = 30 \cdot N$
	Schritte	38	$38 \cdot 11 \cdot N = 418 \cdot N$
	NB-Wechsel	218	$218 \cdot 11 \cdot N = 2398 \cdot N$
	Summe		$2946 \cdot N$

Tabelle 8.6.: Übersichtsweise Anzahl von Operationen für die Algorithmen am Beispiel 8.3.2. Dabei wird die Diskretisierung b), d.h.  $\Delta x = 10^{-3}$  mit  $N = 4001$ , für eine lokale Toleranz  $tol = 10^{-2}$  verwendet.

ten zumindest teilweise integriert werden.

Leider sind gebräuchliche asymptotische Entwicklungen der Ausübungsgrenze nur lokal genau. Sie eignen sich für diesen Zweck daher nicht.

Der Vergleich der Tabellen 8.4 und 8.5 macht im Wesentlichen das unterschiedliche Vorgehen bei der Steuerung über den Ordnungsvergleich und den Schrittweitenvergleich auf der einen und der Zuwachssteuerung auf der anderen Seite deutlich. Innerhalb des Verfahrens unterscheiden sich die zu den Werten gehörenden Probleme nur in einer zehnfachen Verschärfung der lokalen Toleranz.

Sowohl für den Schrittweitenvergleich als auch für den Ordnungsvergleich mit zugeordneter Lösung  $u_{halb}$  bzw.  $u_{CN}$  erhöht sich die Anzahl der Schrittweitenwechsel durch die strengere Toleranzgrenze deutlich. Die Verdopplung der Wechsel schlägt aber nur in einem fünfzigprozentigem Zuwachs der Gesamtanzahl der Iterationsschritte nieder. Ein Indiz dafür, dass sich die Verfeinerung vor allem auf die ersten Iterationsschritte auswirkt. Die Glättung durch das  $L$ -stabile implizite Euler-Verfahren lässt die Unterschiede zwischen den Vergleichslösungen im fortschreitenden Verfahren sehr schnell kleiner werden.

Im Gegensatz dazu steht das Verhalten der Steuerung über den Lösungszuwachs. Hier wird anhand des Unterschiedes zwischen der Lösung vor und nach Durchführung eines Zeit-

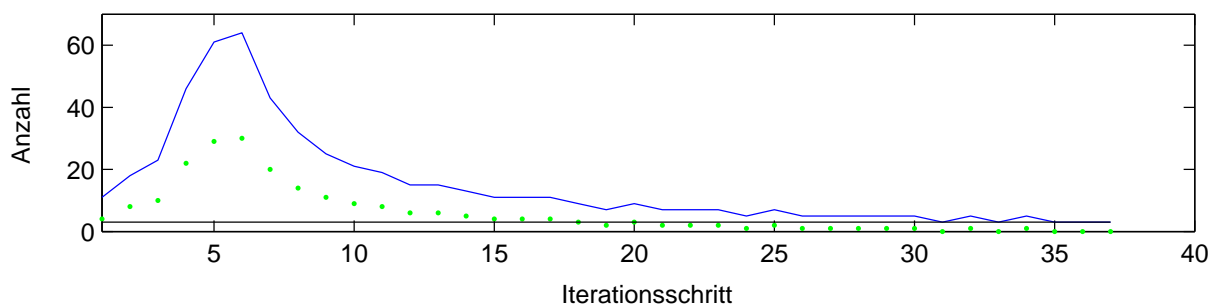


Abbildung 8.6.: Details für die Iteration bei einer Schrittweitensteuerung über den Vergleich der Lösungen von implizitem Euler mit ganzer und halber Schrittweite auf einem sehr feinen Gitter (vgl. Tabelle 8.6). Dazu sei im Vergleich auf das Iterationsverhalten aus der Abbildung 7.7 hingewiesen. Es werden die Anzahl der **gelösten Gleichungssysteme** und der **Knotenwechsel** im  $m$ -ten Schritt dargestellt.

schrittes gesteuert. Dieser Unterschied ist durch das Problem gegeben. Eine Verfeinerung der Toleranz um den Faktor zehn bedeutet nichts anderes, als den zulässigen Unterschied um das Zehnfache zu reduzieren. Als grobe Approximation für das Abklingen der Lösung kann der betragsmäßig größte Eigenwert herangezogen werden. Die Lösung verhält sich für große  $\tau$  etwa wie

$$v(x, \tau) \sim \exp\left(-\frac{\sigma^2}{2}\lambda_{max}^2\tau\right)C(x)$$

mit einer nicht näher spezifizierten von  $x$  abhängigen Konstante  $C$ . Weil  $C(x)$  unabhängig von  $\tau$  ist, kann man die Approximation

$$\begin{aligned} v(x, \tau + \Delta\tau) - v(x, \tau) &\approx \exp\left(-\frac{\sigma^2}{2}\lambda_{max}^2\tau\right)C(x)\left(\exp\left(-\frac{\sigma^2}{2}\lambda_{max}^2\Delta\tau\right) - 1\right) \\ &\doteq \exp\left(-\frac{\sigma^2}{2}\lambda_{max}^2\tau\right)C(x)\left(-\frac{\sigma^2}{2}\lambda_{max}^2\Delta\tau\right) \end{aligned}$$

heranziehen. Eine Verzehnfachung der lokalen Toleranz bedeutet daher in etwa die Durchführung von zehnmal so vielen Zeitschritten wie zuvor. Für die Problemstellung ist  $\tau$  aber kleiner oder gleich 1. Dennoch lässt sich für den EJ-Algorithmus ein ähnliches Verhalten beobachten. Mit der zehnfachen Verfeinerung der Toleranz steigt die Anzahl der Zeitschritte um den Faktor 8. Die Laufzeiten für diesen Algorithmus (vgl. Abb. 8.4) sprechen genauso für diesen Sachverhalt.



### 8.4.3. Zusammenfassende Bemerkungen

Die Ergebnisse bezüglich Aufwand und Laufzeit der drei Steuerungsalgorithmen lassen sich in mehreren Bemerkungen zusammenfassen.

#### **Bemerkung 8.4.4. Verwenden des Ordnungsvergleichs**

*Von der Steuerung der Zeitschrittweite über den Vergleich der Diskretisierungsordnungen ist im speziellen Fall, dem Vergleich mit der  $\mathcal{O}(\Delta\tau^2)$ -Lösung des Crank-Nicolson-Verfahrens, Abstand zu nehmen. Dieses Verfahren verursacht durchschnittlich zwar den wenigsten Aufwand und benötigt deshalb die geringste Laufzeit, bedingt jedoch den Nachteil eines möglichen Verfahrensabbruchs ohne Lösung.*

Die beiden anderen Verfahren können jedoch zur Steuerung der Zeitschrittweite verwendet werden. Wir sprechen dafür die folgende Empfehlung aus:

#### **Bemerkung 8.4.5. Lösungszuwachs- oder Schrittweitenvergleich**

- i) Eine Verwendung der Schrittweitensteuerung ist hinsichtlich des numerischen Aufwandes in jedem Fall sinnvoll.*
- ii) Für den Schrittweitenvergleich wird die Berechnung auf einem feinen Gitter durch den Wechsel des Ausübungsknotens für beide Lösungen deutlich teurer. Hier geht die Anzahl der Gitterpunkte gegenüber der Steuerung über den Lösungszuwachs mit dem Faktor zwei ein.*
- iii) Die Steuerung über den Lösungszuwachs hat den Nachteil, dass mit verkleinerter Toleranz sehr schnell die Anzahl der durchgeführten Zeitschritte ansteigt.*
- iv) Der in Punkt iii) beschriebene Nachteil ist ein Vorteil für die Ermittlung einer numerischen Ausübungsgrenze. Die Zeitschichten liegen näher hintereinander und die Approximation der Grenze fällt genauer aus.*
- v) Um den heutigen Wert einer Option auf einem nicht zu feinen Gitter zu bestimmen, ist es sinnvoll, die Zeitschrittweite über den Vergleich zweier Lösungen des impliziten Euler-Verfahrens bei unterschiedlicher Schrittweite zu steuern.*
- vi) Für ein Gitter mit geringer Maschenweite und zur Approximation der Ausübungsgrenze sollte man die Steuerung über den Lösungszuwachs vorziehen.*



# 9. Anwendbarkeit auf Exoten

Die Resultate der vorangehenden Kapitel beziehen sich, theoretisch wie numerisch, ausschließlich auf Vanilla Optionen amerikanischen Typs. Dies sind Optionen auf ein Underlying mit einem Ausübungspreis, der am Preis dieses Underlyings festgemacht wird. Eine interessante Frage ist nun, ob man die Resultate auf allgemeinere Optionen bzw. kompliziertere Konstrukte übertragen kann. Hier bieten sich zum einen Basket Optionen an, d.h. statt einem Underlying wird die Auszahlung nun an mehrere Wertpapiere (z.B. Aktien) geknüpft. Eine andere Richtung wird durch die Betrachtung von Optionen asiatischen Typs eingeschlagen. In diesem Falle zählt nicht allein der aktuelle Kurs für die Auszahlung, sondern auch ein Durchschnitt dieses Kurses über ein zurückliegendes Zeitintervall.

## 9.1. Basket Optionen

### 9.1.1. Definition

Wie schon in der Einleitung dieses Kapitels angeklungen ist, sind diese Optionen nicht nur von einem Underlying abhängig, sondern beziehen sich auf mehrere Aktien. Basket Optionen werden für zwei Aktien z.B. in [ZFV01, Sey04] betrachtet. Einen allgemeinen Fall mit einem Korb von  $n$  Aktien findet man im Buch von Kwok [Kwo98, Kapitel 3]. Das in Kapitel 1.1 vorgestellte Modell wird hier erweitert.

Für jede Aktie  $S_i$  gilt die Kursentwicklung

$$dS_i = \mu_i S_i dt + \sigma_i S_i dW_i \quad ; \quad i = 1, \dots, n \quad (9.1)$$

mit der dem eindimensionalen Fall entsprechenden Drift  $\mu_i$  und Volatilität  $\sigma_i$ . Die risikofreie Zinsrate bleibt für alle Aktien gleich  $r$ , allerdings zahlt jede Aktie eine unterschiedliche Dividende  $\delta_i$ . Jeweils zwei Aktien  $i$  und  $j$  sind über den Parameter  $\rho_{ij}$  korreliert. Unter diesen Voraussetzungen sowie den aus dem eindimensionalen Fall übernommenen Marktannahmen ergibt sich die  $n$ -dimensionale Black-Scholes Gleichung

$$V_t + \sum_{i=1}^n (r - \delta_i) S_i V_{S_i} + \sum_{i=1}^n \frac{\sigma_i^2}{2} S_i^2 V_{S_i S_i} + \sum_{\substack{i,j=1 \\ i \neq j}}^n \rho_{ij} \frac{\sigma_i \sigma_j}{2} S_i S_j V_{S_i S_j} - rV = 0. \quad (9.2)$$

Der Payoff der Option kann dabei auf unterschiedliche Weise beschrieben werden. Denkbar sind z.B. so genannte Index-Optionen mit Payoff

$$\Psi(S_1, \dots, S_n, t) := \max \left( \sum_{i=1}^n \alpha_i S_i - K, 0 \right) \quad ; \quad \alpha_i \geq 0, \quad \sum_{i=1}^n \alpha_i = 1 \quad (9.3)$$

oder aber Maximums- bzw. Minimums-Optionen mit Payoff der Form

$$\Psi(S_1, \dots, S_n, t) : = \max \left( \max_i(S_i) - K, 0 \right), \quad (9.4)$$

$$\Psi(S_1, \dots, S_n, t) : = \max \left( \min_i(S_i) - K, 0 \right). \quad (9.5)$$

Die Liste der Beispiele lässt sich beliebig erweitern.

### 9.1.2. Übertragbarkeit der Theorie

Die Frage ist nun, ob sich die vorgestellten Transformationen auf diese Erweiterung übertragen lassen. Eine Untersuchung für die Konvektions-Verschiebung findet sich in meiner Arbeit über die Vermeidung numerischer Dispersion für die Optionsbewertung [IV05]. Die dort nicht behandelte neue Transformation wird gesondert untersucht.

#### Konvektions-Verschiebung

Wir folgen hier dem Aufbau aus [IV05, Abschnitt 5] und versuchen zunächst das Resultat für die eindimensionale Konvektions-Verschiebung auf jede einzelne der Aktien anzuwenden. Die eindimensionale Transformation

$$\begin{aligned} \tau &:= T - t, \\ x &:= \ln \left( \frac{S}{K} \right) + \int_t^T r(y) - \delta(y) - \frac{\sigma^2(y)}{2} dy \end{aligned}$$

wird über die Zeitumkehr

$$\tau := T - t \quad (9.6)$$

und eine Verschiebung für jede Aktie  $S_i$  der Form

$$x_i := \ln \left( \frac{S_i}{K} \right) + \int_t^T r(y) - \delta_i(y) - \frac{\sigma_i^2(y)}{2} dy \quad ; \quad i = 1, \dots, n \quad (9.7)$$

auf die  $n$ -dimensionale Gleichung 9.2 übertragen. In den neuen Koordinaten lautet die Gleichung:

$$0 = v_\tau - \sum_{i=1}^n \frac{\sigma_i^2}{2} v_{x_i x_i} - \sum_{\substack{i,j=1 \\ i \neq j}}^n \rho_{ij} \frac{\sigma_i \sigma_j}{2} v_{x_i x_j} + rv. \quad (9.8)$$

Auch hier kann man mittels Finiter-Differenzen eine numerische Näherung bestimmen. Diese Approximation bezeichnen wir wieder mit  $u$ . Um die Zuordnung der Notation zu den einzelnen  $x_i$  zu vereinfachen verwenden wir einen Multiindex  $k_1, \dots, k_n$  statt des für die Numerik verwendeten fortlaufenden Indexes. Damit ergeben sich die Approximationen

$$v \approx \Theta u_{k_1, \dots, k_n}^{(m+1)} + (1 - \Theta) u_{k_1, \dots, k_n}^{(m)}, \quad (9.9)$$

$$v_\tau \approx \frac{u_{k_1, \dots, k_n}^{(m+1)} - u_{k_1, \dots, k_n}^{(m)}}{\Delta \tau}, \quad (9.10)$$

$$v_{x_i x_i} \approx \Theta \frac{u_{k_1, \dots, k_i+1, \dots, k_n}^{(m+1)} - 2u_{k_1, \dots, k_i, \dots, k_n}^{(m+1)} + u_{k_1, \dots, k_i-1, \dots, k_n}^{(m+1)}}{(\Delta x_i)^2} + (1 - \Theta) \frac{u_{k_1, \dots, k_i+1, \dots, k_n}^{(m)} - 2u_{k_1, \dots, k_i, \dots, k_n}^{(m)} + u_{k_1, \dots, k_i-1, \dots, k_n}^{(m)}}{(\Delta x_i)^2}, \quad (9.11)$$

$$v_{x_i x_j} \approx \Theta \left[ \frac{u_{k_1, \dots, k_i+1, \dots, k_j+1, \dots, k_n}^{(m+1)} - u_{k_1, \dots, k_i-1, \dots, k_j+1, \dots, k_n}^{(m+1)}}{\Delta x_i \Delta x_j} + \frac{u_{k_1, \dots, k_i-1, \dots, k_j-1, \dots, k_n}^{(m+1)} - u_{k_1, \dots, k_i+1, \dots, k_j-1, \dots, k_n}^{(m+1)}}{\Delta x_i \Delta x_j} \right] + (1 - \Theta) \left[ \frac{u_{k_1, \dots, k_i+1, \dots, k_j+1, \dots, k_n}^{(m)} - u_{k_1, \dots, k_i-1, \dots, k_j+1, \dots, k_n}^{(m)}}{\Delta x_i \Delta x_j} + \frac{u_{k_1, \dots, k_i-1, \dots, k_j-1, \dots, k_n}^{(m)} - u_{k_1, \dots, k_i+1, \dots, k_j-1, \dots, k_n}^{(m)}}{\Delta x_i \Delta x_j} \right], \quad i \neq j. \quad (9.12)$$

Ebenso werden die diskreten Fourier-Moden auf

$$\tilde{u}_{k_1, \dots, k_n}^{(m)} = \hat{u} e^{i\omega m \Delta \tau} \prod_{j=1}^n e^{i\beta k_j \Delta x_j} \quad (9.13)$$

verallgemeinert. Damit kann nun eine zum Eindimensionalen analoge Dispersions-Analyse stattfinden. Mit Blick auf die Gleichung 9.2 unterscheiden wir den Fall von unkorrelierten Aktien, d.h.  $\rho_{ij} = 0$  für alle  $i \neq j$ , und den korrelierten Fall.

Im unkorrelierten Fall sind sämtliche Aktien voneinander unabhängig und die Differenzenschemata können in die einzelnen Dimensionen separiert werden. Dabei zeigt sich sofort die Dispersionsfreiheit des Differenzens-Schemas.

Es bleibt im Hinblick auf den korrelierten Fall nur noch der Einfluss der gemischten zweiten Ableitungen zu untersuchen. Um die Notation zu vereinfachen, nehmen wir  $n = 2$  an. Damit

wird die einzig verbliebene gemischte Ableitung über

$$v_{x_1 x_2} \approx \Theta \frac{u_{k_1+1, k_2+1}^{(m+1)} - u_{k_1-1, k_2+1}^{(m+1)} - u_{k_1+1, k_2-1}^{(m+1)} + u_{k_1-1, k_2-1}^{(m+1)}}{\Delta x_1 \Delta x_2} \quad (9.14)$$

$$+ (1 - \Theta) \frac{u_{k_1+1, k_2+1}^{(m)} - u_{k_1-1, k_2+1}^{(m)} - u_{k_1+1, k_2-1}^{(m)} + u_{k_1-1, k_2-1}^{(m)}}{\Delta x_1 \Delta x_2} \quad (9.15)$$

approximiert. Wir nehmen zur weiteren Vereinfachung ein rein implizites Verfahren, d.h.  $\Theta = 1$ , an. Für

$$\frac{u_{k_1+1, k_2+1}^{(m+1)} - u_{k_1-1, k_2+1}^{(m+1)} - u_{k_1+1, k_2-1}^{(m+1)} + u_{k_1-1, k_2-1}^{(m+1)}}{\Delta x_1 \Delta x_2}$$

ergeben sich diskrete Fourier-Moden der Form:

$$\begin{aligned} & \frac{\widehat{u} e^{i\omega(m+1)\Delta\tau} e^{ik_1\Delta x_1} e^{ik_2\Delta x_2}}{\Delta x_1 \Delta x_2} (e^{i\Delta x_1} e^{i\Delta x_2} - e^{-i\Delta x_1} e^{i\Delta x_2} - e^{i\Delta x_1} e^{-i\Delta x_2} + e^{-i\Delta x_1} e^{-i\Delta x_2}) \\ = & \frac{\widehat{u} e^{i\omega(m+1)\Delta\tau} e^{ik_1\Delta x_1} e^{ik_2\Delta x_2}}{\Delta x_1 \Delta x_2} ((e^{i\Delta x_1} - e^{-i\Delta x_1}) e^{i\Delta x_2} - (e^{i\Delta x_1} - e^{-i\Delta x_1}) e^{-i\Delta x_2}) \\ = & \frac{\widehat{u} e^{i\omega(m+1)\Delta\tau} e^{ik_1\Delta x_1} e^{ik_2\Delta x_2}}{\Delta x_1 \Delta x_2} (2i \sin(\Delta x_1) 2i \sin(\Delta x_2)) \\ = & -4 \frac{\widehat{u} e^{i\omega(m+1)\Delta\tau} e^{ik_1\Delta x_1} e^{ik_2\Delta x_2}}{\Delta x_1 \Delta x_2} \sin(\Delta x_1) \sin(\Delta x_2). \end{aligned}$$

Der Imaginärteil ist hier identisch Null. Es treten also keine Probleme mit numerischer Dispersion auf. Damit lässt sich die Transformation von Barraquand und Pudet auf Basket Optionen übertragen.

## Neue Transformation

Wir versuchen, zunächst für 2 Aktien, die neue Transformation in analoger Weise auf ein Basket von Optionen zu übertragen. Gemäß der Idee

$$\begin{aligned} \tau & := T - t, \\ x_1 & := \ln\left(\frac{S_1}{K}\right) + \int_t^T r(y) - \delta_1(y) dy, \\ x_2 & := \ln\left(\frac{S_2}{K}\right) + \int_t^T r(y) - \delta_2(y) dy, \end{aligned}$$

ergibt sich analog zum Vorgehen in Kapitel 3.3.3 das Zwischenstadium

$$\begin{aligned} \widehat{V}_\tau - \frac{\sigma_1(x_1, x_2, \tau)^2}{2} [\widehat{V}_{x_1 x_1} - \widehat{V}_{x_1}] - \frac{\sigma_2(x_1, x_2, \tau)^2}{2} [\widehat{V}_{x_2 x_2} - \widehat{V}_{x_2}] \\ - \rho(x_1, x_2, \tau) \sigma_1(x_1, x_2, \tau) \sigma_2(x_1, x_2, \tau) \widehat{V}_{x_1 x_2} + r(T - \tau) \widehat{V} = 0. \end{aligned} \quad (9.16)$$

Der Ansatz der Form

$$\widehat{V}(x_1, x_2, \tau) = \exp(g(x_1, x_2) + h(\tau)) v(x_1, x_2, \tau) \quad (9.17)$$

liefert dann eine weitere PDE in  $v$ . Der Wunsch nach Konvektionstermen identisch zu Null führt auf Bedingungsgleichungen der Form

$$\sigma_1(x_1, x_2, \tau)(1 - 2g_{x_1}) + 2\rho(x_1, x_2, \tau)\sigma_2(x_1, x_2, \tau)g_{x_2} \stackrel{!}{=} 0, \quad (9.18a)$$

$$\sigma_2(x_1, x_2, \tau)(1 - 2g_{x_2}) + 2\rho(x_1, x_2, \tau)\sigma_1(x_1, x_2, \tau)g_{x_1} \stackrel{!}{=} 0. \quad (9.18b)$$

Diese Gleichungen sind bei unkorrelierten Kursen leicht zu lösen. Sie stellen unabhängige Erweiterungen des eindimensionalen Falles dar. Für korrelierte Aktien kann jedoch keine geschlossene Lösung angegeben werden. Die Schwierigkeit steckt in der Abhängigkeit der Parameter  $\rho$  und  $\sigma_i$  von den Variablen  $x_1$  und  $x_2$ .

**Bemerkung 9.1.1. Lösbarkeit für konstante Parameter**

*Über den Spezialfall konstanter Parameter denken wir an dieser Stelle nicht mehr nach, er ist bereits durch die Konvektions-Verschiebung abgedeckt. Die neue Transformation wurde im Eindimensionalen hergeleitet, um Fälle von kursabhängiger Volatilität zu behandeln.*

Die Verwendung der neuen Transformation bleibt somit auf eindimensionale Probleme limitiert.

**Bemerkung 9.1.2. Frage der Modellierung**

*Wir lassen die Frage offen, ob die Ermittlung der Parameter für die Aufgabenstellung hinreichend genau erfolgen kann. Die Ermittlung der Volatilität läuft auf die Lösung eines inversen Problems hinaus und ist damit sehr schlecht konditioniert. Insbesondere bleibt offen, ob es nicht sinnvoller ist, mit ausschließlich zeitabhängigen Parametern zu modellieren, dies evtl. auch über die Bildung eines Durchschnittswertes. In diesem Falle aber könnte man wieder die Konvektions-Verschiebung verwenden.*

### 9.1.3. Reduktion des Aufwandes

Wie wir gesehen haben, bereiten die gemischten Ableitungen bezüglich numerischer Dispersion keine Schwierigkeiten. ‚Störend‘ sind sie jedoch dann, wenn es darum geht, die entstehenden Gleichungssysteme zu lösen. Zwar lassen sich die entsprechenden Diskretisierungsmatrizen auch mit diesen gemischten Ableitungen aufstellen, jedoch benötigen die verwendeten Differenzensterne wesentlich mehr Einträge.

Am Beispiel der zweidimensionalen Gleichung

$$0 = V_\tau - \frac{\sigma_1^2}{2} V_{x_1 x_1} - \frac{\sigma_2^2}{2} V_{x_2 x_2} - \rho_1 \rho_2 \sigma_1 \sigma_2 V_{x_1 x_2} + rV \quad (9.19)$$

sehen die Differenzensterne für die Diffusion in  $x_1$  und  $x_2$  wie folgt aus:

$$\frac{\sigma_1^2}{2} \frac{1}{\Delta x_1^2} \begin{bmatrix} -1 & +2 & -1 \end{bmatrix}_{\Delta x_1} u_i^m + \frac{\sigma_2^2}{2} \frac{1}{\Delta x_2^2} \begin{bmatrix} -1 \\ +2 \\ -1 \end{bmatrix}_{\Delta x_2} u_i^m.$$

Für die Kreuzdiffusion nach  $x_1$  und  $x_2$  kommt noch der diskrete Operator

$$\rho_{12} \sigma_1 \sigma_2 \frac{1}{\Delta x_1 \Delta x_2} \begin{bmatrix} -1 & 0 & +1 \\ 0 & 0 & 0 \\ +1 & 0 & -1 \end{bmatrix}_{\Delta x_1 \Delta x_2} u_i^m$$

hinzu. Bei der Diskretisierung dieser Diffusionsgleichung entsteht eine dünnbesetzte Blockdiagonalmatrix. Wir betrachten diese Matrix für eine lexikografische Anordnung der Gitterpunkte im Diskretisierungsrechteck. Für jede gemischte zweite Ableitung kommen dabei vier weitere Diskretisierungspunkte, d.h. auch Matrixeinträge, hinzu. Gegenüber den  $2n + 1$  Einträgen für die unkorrelierte Gleichung werden nun also für je  $\frac{n(n-1)}{2}$  gemischte Ableitungen jeweils vier zusätzliche Punkte benötigt. Dies ergibt für das vollständig korrelierte Modell dann insgesamt  $2n^2 + 1$  Einträge je Matrixzeile statt der nur  $2n + 1$  Einträge für das unkorrelierte Modell. Damit macht es Sinn, die Transformation so abzuwandeln, dass sich eine Form ergibt, in der die gemischten Terme fehlen.

Hierfür sind in [IV05, Abschnitt 5.2] bereits Abwandlungen der Transformation angegeben worden. Diese wollen wir hier noch einmal betrachten. Für  $n = 2$  starten wir mit dem Ansatz:

$$\begin{aligned} \tau &:= T - t, \\ x_1 &:= a_{11} \ln\left(\frac{S_1}{K}\right) + a_{12} \ln\left(\frac{S_2}{K}\right) + (T - t) \widehat{R}_1, \\ x_2 &:= a_{21} \ln\left(\frac{S_1}{K}\right) + a_{22} \ln\left(\frac{S_2}{K}\right) + (T - t) \widehat{R}_2. \end{aligned} \tag{9.20}$$

Wir berechnen die partiellen Ableitungen wie in [IV05, Anhang A] und setzen diese in die Gleichung (9.2), mit  $n=2$ , ein. Nach dem Zusammenfassen der Terme ergibt sich die Gleichung:

$$\begin{aligned} 0 &= V_\tau + rV \\ &\quad - \left[ (r - \delta_1 - \frac{\sigma_1^2}{2}) a_{11} + (r - \delta_2 - \frac{\sigma_2^2}{2}) a_{12} - \widehat{R}_1 \right] V_{x_1} \\ &\quad - \left[ (r - \delta_1 - \frac{\sigma_1^2}{2}) a_{21} + (r - \delta_2 - \frac{\sigma_2^2}{2}) a_{22} - \widehat{R}_2 \right] V_{x_2} \\ &\quad - \left[ \frac{\sigma_1^2}{2} a_{11}^2 + \rho \sigma_1 \sigma_2 a_{11} a_{12} + \frac{\sigma_2^2}{2} a_{12}^2 \right] V_{x_1 x_1} \\ &\quad - \left[ \sigma_1^2 a_{11} a_{21} + \sigma_2^2 a_{12} a_{22} + \rho \sigma_1 \sigma_2 (a_{11} a_{22} + a_{12} a_{21}) \right] V_{x_1 x_2} \\ &\quad - \left[ \frac{\sigma_1^2}{2} a_{21}^2 + \rho \sigma_1 \sigma_2 a_{21} a_{22} + \frac{\sigma_2^2}{2} a_{22}^2 \right] V_{x_2 x_2}. \end{aligned} \tag{9.21}$$



Für die partiellen Ableitungen  $V_{x_1}$ ,  $V_{x_2}$  und  $V_{x_1x_2}$  sollen die Koeffizienten identisch Null sein. Dies führt auf ein unterbestimmtes Gleichungssystem mit sechs Unbekannten in den drei Gleichungen

$$\left(r - \delta_1 - \frac{\sigma_1^2}{2}\right) a_{11} + \left(r - \delta_2 - \frac{\sigma_2^2}{2}\right) a_{12} - \widehat{R}_1 \stackrel{!}{=} 0, \quad (9.22)$$

$$\left(r - \delta_1 - \frac{\sigma_1^2}{2}\right) a_{21} + \left(r - \delta_2 - \frac{\sigma_2^2}{2}\right) a_{22} - \widehat{R}_2 \stackrel{!}{=} 0, \quad (9.23)$$

$$\sigma_1^2 a_{11} a_{21} + \sigma_2^2 a_{12} a_{22} + \rho_{12} \sigma_1 \sigma_2 (a_{11} a_{22} + a_{12} a_{21}) \stackrel{!}{=} 0. \quad (9.24)$$

Hier sehen wir sofort die Abhängigkeit der Shift-Parameter  $\widehat{R}_1$  und  $\widehat{R}_2$  von jeweils nur einer einzigen Gleichung. Diese können aus diesem Grund zuletzt bestimmt werden. Wir beginnen daher mit dem Auflösen der unterbestimmten Gleichung (9.24). Dabei unterscheiden wir die zwei Fälle  $\rho_{12} = 0$  and  $\rho_{12} \neq 0$ , d.h. die Gleichungen

$$\sigma_1^2 a_{11} a_{21} + \sigma_2^2 a_{12} a_{22} \stackrel{!}{=} 0, \quad (9.25a)$$

$$\sigma_1^2 a_{11} a_{21} + \sigma_2^2 a_{12} a_{22} + \rho_{12} \sigma_1 \sigma_2 (a_{11} a_{22} + a_{12} a_{21}) \stackrel{!}{=} 0. \quad (9.25b)$$

Aus Gleichung (9.25a) folgen sofort die beiden hinreichenden Bedingungen  $a_{11} a_{21} \stackrel{!}{=} 0$  und  $a_{12} a_{22} \stackrel{!}{=} 0$ . Eine natürliche Wahl, die an den Fall  $n = 2$  der Gleichung (9.8) mit gemischten zweiten Ableitungen anschließt, ist dann  $a_{12} = a_{21} = 0$  und  $a_{11} = a_{22} = 1$ .

Die Gleichung (9.25b) können wir, in Anlehnung an den Spezialfall  $\rho = 0$ , durch die Wahl von  $a_{11} = a_{22} = 1$  und  $a_{12} = 0$  auf

$$\sigma_1^2 a_{21} + \rho_{12} \sigma_1 \sigma_2 \stackrel{!}{=} 0$$

vereinfachen. Hier folgt dann sofort

$$a_{21} = -\rho_{12} \frac{\sigma_2}{\sigma_1}.$$

Mit den so bestimmten vier  $a_{ij}$  lassen sich  $\widehat{R}_1$  und  $\widehat{R}_2$  aus den verbleibenden Gleichungen (9.22) und (9.23) zu

$$\widehat{R}_1 := \left(r - \delta_1 - \frac{\sigma_1^2}{2}\right) = R_1,$$

$$\widehat{R}_2 := -\rho_{12} \frac{\sigma_2}{\sigma_1} \left(r - \delta_1 - \frac{\sigma_1^2}{2}\right) + \left(r - \delta_2 - \frac{\sigma_2^2}{2}\right) = R_2 - \rho_{12} \frac{\sigma_2}{\sigma_1} R_1$$

mit  $R_i := r - \delta_i - \frac{\sigma_i^2}{2}$  für konstante Parameter  $r$ ,  $\delta$  und  $\sigma$  ermitteln. Wir sehen sofort, dass der Ansatz mit dem Gleichungssystem (9.20) nicht mit zeitabhängigen Parametern durchzuführen wäre. Am Beispiel des totalen Differentials für

$$V_t = V_{x_1} \frac{\partial x_1}{\partial t} + V_{x_2} \frac{\partial x_2}{\partial t} + V_\tau \frac{\partial \tau}{\partial t} \quad (9.26)$$

wird dies deutlich. Dort würde dann, statt dem bisher bestimmten Ergebnis  $\frac{\partial x_2}{\partial t} = -\widehat{R}_2$ , ein Ausdruck der Form

$$\frac{\partial x_2}{\partial t} = \frac{\partial a_{21}}{\partial t} \left( x_1 + \int_0^\tau r(T-y) - \delta_1(T-y) - \frac{\sigma_1(T-y)^2}{2} dy \right) - \widehat{R}_2 \quad (9.27)$$

stehen. Die an dieser Stelle entstehende Transformation würde jedoch so kompliziert werden, dass ihr Einsatz nicht mehr gerechtfertigt wäre. Der Vermeidung der zusätzlichen Matrixeinträge steht der Aufwand für die Auswertung des Payoffs gegenüber. Statt dessen geben wir noch einmal die vollständige Transformation für  $n = 2$  an. Sie lautet:

$$\begin{aligned} \tau &:= T - t, \\ x_1 &:= \ln\left(\frac{S_1}{K}\right) + (T-t)R_1, \\ x_2 &:= \ln\left(\frac{S_2}{K}\right) + (T-t)R_2 - \rho_{12} \frac{\sigma_2}{\sigma_1} \left( \ln\left(\frac{S_1}{K}\right) + (T-t)R_1 \right), \end{aligned}$$

mit der Bewertungsgleichung

$$V_\tau - \frac{\sigma_1^2}{2} V_{x_1 x_1} - \frac{\sigma_2^2}{2} (1 - \rho_{12}^2) V_{x_2 x_2} + r V = 0. \quad (9.28)$$

In meiner Arbeit [IV05] wurde diese Transformation noch auf drei Aktien erweitert. Wir verzichten auf eine Darstellung dieser Erweiterung und wollen uns statt dessen der numerischen Berechnung zuwenden.

## Numerisches Ergebnis

Mit der letzten Gleichung und den entsprechend transformierten Gleichungen haben wir für ein quaderförmiges Gebiet im  $(x_1, x_2, \tau)$ -Raum den Wert einer Putoption auf das Minimum zweier Aktien berechnet. Wir verwenden

### Beispiel 9.1.3. Put auf das Minimum zweier Aktien

Man berechne den Preis eines amerikanischen Puts auf das Minimum von  $S_1$  und  $S_2$  mit  $K = 5.0$ ,  $T = 1.0$ ,  $r = 0.15$ ,  $\delta_1 = \delta_2 = 0$ ,  $\sigma_1 = 0.2$ ,  $\sigma_2 = 0.3$  und  $\rho = 0.3$ .

Für dieses Beispiel ist die Auszahlungsfunktion

$$\Psi(S_1, S_2, t) := \max [K - \min[S_1, S_2], 0] \quad (9.29)$$

vorgegeben. In den transformierten Koordinaten werden für die Ränder des Berechnungsgebietes Dirichlet-Randbedingungen vorgegeben. Der Wert auf den Rändern für  $x_1 = x_{1 \min}$  oder  $x_1 = x_{1 \max}$  bzw.  $x_2 = x_{2 \min}$  oder  $x_2 = x_{2 \max}$  soll exakt dem Wert der Auszahlungsfunktion an diesen Stellen entsprechen. Der Ansatz von Dempster und Hutton, das Problem in ein Lineares Programm zu überführen, kann aus dem eindimensionalen Fall übernommen werden. Eine Aussage in Analogie zu Lemma 2.2.3 ergibt sich sofort. Es

reicht zu zeigen, dass die Nebendiagonaleinträge der Diskretisierungsmatrix negativ oder Null sind. Diese Bedingung ist aber durch die Diskretisierung erfüllt. Das Problem lässt sich damit für jeden Zeitschritt in ein Lineares Programm überführen.

Dieses Lineare Programm wird jedoch nicht, so wie es Dempster, Hutton und Richards [DHR98] im eindimensionalen Fall gemacht haben, strukturiert und mittels des DHR-Algorithmus gelöst. Statt dessen wird die zeitschrittweise Abfolge der Probleme festgelegt und das Problem für einen Zeitschritt diskretisiert. Für jeden Zeitschritt wird das diskrete zweidimensionale Problem dann mit CPLEX<sup>®</sup> [ILO99a], einem kommerziellen LP-Löser, gelöst. Dabei haben wir allerdings (noch) **keine** Zeitschrittweitensteuerung verwendet.

Für das Beispiel 9.1.3 ergeben sich Optionswerte, die in der Abbildung 9.1 veranschaulicht werden. Dem in die  $(S, t)$ -Koordinaten zurück transformierten Gitter sieht man deutlich die

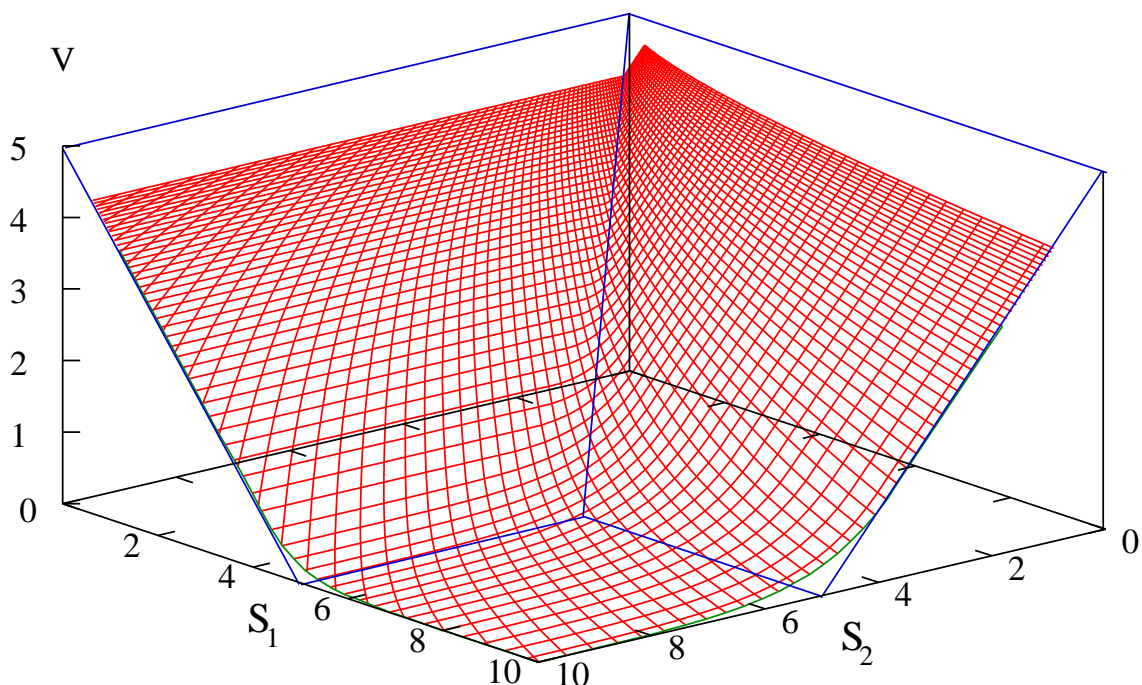


Abbildung 9.1.: Optionswert einer Basket-Option. Es handelt sich um einen Put auf das Minimum zweier Aktien. Die Parameter stammen aus dem Beispiel 9.1.3. Der Plot gibt die Wertfunktion zum Zeitpunkt  $\tau = T = 1$ , den Inneren Wert der Option sowie die Schnittkanten mit den Ebenen  $S_1 = 10$  bzw.  $S_2 = 10$  wieder.

Verzerrung in der  $S_2$ -Komponente an. Für die Berechnung wurden  $x_1$  und  $x_2$  als unabhängig voneinander gewählt, daher sind  $S_1$  und  $S_2$  dann mittels des  $a_{21}$  korreliert. Auf diesen Fakt ist ebenfalls schon in [IV05] hingewiesen worden. Qualitativ ergibt sich für die beiden einzelnen Komponenten ein ähnliches Verhalten wie für eine eindimensionale Option. Das



Gleichzeitig ist die Diskretisierungsungleichung eine zweidimensionale Wärmeleitungs(un)-gleichung. Auf dem Haltebereich gilt für die PDE Gleichheit, so dass wieder ein ähnliches Verhalten wie für die Wärmeleitungsgleichung zu erwarten ist. Die hochfrequenten Anteile der Lösung klingen sehr schnell ab, so dass für kleines  $\tau$  auch kleine Zeitschritte  $\Delta\tau$  durchzuführen sind.

Im Wesentlichen übertragen sich die Ideen aus dem eindimensionalen Fall. Wir halten dies in der nachfolgenden Bemerkung fest:

**Bemerkung 9.1.4.  $n$ -dimensionale Schrittweitensteuerung**

- i) Die Aussagen zur Verwendung des impliziten Euler bzw. Crank-Nicolson-Verfahrens aus Lemma 5.2.12 und Korollar 5.2.15 übertragen sich auf diesen Fall. Die Ausübungsgrenze ist hier nicht mehr ein Punkt  $x_f(\tau)$ , sondern eine von  $x_1, \dots, x_n$  abhängige Menge  $X_f(\tau)$ . Die nicht ausreichende Glattheit an dieser Grenze bleibt auch für  $n$  Dimensionen erhalten und erfordert für Rannacher smoothing wieder die Berechnung mittels eines  $L$ -stabilen Verfahrens (z.B. implizites Euler-Verfahren).*
- ii) Die Steuerungsalgorithmen für die Steuerung der Zeitschrittweite über den Lösungszuwachs (Algorithmus 6.1.2), den Vergleich der Diskretisierungsordnung (Algorithmus 6.1.6) sowie den Vergleich zweier Schrittweiten (Algorithmus 6.1.8) können von der Struktur her übernommen werden. Die Ermittlung der neuen Schrittweite bleibt unproblematisch, da zuvor bereits mit der Maximumsnorm (Korollar 7.1.2) gearbeitet wurde. Es muss nur der Löser für den einzelnen Zeitschritt, z.B. gegen die CPLEX<sup>®</sup>-Implementierung, ausgetauscht werden.*

Auf eine numerische Umsetzung für das mehrdimensionale Beispiel haben wir verzichtet.

## 9.2. Asiatische Optionen

### 9.2.1. Definition

Asiatische Optionen können mitunter auch als Basket-Option von mehreren Underlyings abhängen. Wir behandeln hier der Einfachheit halber nur den Fall eines Underlyings. Statt dessen wollen wir die Abhängigkeit der Auszahlungsfunktion von einem Durchschnitt als wichtigste Eigenschaft dieses Optionstyps herausstellen. Dieser Durchschnitt kann als diskretes oder kontinuierliches, arithmetisches oder geometrisches Mittel bestimmt werden (vgl. [WHD95, Kapitel 14], [Sey04, Kapitel 6.2]). An dieser Stelle wollen wir nur das diskrete arithmetische Mittel betrachten.

Es ist eine Folge von Zeitpunkten  $t_1 < t_2 < \dots < t_J = T$  gegeben, an denen das arithmetische Mittel des Kurses neu bestimmt wird. Diese Zeitpunkte bezeichnen wir im Folgenden als Mittelungs- oder Beobachtungszeitpunkte. In der Praxis können dies z.B. die Schlusskurse eines Handelstages sein. Für einen Zeitpunkt  $t$  bestimmt sich der Durchschnitt nun

als

$$A(t) = \frac{1}{J_t} \sum_{i=1}^{J_t} S(t_i), \quad (9.33)$$

mit  $J_t$  als dem größten Index, der  $t_{J_t} < t$  erfüllt. Aufgrund der stufenweisen Entwicklung von  $A(t)$  reicht es aus, den Durchschnitt an den Mittelungszeitpunkten zu aktualisieren.

$$\begin{aligned} A(t_l) &= \frac{1}{l} \sum_{i=1}^l S(t_i) \\ &= A(t_{l-1}) + \frac{1}{l} (S(t_l) - A(t_{l-1})). \end{aligned} \quad (9.34)$$

Nach dem Vorgehen von Barraquand und Pudet [BP96] erhalten wir im europäischen Fall eine zu Black-Scholes verwandte PDE

$$V_t + \frac{\sigma^2}{2} S^2 V_{SS} + (r - \delta) S V_S + \frac{S - A}{t} V_A - r V = 0 \quad (9.35)$$

für den Preis  $V(S, A, t)$  einer asiatischen Option. Dieser hängt nun auch vom Durchschnitt  $A$  ab. Für eine Option amerikanischen Typs ergibt sich hier die entsprechende Ungleichung.

Für einen Floating Strike Put hat die Auszahlungsfunktion z.B. die Form:

$$\phi(S, A, t) = \max[A - S, 0]. \quad (9.36)$$

Auch hier sind weitere Varianten denkbar. Wir wollen an dieser Stelle nicht weiter auf die verschiedenen Payoffs eingehen und verweisen daher auf die in diesem Abschnitt zitierten Artikel und Bücher.

### 9.2.2. Berechnung diskret gemittelter asiatischer Optionen

Bevor wir uns der Berechnung zuwenden, soll auch hier eine Zeitumkehr der Form  $\tau = T - t$  durchgeführt werden. Aus der Gleichung (9.35) wird die PDE

$$V_\tau - \frac{\sigma^2}{2} S^2 V_{SS} - (r - \delta) S V_S - \frac{S - A}{T - \tau} V_A + r V = 0. \quad (9.37)$$

Ebenso muss die Mittelungsvorschrift (9.34) angepasst werden. Für  $\tau_l := T - t_l$  gilt dann:

$$A(\tau_{l-1}) = A(\tau_l) + \frac{1}{l-1} (A(\tau_l) - S(\tau_l)). \quad (9.38)$$

Wie schon oben angemerkt, ist der Durchschnitt zwischen je zwei Mittelungszeitpunkten konstant. Das heißt aber, dass der Konvektionsterm der PDE (9.37) in  $A$ -Richtung identisch zu Null zwischen diesen Zeitpunkten ist. Im europäischen Fall vereinfacht sich die Gleichung

zur schon bekannten Black-Scholes Gleichung. Die Ungleichung für eine asiatische Option amerikanischen Typs vereinfacht sich dementsprechend auch auf das eindimensionale Komplementaritätsproblem. Mit dieser Überlegung haben Wilmott et al [WHD95] bereits einen Algorithmus für die Bewertung entwickelt.

**Algorithmus 9.2.1. Bewertung asiatischer Optionen [WHD95, S. 234]**

- i) Man finde den nächsten Mittelungszeitpunkt  $\tau_l$ .
- ii) Bewertung der Option für das Zeitintervall  $]\tau_{l+1}, \tau_l[$  mit einem Verfahren für die eindimensionale Gleichung. Dieser Schritt muss für eine Menge möglicher Durchschnittswerte  $A$  vollzogen werden.
- iii) Anwenden geeigneter Sprungbedingungen für den Zeitpunkt  $\tau_l$ , um den Wert der Option nach dem Mittelungszeitpunkt zu erhalten.
- iv) Beginn eines neuen Durchlaufs für  $l := l - 1$ .

Dieser Algorithmus gibt nur ein grobes Vorgehen wieder. Insbesondere der Schritt iii) erfordert einigen technischen Aufwand. Genauso wie wir uns hier nicht näher über die Wahl der Randbedingungen äußern, so sollen die für den Schritt iii) benötigten Sprungbedingungen ebenfalls nur angerissen werden. Wir halten uns dabei an das Vorgehen von Zvan et al. [ZFV99], einem Artikel der auch genauere Auskunft über die Wahl der Randbedingungen gibt.

Für einen Mittelungszeitpunkt  $\tau_l$  definieren wir

$$A^+ := A(\tau_{l-1}), A^- := A(\tau_l); \quad \tau^+ := \tau_l + \epsilon, \tau^- := \tau_l - \epsilon. \tag{9.39}$$

Die Indizierung der  $A$  erfolgt umgekehrt zur Indizierung der Zeitpunkte, denn wir bewegen uns im Gegensatz zum Index  $l$  der Echtzeit  $t$  rückwärts in der Zeit  $\tau$ . An der Stelle  $\tau_l$  muss nun eine Sprungbedingung, vergleichbar mit der Sprungbedingung für Dividenden aus Kapitel 1.5 der Form

$$V(S, A^+, \tau^+) = V(S, A^-, \tau^-) \tag{9.40}$$

implementiert werden. Der Wert für  $A^+$  ergibt sich aus der Gleichung (9.38) als

$$A^+ = A^- + \frac{1}{l-1}(A^- - S). \tag{9.41}$$

Wegen einer Diskretisierung mit einer fixen Menge von Durchschnittswerten  $A$  muss für die Anwendung der Sprungbedingung in der Regel zwischen den Optionswerten interpoliert werden. Auch hier verweisen wir für die Details auf die Arbeit von Zvan, Forsyth und Vetzal. Wir wollen jedoch noch festhalten, dass die Richtung der Konvektion in  $A$ -Richtung durch

$$-\frac{S-A}{T-\tau} \begin{cases} > 0 & \text{falls } S < A \\ = 0 & \text{falls } S = A \\ < 0 & \text{falls } S > A \end{cases} \tag{9.42}$$

bestimmt wird. Die Information wird somit in einem Beobachtungszeitpunkt von der Diagonalen  $S = A$  entlang der Geraden  $S$  konstant transportiert. Dies ergibt ein schematisches Verhalten, wie es in Abbildung 9.2 dargestellt wird.

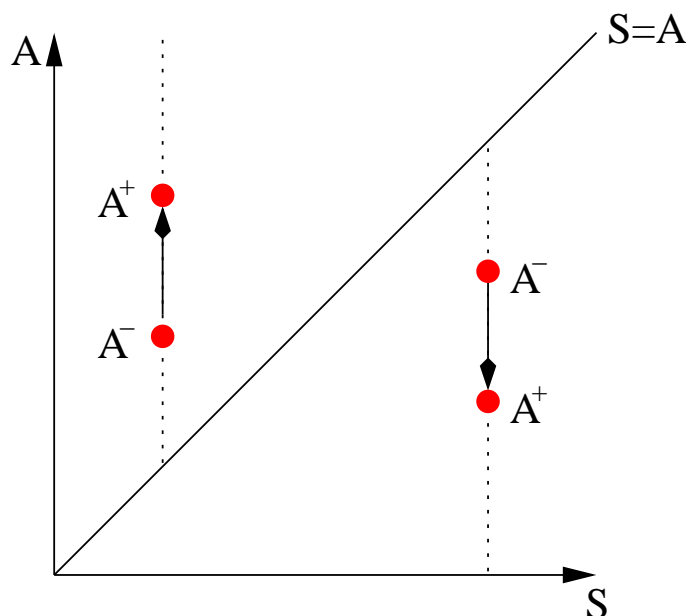


Abbildung 9.2.: Schematisch: Informationsfluss für eine diskret gemittelte asiatische Option. Für einen Mittelungszeitpunkt ist die  $(S, A)$ -Ebene wiedergegeben. Durch die Anwendung der Sprungbedingungen fließt die Information von der Diagonalen  $S = A$  nach außen. Die Abbildung ist ähnlich zu [ZFV99, Abb. 1, S. 48], jedoch unterscheidet sich die Indizierung durch die verschiedenen Zeitdefinitionen.

Im Rahmen dieser Arbeit ist jedoch der Schritt ii) des Algorithmus 9.2.1 interessanter. Hier ist insbesondere die

**Bemerkung 9.2.2. Zerlegbarkeit des Problems [ZFV99, S. 46]**

*“Conceptually, this problem [Bewertung einer asiatischen Option] can be considered to be a collection of one-dimensional PDEs, embedded in a higher- (in this case two- [...]) dimensional space. Communication between these independent one-dimensional problems only occurs at observation times. Clearly, this approach is ideally suited to an implementation on parallel architectures, ...”<sup>1</sup>*

entscheidend. Zum einen wird hier noch einmal die Tatsache hervorgehoben, die den Schritt ii) ausmacht. Für eine Diskretisierung der  $A$ -Achse werden, für verschiedene Wer-

<sup>1</sup>Die Einfügung bzw. Auslassung ist zur besseren Verständlichkeit im Kontext vorgenommen worden.



te von  $A$ , **unabhängige** eindimensionale Optionswerte berechnet. Auf der anderen Seite wird angesprochen, dass sich diese Aufgabenstellung, aufgrund der Unabhängigkeit der Teilprobleme, als ideal für den Einsatz eines Parallelrechners darstellt. Unter diesen Gesichtspunkten können wir uns der Übertragbarkeit der Theorie zuwenden.

### 9.2.3. Übertragbarkeit der Theorie

Dieser Abschnitt umfasst wieder die zwei Punkte: Transformation der Gleichung und Steuerung der Zeitschrittweite. Zum einen sind nun, bei einem Gitter auf der  $A$ -Achse mit z.B. 100 Knoten, 100 voneinander unabhängige eindimensionale Black-Scholes (Un)gleichungen zu lösen. Diese unterscheiden sich von den zuvor behandelten Vanilla Optionen nur durch die Rand- und Auszahlungsbedingungen. Damit gilt:

#### **Korollar 9.2.3. Anwendbarkeit der Transformationen für asiatische Optionen**

- i) Die Anwendbarkeit der dispersionsunterdrückenden Transformationen für diskret gemittelte asiatische Optionen ist möglich. Für jede Gleichung, der im Schritt ii) des Algorithmus 9.2.1 entkoppelten Teilprobleme, wird die PDE mittels einer der Transformationen auf eine Bewertungsgleichungen mit Form der Gleichung (1.10) bzw. Gleichung (1.11) gebracht.*
- ii) Der Algorithmus von Dempster, Hutton und Richards ist zur Bewertung dieser Teilprobleme geeignet.*

#### **Beweis:**

- i) Folgt aus der Reduktion des Problems zwischen den Beobachtungszeitpunkten gemäß Algorithmus 9.2.1 auf eindimensionale Black-Scholes PDEs (evtl. mit nichtkonstanten Parametern) und die Entkopplung dieser Gleichungen nach Bemerkung 9.2.2.
- ii) Der DHR-Algorithmus ist bereits erfolgreich von Dempster, Hutton und Richards auf diese Problemklasse angewandt worden ([DHR98, Kapitel 4.3]). Die Modifikation des Algorithmus durch das inverse Update (Algorithmus 4.2.5) schränkt diese Aussage nicht ein. □

Auch für die Steuerung der Zeitschrittweite lässt sich eine ähnliche Aussage treffen:

#### **Korollar 9.2.4. Steuerung der Zeitschrittweite für asiatische Optionen**

*Eine Anwendung der Zeitschrittweitensteuerung ist auch für diskret gemittelte asiatische Optionen möglich.*

#### **Beweis:**

Auch hier folgt die Aussage sofort aus dem Vorgehen des Algorithmus 9.2.1 und der Bemerkung 9.2.2. Zwischen zwei Beobachtungszeitpunkten liegt wieder ein Problem in der bereits behandelten Form vor. Für jedes  $A$  muss eine eindimensionale amerikanische Vanilla Option mit abgewandeltem Payoff sowie geänderten Anfangswerten berechnet

werden. Zuvor haben wir gezeigt, dass eine Zeitschrittweitensteuerung für diese Klasse von Problemen möglich ist.  $\square$

Eine numerische Umsetzung hat für die Zeitschrittweitensteuerung asiatischer Optionen genau wie für die Basket Optionen nicht stattgefunden. Wir schließen dieses Kapitel mit der nachfolgenden Überlegung ab.

### **Bemerkung 9.2.5.**

- i) Es ist **nicht** möglich, die Aussage des Korollars 8.1.3 über die Homogenität des Optionswertes im Strikepreis für asiatische Optionen zu retten. Nach der ersten Anwendung der Sprungbedingungen haben die aktuellen ‚Startwerte‘, d.h. die Anfangswerte für die unabhängige Bewertung im Schritt ii) des Algorithmus 9.2.1, für verschiedene  $A$  nicht mehr die Struktur eines homogenen Payoffs. Diesen benötigt Merton aber für den Beweis von Lemma 8.1.2.*
- ii) Für die Parallelisierung des Algorithmus 9.2.1 lässt sich vermuten, dass der Einsatz einer Zeitschrittweitensteuerung besonders sinnvoll ist. Hier gilt die Faustregel, dass sich der Effekt der Parallelisierung für ein performantes serielles Programm besonders stark auswirkt.*

# 10. Zusammenfassung und Ausblick

In dieser Arbeit haben wir uns mit verschiedenen Aspekten der numerischen Optionsbewertung befasst. Dies sind im Wesentlichen das Auftreten von numerischer Dispersion auf der eine Seite und das Verhalten des Optionswertes im Zeitablauf auf der anderen Seite.

Im Hinblick auf das Dissipations-Dispersions-Verhalten der Black-Scholes Gleichung haben wir aus der Literatur bekannte Transformationen analysiert. Diese liefern zum Teil ohne weitergehende numerische Hilfsmittel dispersionsfreie Lösungen der PDE. Für den in der Literatur bisher nicht abgedeckten Fall einer Modellierung mit lokaler Volatilität haben wir im Eindimensionalen eine neue Transformation hergeleitet, die auch die geeignete Eigenschaft, das Fehlen eines Konvektionsterms, besitzt. Im Vorfeld wurde die Ähnlichkeit der Bedingungen für eine TVD-Diskretisierung der Gleichung und eine Übertragung des Problems in ein Lineares Programm dargestellt. Dabei stand insbesondere die Abhängigkeit der Maschenweiten von der Péclet-Zahl im Vordergrund. Mit der Transformation aus [BP96, Smi00] und der von uns neu entwickelten Transformation ist es möglich geworden, die Black-Scholes PDE ohne numerische Dispersion zu lösen. Die so berechneten Lösungen sind auch in den Ableitungen frei von Oszillationen und ermöglichen dadurch ein effektives Hedging. Gleichzeitig vermeiden wir den Einsatz von nichtlinearen Flussbegrenzern, einem numerischen Instrument, welches mit dem Algorithmus von Dempster, Hutton und Richards für die Darstellung des Problems als Lineares Programm nicht vereinbar ist. Allerdings sind wir durch die Transformationen gezwungen, den DHR-Algorithmus zu modifizieren.

Aufgrund der Verwandtschaft der Black-Scholes PDE zur Wärmeleitungsgleichung haben wir die Verwendung einer Steuerung der Zeitschrittweite motiviert. Um diese implementieren zu können, mussten wir uns mit den Eigenschaften des Problems auseinandersetzen. Dabei spielte insbesondere die eingeschränkte Differenzierbarkeit der Wertfunktion einer amerikanischen Option eine Rolle. Eine Behandlung der Auszahlung zur Fälligkeit haben wir, wie in [Ran84, PFV03] vorgeschlagen, analog zum europäischen Gegenstück über die Glättung durch L-stabile Verfahren vorgenommen. Wir zeigen, dass die Berechnung einer amerikanischen Option konsequenter Weise in jedem Zeitschritt dieses ‚Rannacher smoothing‘ erfordert.

Für die Steuerung der Zeitschrittweite stellen wir, neben einer bereits von Forsyth und Vetzal verwendeten Methode, zwei weitere Verfahren vor. Diese führen die Steuerung jeweils über den Vergleich mit einer alternativen Lösung durch. Wir haben dann die Eigen-

schaften dieser drei Verfahren im Zusammenhang mit dem DHR-Algorithmus analysiert. Der Vergleich der Diskretisierungsordnungen hat sich durch die Verwendung des Crank-Nicolson-Verfahrens als nur eingeschränkt tauglich erwiesen. Nachdem wir uns auf diese Weisen einen theoretischen Überblick verschafft haben, wurde die praktische Umsetzung der Verfahren angegangen. Dabei wurde insbesondere der Tatsache Rechnung getragen, dass es sich auch um ein Gleichungssystem in der Größe  $N$  der Raumdiskretisierung handelt. Der entstehende Aufwand konnte durch eine Anpassung der Steuerungsvorschrift erheblich reduzieren werden. Genau wie bereits für die Dispersions-Unterdrückung müssen die Algorithmen angepasst werden, um die Transformationen verwenden zu können.

Mit den von uns entwickelten Algorithmen lassen sich Ergebnisse berechnen, die qualitativ und quantitativ mit den bisher verwendeten Methoden konkurrieren können. Insbesondere wurde dabei Wert auf die Berechnung des Optionswertes zum Bewertungszeitpunkt und die Ermittlung des Verlaufes der numerisch ermittelten Ausübungsgrenze gelegt. Aus zuvor hergeleiteten Eigenschaften der Option konnten die für den Vergleich benötigten Kriterien abgeleitet werden. Insbesondere haben wir eine gewisse Skalierbarkeit des Optionswertes und die Invarianz der Ausübungsgrenze gegenüber Variationen des Strikes  $K$  nach der Transformation hergeleitet. Zu erwähnen bleibt, dass sich die neuen Lösungen in gleicher Qualität mit einem Speed-up der Größenordnung drei bis sechs gegenüber dem bisherigen Verfahren mit konstanter Schrittweite berechnen lassen. Gleichzeitig stellt sich heraus, dass der eigentliche Aufwand der Algorithmen nicht mehr in den Zeitschritten, sondern in der Ermittlung der optimalen Ausübungsgrenze steckt.

Abschließend haben wir die Ergebnisse auf ausgewählte exotische Papiere übertragen. Die eindimensionalen Resultate bleiben in diesen Fällen theoretisch erhalten und wurden teilweise auch numerisch verifiziert.

Sowohl vom theoretischen als auch praktischen Standpunkt bleiben einige Fragen offen. Insbesondere konnte für den Fall lokaler Volatilität die Frage nach einer die numerische Dispersion unterdrückenden Transformation für die mehrdimensionale Black-Scholes Gleichung nicht geklärt werden. Die Einschränkung der Klasse L-stabiler Verfahren auf den impliziten Euler ist nicht zwingend. Aussagen über den benötigten Aufwand bei der Verwendung anderer L-stabiler Verfahren stehen noch aus.

Für die Praxis entscheidend ist die Frage, ob ein zusätzlicher Speed-up durch bessere näherungsweise Kenntnis der optimalen Ausübungsgrenze möglich ist. Unabhängig davon ist die Umsetzung der Theorie und der Algorithmen für diskret gemittelte asiatische Optionen zu leisten. In diesem Punkt wird neben einer seriellen auch an die parallele Implementierung gedacht. Die Untersuchung von Alternativen zum DHR-Algorithmus im Zusammenhang mit den Transformationen und der Steuerung der Zeitschrittweite wäre ein interessantes Projekt.

Als größte Herausforderung ist die Entwicklung einer Theorie für den globalen Diskretisierungsfehler einer partiellen Differentialgleichung zu sehen.

# A. Berechnungsaufwand

Der Aufwand für die wesentlichen Elemente des Algorithmus von Dempster, Hutton und Richards wird in diesem Kapitel hergeleitet. Gleichzeitig wird damit ein Beweis für den Satz 5.3.2 geliefert. Wie ebenda wird angenommen, dass die Länge der Iterationsvektoren  $N$  ist. Im gesamten Kapitel beziehen sich Gleichungsnummern mit dem Zusatz ‚a‘ auf die Konvektions-Verschiebung und mit dem Zusatz ‚b‘ auf die neue Transformation.

## Matrizen besetzen

Für die Iteration werden folgende Matrizen benötigt:

i) Für das implizite Euler-Verfahren:

$$A := I - \frac{\Theta}{\Delta\tau} \frac{\sigma^2}{2} \hat{D}, \quad B := I. \quad (\text{A.1a})$$

$$A := I - \frac{\Theta}{\Delta\tau} \frac{\sigma^2}{2} (\hat{D} - I), \quad B := I. \quad (\text{A.1b})$$

ii) Für das Crank-Nicolson-Verfahren:

$$A := I - \frac{\Theta}{\Delta\tau} \frac{\sigma^2}{2} \hat{D}, \quad B := I + \frac{1 - \Theta}{\Delta\tau} \frac{\sigma^2}{2} \hat{D}. \quad (\text{A.2a})$$

$$A := I - \frac{\Theta}{\Delta\tau} \frac{\sigma^2}{2} (\hat{D} - I), \quad B := I + \frac{1 - \Theta}{\Delta\tau} \frac{\sigma^2}{2} (\hat{D} - I). \quad (\text{A.2b})$$

Hierbei ist  $\hat{D}$  die Matrix, die aus der Diskretisierung des Diffusionsoperators hervorgeht und  $I$  die  $N \times N$ -Einheitsmatrix.

Definiert man nun die Matrix  $\hat{A} := \hat{D}$  bzw.  $\hat{A} := \hat{D} - I$ , so müssen zum einen die Vorfaktoren  $\frac{\Theta}{\Delta\tau} \frac{\sigma^2}{2}$  und  $\frac{1 - \Theta}{\Delta\tau} \frac{\sigma^2}{2}$  berechnet werden. Hierfür entsteht bei konstantem  $\sigma$  ein Aufwand von 4 Multiplikationen für den impliziten Euler bzw. 5 Multiplikationen für das Crank-Nicolson-Verfahren. Für  $\sigma$  als Funktion von  $x$  sind dies  $2N + 2$  bzw.  $3N + 3$  Multiplikationen. Diese müssen dann mit den entsprechenden Einträgen in  $\hat{A}$  multipliziert werden. Dabei sind bei Tridiagonalgestalt  $3N$  Einträge pro Matrix betroffen. Dies macht  $3N$  Multiplikationen für

das implizite Euler-Verfahren und  $6N$  Multiplikationen für das Crank-Nicolson-Verfahren. Die Addition zur Einheitsmatrix geht jeweils mit  $N$  Additionen ein.

Es bleibt also im Wesentlichen noch der Aufwand, um die Einträge in  $\hat{A}$  zu bestimmen. Bei einem beliebigen Gitter ergeben sich zur Berechnung von  $\hat{A} = \hat{D} - I$

je Zeile für: $l = x_i - x_{i-1},$	$h = x_{i+1} - x_i$		2 Additionen
	$\hat{a}_{i,i-1} = \frac{2}{l(l+h)}$	2 Multiplikationen	1 Addition
	$\hat{a}_{i,i} = -\frac{2}{lh} - 1$	2 Multiplikationen	1 Addition
	$\hat{a}_{i,i+1} = \frac{2}{h(l+h)}$	2 Multiplikationen	1 Addition
		6 Multiplikationen	5 Additionen

Insgesamt werden so  $6N$  Multiplikationen und  $5N$  Additionen benötigt. Falls  $\hat{A}$  die Form  $\hat{D}$  hat, entfällt für den Diagonaleintrag eine Addition. In der Summe verbleiben dann  $6N$  Multiplikationen und  $4N$  Additionen. Bei Verwendung eines äquidistanten Gitters reduziert sich der Aufwand zu der Berechnung von  $\frac{1}{\Delta x^2}$ , d.h. 2 Multiplikationen für den Vorfaktor sowie  $3N$  Multiplikationen, um die Einträge in der Matrix zu bestimmen. Für die neue Transformation kommen hier weitere  $N$  Additionen hinzu.

Zusammenfassend ergeben sich

a) für das implizite Euler-Verfahren

i) mit einem beliebigen Gitter

$$\begin{aligned}
 & 5N + 2 \text{ Multiplikationen} + N \text{ Additionen} \\
 & + 6N \text{ Multiplikationen} + 4N \text{ Additionen} \\
 = & 11N + 2 \text{ Multiplikationen} + 5N \text{ Additionen,}
 \end{aligned} \tag{A.3a}$$

$$\begin{aligned}
 & 5N + 2 \text{ Multiplikationen} + N \text{ Additionen} \\
 & + 6N \text{ Multiplikationen} + 5N \text{ Additionen} \\
 = & 11N + 2 \text{ Multiplikationen} + 6N \text{ Additionen.}
 \end{aligned} \tag{A.3b}$$

ii) mit äquidistantem Gitter

$$\begin{aligned}
 & 5N + 2 \text{ Multiplikationen} + N \text{ Additionen} \\
 & + 3N + 2 \text{ Multiplikationen} \\
 = & 8N + 4 \text{ Multiplikationen} + N \text{ Additionen,}
 \end{aligned} \tag{A.4a}$$

$$\begin{aligned}
 & 5N + 2 \text{ Multiplikationen} + N \text{ Additionen} \\
 & + 3N + 2 \text{ Multiplikationen} + N \text{ Additionen} \\
 = & 8N + 4 \text{ Multiplikationen} + 2N \text{ Additionen.}
 \end{aligned} \tag{A.4b}$$

b) für das Crank-Nicolson-Verfahren

i) mit einem beliebigen Gitter

$$\begin{aligned} & 8N + 2 \text{ Multiplikationen} + N \text{ Additionen} \\ & + 6N \text{ Multiplikationen} + 4N \text{ Additionen} \\ = & 14N + 2 \text{ Multiplikationen} + 5N \text{ Additionen,} \end{aligned} \tag{A.5a}$$

$$\begin{aligned} & 8N + 2 \text{ Multiplikationen} + N \text{ Additionen} \\ & + 6N \text{ Multiplikationen} + 5N \text{ Additionen} \\ = & 14N + 2 \text{ Multiplikationen} + 6N \text{ Additionen.} \end{aligned} \tag{A.5b}$$

ii) mit äquidistantem Gitter

$$\begin{aligned} & 8N + 2 \text{ Multiplikationen} + N \text{ Additionen} \\ & + 3N + 2 \text{ Multiplikationen} \\ = & 11N + 4 \text{ Multiplikationen} + N \text{ Additionen,} \end{aligned} \tag{A.6a}$$

$$\begin{aligned} & 8N + 2 \text{ Multiplikationen} + N \text{ Additionen} \\ & + 3N + 2 \text{ Multiplikationen} + N \text{ Additionen} \\ = & 11N + 4 \text{ Multiplikationen} + 2N \text{ Additionen.} \end{aligned} \tag{A.6b}$$

## UL-Zerlegung

Die sukzessive Abfolge des Spaltentausches im Algorithmus 4.1.1, dem Algorithmus von Dempster, Hutton und Richards, kann sinnvoller als eine UL-Zerlegung des Problems analysiert werden. Es ergibt sich dann in den einzelnen Schritten folgender Aufwand:

$$\begin{aligned} L_{n,n} &= a_{n,n} \\ U_{n-1,n} &= \frac{a_{n-1,n}}{L_{n,n}} \end{aligned} \quad 1 \text{ Multiplikation}$$

*i*-Schleife  $i = n - 1, \dots, nb + 1$

$$\begin{aligned} L_{i+1,i} &= a_{i+1,i} \\ L_{i,i} &= a_{i,i} - L_{i+1,i} U_{i,i+1} & 1 \text{ Multiplikation} & 1 \text{ Addition} \\ U_{i-1,i} &= \frac{a_{i-1,i}}{L_{i,i}} & 1 \text{ Multiplikation} \end{aligned}$$

*i*-Schleife  $i = nb, \dots, 1$

$$\begin{aligned} L_{i+1,i} &= 0 \\ U_{i-1,i} &= 0 \\ L_{i,i} &= -1 \end{aligned}$$

Die volle UL-Zerlegung entspricht dem Fall  $nb = 0$ . Wir unterstellen immer diesen Fall und überschätzen damit den Aufwand. Dabei gehen wir von notwendigen  $2N - 1$  Multiplikationen und  $N - 1$  Additionen aus.

## Lösen des Gleichungssystems $(UL)z^{(m)} = b^{(m)}$

Vor der Lösung des eigentlichen Systems muss der Vektor der rechten Seite  $b^{(m)}$  ermittelt werden. Dieser ergibt sich aus der Gleichung

$$b^{(m)} = B u^{(m)} - A \phi^{(m)}.$$

Um diese auszuwerten, ist für das implizite Euler-Verfahren eine Matrix-Vektor-Multiplikation sowie die Addition des Ergebnisvektors zu  $u^{(m)}$  durchzuführen. Dafür werden je Zeile 3 Multiplikationen und 3 Additionen benötigt. Für das Crank-Nicolson-Verfahren sind dies zwei Matrix-Vektor-Multiplikationen und die Addition beider Ergebnisse. Je Zeile also 6 Multiplikationen und 5 Additionen. Es ergeben sich damit

- a) für das implizite Euler-Verfahren  $3N$  Multiplikationen und  $3N$  Additionen,
- b) für das Crank-Nicolson-Verfahren  $6N$  Multiplikationen und  $5N$  Additionen,

um die rechte Seite des Gleichungssystems zu bestimmen. Die Auflösung des Systems  $U y^{(m)} = b^{(m)}$  mittels

$$y_n^{(m)} = b_n^{(m)}$$

$i$ -Schleife  $i = n - 1, \dots, 1$

$$y_i^{(m)} = b_i^{(m)} - U_{i,i+1} y_{i+1}^{(m)} \quad 1 \text{ Multiplikation} \quad 1 \text{ Addition}$$

sowie des Systems  $L z^{(m)} = y^{(m)}$  über

$$z_1^{(m)} = \frac{y_1^{(m)}}{L_{1,1}} \quad 1 \text{ Multiplikation}$$

$i$ -Schleife  $i = 2, \dots, n$

$$z_i^{(m)} = \frac{y_i^{(m)} - L_{i,i-1} z_{i-1}^{(m)}}{L_{i,i}} \quad 2 \text{ Multiplikationen} \quad 1 \text{ Addition}$$

ist unabhängig vom verwendeten Verfahren. Der Aufwand beträgt hier für beide Systeme zusammen  $3N - 2$  Multiplikationen und  $2N - 2$  Additionen.

Insgesamt benötigt die Lösung des Gleichungssystems

- a) für das implizite Euler-Verfahren  $6N - 2$  Multiplikationen und  $5N - 2$  Additionen.
- b) für das Crank-Nicolson-Verfahren  $9N - 2$  Multiplikationen und  $7N - 2$  Additionen.



## B. Bemerkungen zur Implementierung

Wir machen in diesem Teil des Anhangs einige Angaben zur durchgeführten Implementierung. Der allgemeine Teil bezieht sich dabei auf die Umsetzung aller Algorithmen. Nur für die Implementierung der Berechnung von Basket Optionen sind wir davon abgewichen.

### Allgemeine Implementierung

Die jeweiligen Verfahren sind mit MATLAB<sup>®</sup> programmiert worden. Dabei wurde die Version 6.5 im Release 13 verwendet. Eine Einführung zu den verwendeten Befehlen und der MATLAB<sup>®</sup> Umgebung gibt es in der Online-Hilfe oder im Benutzerhandbuch [Mat02b]. MATLAB<sup>®</sup> ermöglicht die einfache Programmierung numerischer Algorithmen ohne die explizite Zuweisung von Speicher. Gleichzeitig besteht die Möglichkeit zur Grafikausgabe innerhalb des Programms. Zu dem Basispaket des Programms sind weitere Toolboxen verfügbar. Insbesondere gibt es eine ‚Optimization Toolbox‘, die einen fertigen Simplex-Löser anbietet. Diese Toolbox ist jedoch **nicht** verwendet worden, um die Abhängigkeit der Methoden von kommerzieller Software auszuschließen.

Zu beachten ist weiterhin der selektive Umgang mit den von MATLAB<sup>®</sup> mitgelieferten Routinen. So haben wir durchaus auf die ‚sparse‘-Anweisung zurückgegriffen, die eine einfachere Verwendung dünnbesetzter Matrizen ermöglicht. Allerdings ist auf den Gebrauch der mitgelieferten Routinen für die Lösung linearer Gleichungssysteme verzichtet worden. Zu den Gründen: Ein Vergleich zeigt, dass die eingebauten Routinen in etwa um den Faktor drei schneller sind als der über das Interpreterfenster oder so genannte ‚m-files‘ ausgeführte Code. Ein wesentlicher Augenmerk in der Arbeit liegt aber auf dem Ausgleich des Aufwandes für die Lösung der Gleichungssysteme und dem Aufwand für die Neubesetzung und Zerlegung der Matrizen bei Schrittweitenwechseln. Um bzgl. der Laufzeit gleiche Verhältnisse zu schaffen, ist hier bewusst auf die fertigen Routinen verzichtet worden. Unter diese fertigen Routinen würde die ‚Optimization Toolbox‘ ebenfalls fallen.

Die Berechnungen sind jeweils auf einem PC mit Pentium III Prozessor mit 1 Gigaherz Taktfrequenz und 256 MB Speicher ausgeführt worden. MATLAB<sup>®</sup> liefert dabei generell eine Maschinengenauigkeit von  $10^{-16}$ .

## **Basket Optionen**

Für die Berechnung der Optionswerte in Kapitel 9.1.3 wurde, wie dort schon angemerkt, auf die CPLEX<sup>®</sup> callable library [ILO99a] zurückgegriffen. Die Bibliothek wurde in ein C-Programm integriert und mit diesem wurde dann der Optionswert berechnet. Zu den verwendeten Befehlen verweisen wir auf die Erläuterungen im Benutzerhandbuch [ILO99b]. Auch wenn wir für dieses Beispiel keine Laufzeiten angeben, so sei angemerkt, dass die Berechnungen auf einer SUN<sup>®</sup> Workstation durchgeführt wurden.

## C. Vergleichszahlen

Dieser Teil des Anhangs gibt einige absolute Vergleichszahlen wieder. Dabei greifen wir auf das Beispiel 8.3.2 zurück. Nach der Berechnung mit einer leicht abgewandelten Diskretisierung werden zusätzlich zu den Laufzeiten (vgl. auch Tabellen 8.2 und 8.3) auch die Optionspreise aufgeführt. Wir geben dabei als Indikator **nur** den Wert der Option am Geld wieder, d.h. den Wert, der sich für  $V(K, 0)$  oder  $S_0 = K$  ergibt.

Zur Erinnerung nochmals die Parameter des Beispiels:

*Man berechne den Preis eines amerikanischen Vanilla Puts mit  $K = 1.0$ ,  $T = 1.0$ ,  $r = 0.1$ ,  $\delta = 0$  und  $\sigma = 0.3$ .*

Hier wird die Gleichung in x-Koordinaten im Intervall  $[-2, 2]$  diskretisiert. Es werden drei Schrittstufen je Zehnerpotenz zugelassen, d.h.  $\alpha = 3$ . Dabei wird mit einer maximal zulässigen Zeitschrittweite  $\Delta\tau_{max} = 10^{-1.5}$  gearbeitet. Wir transformieren das Koordinatensystem mit der Konvektions-Verschiebung der auf Barraquand und Pudet [BP96] bzw. Smith [Smi00] zurückgehenden Transformation.

Die ermittelten Werte finden sich in Tabellen auf der nächsten Seite.

$\Delta\tau$	$\Delta x = 10^{-2}$		$\Delta x = 10^{-2}$	
	$V_{aequidist.}(K, 0)$	sec	$V_{aequidist.}(K, 0)$	sec
$10^{-2}$	0.0832418	5.22	0.0832561	52.64
$5 \cdot 10^{-3}$	0.0832997	6.97	0.0833142	71.26
$10^{-3}$	0.0833485	29.38	0.0833633	222.29
$5 \cdot 10^{-4}$	0.0833589	58.15	0.0833699	402.65

Tabelle C.1.: Optionswerte und Laufzeiten in CPU-Sekunden bei einer Berechnung **ohne** Schrittweitensteuerung, d.h. mit äquidistanten Zeitschritten.

$\Delta x$	tol	$V_{halb}(K, 0)$	sec	$V_{CN}(K, 0)$	sec	$V_{impl}(K, 0)$	sec
$10^{-2}$	$10^{-2}$	0.0830351	3.47	0.0830371	2.38	0.0830496	2.00
	$10^{-3}$	0.0830591	3.83	0.0830680	2.61	0.0832856	5.62
	$10^{-4}$	0.0831343	5.69	0.0831395	3.92	0.0833538	45.76
	$10^{-5}$	0.0832562	13.25	0.0832597	7.59	0.0833607	464.47
$10^{-3}$	$10^{-2}$	0.0830454	82.16	0.0830523	55.01	0.0830661	39.53
	$10^{-3}$	0.0830738	88.69	Error	-	0.0833007	66.39
	$10^{-4}$	0.0831489	105.57	0.0831540	66.18	0.0833690	354.42
	$10^{-5}$	0.0832713	180.25	0.0832747	90.69	0.0833759	3077.23

Tabelle C.2.: Optionswerte und Laufzeiten in CPU-Sekunden bei einer Berechnung **mit** einer Zeitschrittweitensteuerung. Das jeweils verwendete Verfahren ist dem Index des Optionswertes zu entnehmen. Wir übernehmen dabei die auf Seite 109 eingeführte Notation.

# Literaturverzeichnis

- [ABR98] L. B. G. Andersen and R. Brotherton-Ratcliffe. The equity option volatility smile: an implicit finite-difference approach. *The Journal of Computational Finance*, 1(2):5–37, 1998.
- [BG69] R. H. Bartels and G. H. Golub. The simplex method of linear programming using LU decomposition. *Communications of the ACM*, 12(5):266 – 268, 1969.
- [BP96] J. Barraquand and T. Pudet. Pricing of American Path-Dependent Contingent Claims. *Mathematical Finance*, 6(1):17–51, 1996.
- [BS73] F. Black and M. Scholes. The Pricing of Options and Corporate Liabilities. *Journal of Political Economy*, 81(1):637–654, 1973.
- [BS78] M. J. Brennan and E. S. Schwartz. Finite difference methods and jump processes arising in the pricing of contingent claims: A synthesis. *Journal of Financial & Quantitative Analysis*, 13(1):461–474, 1978.
- [CLV02] T. F. Coleman, Y. Li, and A. Verma. A Newton method for American option pricing. *The Journal of Computational Finance*, 5(3):51–78, 2002.
- [CP99] N. Clarke and K. Parrott. Multigrid for American option pricing with stochastic volatility. *Applied Mathematical Finance*, 6(3):177–195, 1999.
- [CR76] J. C. Cox and S. A. Ross. The Valuation of Options for Alternative Stochastic Processes. *Journal of Financial Economics*, 3(2):145–166, 1976.
- [CRR79] J. C. Cox, S. A. Ross, and M. Rubinstein. Option Pricing: A Simplified Approach. *Journal of Financial Economics*, 7(3):229–263, 1979.
- [DH97] M. A. H. Dempster and J. P. Hutton. Fast numerical valuation of American, exotic and complex options. *Applied Mathematical Finance*, 4(1):1–20, 1997.
- [DH99] M. A. H. Dempster and J. P. Hutton. Pricing American Stock Options by Linear Programming. *Mathematical Finance*, 9(3):229–254, 1999.
- [DHR98] M. A. H. Dempster, J. P. Hutton, and D. G. Richards. LP Valuation of exotic American options exploiting structure. *The Journal of Computational Finance*, 2(1):61–84, 1998.

- [DK94] E. Derman and I. Kani. Riding on a Smile. *RISK*, 7(2):32–39, 1994.
- [DR00] M. A. H. Dempster and D. G. Richards. Pricing American options fitting the smile. *Math. Finance*, 10(2):157–177, 2000. INFORMS Applied Probability Conference (Ulm, 1999).
- [Dup94] B. Dupire. Pricing with a Smile. *RISK*, 7(1):18–20, 1994.
- [EJ87] K. Eriksson and C. Johnson. Error estimates and automatic time step control for nonlinear parabolic problems. I. *SIAM J. Numer. Anal.*, 24(1):12–23, 1987.
- [Feh64] E. Fehlberg. New high-order Runge-Kutta formulas with step size control for systems of first- and second-order differential equations. *Z. Angew. Math. Mech.*, 44:T17–T29, 1964.
- [FV02] P. A. Forsyth and K. R. Vetzal. Quadratic convergence for valuing American options using a penalty method. *SIAM J. Sci. Comput.*, 23(6):2095–2122 (electronic), 2002.
- [GJ84] R. Geske and H. E. Johnson. The American Put Option Valued Analytically. *Journal of Finance*, 39(5):1511–1524, 1984.
- [Gla04] P. Glasserman. *Monte Carlo methods in financial engineering*, volume 53 of *Applications of Mathematics (New York)*. Springer-Verlag, New York, 2004. Stochastic Modelling and Applied Probability.
- [GSB87] I. Gladwell, L. F. Shampine, and R. W. Brankin. Automatic selection of the initial step size for an ODE solver. *J. Comput. Appl. Math.*, 18(2):175–192, 1987.
- [HNW91] E. Hairer, S. P. Nørsett, and G. Wanner. *Solving ordinary differential equations. I*, volume 8 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, 2 edition, 1991. Nonstiff problems.
- [Hul00] J. C. Hull. *Options, Futures and other Derivates*. Prentice-Hall, London, 4 edition, 2000.
- [HW96] E. Hairer and G. Wanner. *Solving ordinary differential equations. II*, volume 14 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, second edition, 1996. Stiff and differential-algebraic problems.
- [ILO99a] ILOG. CPLEX<sup>®</sup> 6.5 callable library, 1999.
- [ILO99b] ILOG. CPLEX<sup>®</sup> 6.5 Reference Manual, 1999.

- 
- [IT04] S. Ikonen and J. Toivanen. Pricing American Options Using LU Decomposition. Report B 4/2004, University of Jyväskylä, Department of Mathematical Information Technology, 2004. Reports of the Department of Mathematical Information Technology Series B. Scientific Computing.
- [IV01] R. Int-Veen. *Darstellung und Bewertung Amerikanischer Optionen mittels Linearer Programmierung*. Diplomarbeit, Universität zu Köln, Mathematisches Institut, 7 2001.
- [IV05] R. Int-Veen. Avoiding Numerical Dispersion in Option Valuation. *Comput Visual Sci.*, 2005. to appear.
- [JLL90] P. Jaillet, D. Lamberton, and B. Lapeyre. Variational inequalities and the pricing of American options. *Acta Appl. Math.*, 21(3):263–289, 1990.
- [JNT90] C. Johnson, Y. Y. Nie, and V. Thomée. An a posteriori error estimate and adaptive timestep control for a backward Euler discretization of a parabolic problem. *SIAM J. Numer. Anal.*, 27(2):277–291, 1990.
- [Kwo98] Y. K. Kwok. *Mathematical Models of Financial Derivatives*. Springer-Verlag Singapore, Singapore, 1998.
- [LW60] P. Lax and B. Wendroff. Systems of conservation laws. *Comm. Pure Appl. Math.*, 13:217–237, 1960.
- [Mat02a] MathWorks Inc. MATLAB<sup>®</sup> 6.5, release 13, 2002.
- [Mat02b] MathWorks Inc. *Using MATLAB<sup>®</sup> (Version 6)*, 2002. The Language of Technical Computing.
- [Mer73] R. C. Merton. Theory of rational option pricing. *Bell Journal of Economics and Management Science*, 4(1):141–183, 1973.
- [OFG04] C. W. Oosterlee, J. C. Frisch, and F. J. Gaspar. TVD, WENO and blended BDF discretizations for Asian options. *Comput. Vis. Sci.*, 6(2-3):131–138, 2004.
- [PFV03] D. M. Pooley, P. A. Forsyth, and K. R. Vetzal. Convergence remedies for non-smooth payoffs in option pricing. *The Journal of Computational Finance*, 6(4):25–40, 2003.
- [Ran84] R. Rannacher. Finite Element Solution of Diffusion Problems with Irregular Data. *Numer. Math.*, 43(2):309–327, 1984.
- [SB00] J. Stoer and R. Bulirsch. *Numerische Mathematik. 2*. Springer-Verlag, Berlin, fourth edition, 2000.
- [Sey04] R. Seydel. *Tools for Computational Finance*. Universitext. Springer-Verlag, Berlin, second edition, 2004.

- [Smi00] R. Smith. Optimal and near-optimal advection-diffusion finite-difference schemes. III. Black-Scholes equation. *The Royal Society of London. Proceedings. Series A. Mathematical, Physical and Engineering Sciences*, 456(1997):1019–1028, 2000.
- [Swe84] P. K. Sweby. High resolution schemes using flux limiters for hyperbolic conservation laws. *SIAM J. Numer. Anal.*, 21(5):995–1011, 1984.
- [Tho95] J. W. Thomas. *Numerical Partial Differential Equations: Finite Difference Methods*. Springer-Verlag, New York, 1995.
- [Tho99] J. W. Thomas. *Numerical Partial Differential Equations*, volume 33 of *Texts in Applied Mathematics*. Springer-Verlag, New York, 1999. Conservation Laws and Elliptic Equations.
- [vL74] Bram van Leer. Towards the Ultimate Conservative Difference Scheme. II. Monotonicity and Conservation Combined in a Second-Order Scheme. *J. Comput. Phys.*, 14:361–370, 1974.
- [vM76] P. van Moerbeke. On optimal stopping and free boundary problems. *Arch. Rational Mech. Anal.*, 60(2):101–148, 1975/76.
- [Wes01] P. Wesseling. *Principles of Computational Fluid Dynamics*, volume 29 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, 2001.
- [WHD95] P. Wilmott, S. Howison, and J. Dewynne. *The Mathematics of Financial Derivatives*. Cambridge University Press, Cambridge, 1995. A Student Introduction.
- [ZFV98] R. Zvan, P. A. Forsyth, and K. R. Vetzal. Robust numerical methods for PDE models of asian options. *The Journal of Computational Finance*, 1(3):39–78, 1998.
- [ZFV99] R. Zvan, P. A. Forsyth, and K. R. Vetzal. Discrete Asian barrier options. *The Journal of Computational Finance*, 3(1):41–67, 1999.
- [ZFV01] R. Zvan, P. A. Forsyth, and K. R. Vetzal. A Finite Volume Approach for Contingent Claims valuation. *IMA Journal of Numerical Analysis*, 21(3):703–731, 2001.



# Abbildungsverzeichnis

2.1.	Oszillationen in der Näherungslösung für verschiedene Zeitschrittweiten . . .	24
4.1.	Ablauf eines Lösungsschrittes des DHR-Algorithmus . . . . .	49
4.2.	Schematischer Verlauf der Ausübungsgrenze . . . . .	52
4.3.	Ablauf eines Lösungsschrittes des modifizierten DHR Algorithmus . . . . .	53
5.1.	Verlauf der Ausübungsgrenze . . . . .	56
5.2.	Knick in der Wertfunktion . . . . .	59
5.3.	Knick in der Ableitung nach $x$ der Wertfunktion . . . . .	63
5.4.	Finite-Differenzen-Stern der Diskretisierung schematisch . . . . .	64
5.5.	Crank-Nicolson-Differenzenstern mit Ausübungsgrenze . . . . .	64
5.6.	Schematischer Ablauf des Wechsel des Grenzknotens . . . . .	69
7.1.	Maximum der zeitlichen Änderung des Optionswertes $v_\tau(x, \tau)$ . . . . .	91
7.2.	Schrittweite und Fehlerapproximation im Iterationsablauf für das Einstiegsproblem . . . . .	92
7.3.	Absturz der Schrittweite bei Implizit/Crank-Nicolson Vergleich . . . . .	93
7.4.	Variation von $\alpha$ : Iterationsablauf für das Einstiegsproblem . . . . .	96
7.5.	Verschiedene Schrittweiten und Suchrichtungen beim modifiziertem DHR-Algorithmus . . . . .	98
7.6.	Details für die Steuerung über Implizit/Crank-Nicolson Vergleich . . . . .	100
7.7.	Details für die Steuerung über Implizit/Implizit-Halbe Vergleich . . . . .	101
8.1.	Konsistenzprüfung für den Optionswert . . . . .	107
8.2.	Vergleich der Optionswerte für verschiedene Steuerungen . . . . .	110
8.3.	Verlauf der Ausübungsgrenze für verschiedene Steuerungen . . . . .	112
8.4.	Visualisierung der Laufzeiten aus Tabelle 8.2 . . . . .	114
8.5.	Visualisierung der Laufzeiten aus Tabelle 8.3 . . . . .	115
8.6.	Details zur Anzahl der gelösten Gleichungssysteme pro Iterationsschritt . .	120
9.1.	Optionswert einer Basket-Option . . . . .	131
9.2.	Informationsfluss für eine diskret gemittelte asiatische Option . . . . .	136

# Tabellenverzeichnis

6.1. Zusammenfassung der Eigenschaften aller vier Verfahren . . . . .	87
7.1. Auswirkungen der Variation von $\alpha$ auf den Aufwand . . . . .	95
8.1. Kurzvergleich der Steuerungsverfahren . . . . .	110
8.2. Vergleich der Laufzeiten für die Algorithmen: Diskretisierung a) . . . . .	114
8.3. Vergleich der Laufzeiten für die Algorithmen: Diskretisierung b) . . . . .	115
8.4. Überslagsweise Anzahl von Operationen, Detail 1 . . . . .	117
8.5. Überslagsweise Anzahl von Operationen, Detail 2 . . . . .	118
8.6. Überslagsweise Anzahl von Operationen, Detail 3 . . . . .	119
C.1. Zahlenwerte bei äquidistanter Zeitschrittweite . . . . .	148
C.2. Zahlenwerte bei gesteuerter Zeitschrittweite . . . . .	148

# Verzeichnis der Algorithmen und Flussdiagramme

## Verzeichnis der Algorithmen

4.1.1	Algorithmus von Dempster, Hutton und Richards (DHR-Algorithmus) .....	48
4.1.2	UL-Update für DHR-Algorithmus .....	48
4.2.5	UL-Update <sup>-1</sup> für DHR-Algorithmus .....	52
5.2.6	Rannacher smoothing .....	60
5.3.7	Automatische Startschrittweite .....	71
6.1.1	Algorithmus von Eriksson und Johnson (EJ-Algorithmus) (Grundform) .....	76
6.1.2	EJ-Algorithmus (1-D Diffusion) .....	77
6.1.6	Implizit/Crank-Nicolson Steuerung .....	80
6.1.8	Implizit/Implizit-Halbe Steuerung .....	82
7.2.1	Schrittweitensteuerung und DHR-Algorithmus (Grundform) .....	97
7.2.2	Schrittweitensteuerung und DHR-Algorithmus (lauffähig) .....	98
9.2.1	Bewertung asiatischer Optionen .....	135

## Verzeichnis der Flussdiagramme

Abb.4.1	Ablauf eines Lösungsschrittes des DHR-Algorithmus .....	49
Abb.4.3	Ablauf eines Lösungsschrittes des modifizierten DHR-Algorithmus .....	53



# Erklärung

Ich versichere, dass ich die von mir vorgelegte Dissertation selbständig angefertigt, die benutzten Quellen und Hilfsmittel vollständig angegeben und die Stellen der Arbeit - einschließlich Tabellen, Karten und Abbildungen -, die anderen Werken im Wortlaut oder dem Sinn nach entnommen sind, in jedem Einzelfall als Entlehnung kenntlich gemacht habe; dass diese Dissertation noch keiner anderen Fakultät oder Universität zur Prüfung vorgelegen hat; dass sie - abgesehen von unten angegebenen Teilpublikationen - noch nicht veröffentlicht worden ist sowie, dass ich eine solche Veröffentlichung vor Abschluss des Promotionsverfahrens nicht vornehmen werde. Die Bestimmungen dieser Promotionsordnung sind mir bekannt. Die von mir vorgelegte Dissertation ist von Prof. Dr. R. Seydel betreut worden.

Köln, im Februar 2005

## Teilpublikation

[IV05] Rainer Int-Veen. Avoiding Numerical Dispersion in Option Valuation.  
*Computing and Visualization in Science*, 10 pages, to appear