

**ZHAW**  
**Zürcher Hochschule für Angewandte Wissenschaften**

School of Management and Law

---

**BACHELORARBEIT**

Bewertung von Floating Strike Lookback-Optionen anhand  
der Finiten Differenzen Methode

---

**vorgelegt von:** Armand Patrice Hofstetter  
**Adresse:** Hotzenmattstrasse 3  
8915 Hausen am Albis  
**Matrikelnummer:** 12-470-274  
**E-Mail-Adresse:** hofstarm@students.zhaw.ch

**eingereicht bei:** Dr. Norbert Hilber

Zürich, 25. März 2016

## **Wahrheitserklärung**

„Ich erkläre hiermit, dass ich die vorliegende Arbeit selbständig, ohne Mithilfe Dritter und nur unter Benützung der angegebenen Quellen verfasst habe und dass ich ohne schriftliche Zustimmung der Studiengangleitung keine Kopien dieser Arbeit an Dritte aushändigen werde.“

Gleichzeitig werden sämtliche Rechte am Werk an die Zürcher Hochschule für angewandte Wissenschaften (ZHAW) abgetreten. Das Recht auf Nennung der Urheberschaft bleibt davon unberührt.

Name der / des Studierenden (Druckbuchstaben)

Armand Patrice Hofstetter

Unterschrift der / des Studierenden

.....

## Herausgabeerklärung der / des Dozierenden

Herausgabe<sup>1)</sup> der Bachelor-Arbeit „Bewertung einer Floating Strike Lookback-Option anhand der Finiten Differenzen Methode“

Die vorliegende Bachelor-Arbeit wird

- nicht herausgegeben.
- nicht herausgegeben werden bis ins Jahr .
- für eine uneingeschränkte Herausgabe freigegeben.

.....  
(Ort, Datum)

.....  
(Unterschrift der / des Dozierenden)

<sup>1)</sup> Unter "Herausgabe" wird sowohl die Einsichtnahme im Hause wie auch die Ausleihe bzw. die Abgabe zu Selbstkostenpreisen verstanden.

---

## Management Summary

Optionen datieren in der Geschichte bereits weit zurück und sind heute ein wichtiges Finanzprodukt für Handelsgeschäfte, Spekulationen und Transaktionen dergleichen. Im heutigen Ausmass hat sich der Derivatehandel jedoch erst in den letzten Jahrzehnten herausgebildet, wobei die neuen Methoden zur Preisbestimmung eine wichtige Rolle spielen.

Im Laufe der Zeit wurden viele Finanzprodukte entwickelt, welche, wie beispielsweise Lookback-Optionen, von ihren Plain Vanilla Gegenstücken abweichen. So hängt die *Floating Strike Lookback-Option* nicht nur vom Preis bei der Ausübung und einem fixen Strike-Preis ab, wie dies die *Europäischen Plain Vanilla Optionen* tun. Vielmehr tritt das Minimum oder Maximum des Basiswertes – gemessen während einer bestimmten Lookback-Periode – an die Stelle des Strike-Preises: daher “*Floating Strike*”.

Für die Berechnung der Extrema gibt es unterschiedliche Ansätze: Es ist beispielsweise möglich von einer kontinuierlichen Stichprobennahme auszugehen, was das Problem vereinfacht und eine exakte Berechnung des Preises mittels einer Closed-Form-Lösung ermöglicht. Im Gegensatz dazu ist die diskrete Stichprobennahme zwar *realitätsnäher*, für die entsprechende Closed-Form-Lösung ist aber nunmehr die multivariate Normalverteilung notwendig. Es bietet sich daher an, auf approximative Methoden zurückzugreifen.

Im vorliegenden Fall wird der Preis mittels der Finiten Differenzen Methode approximiert. Hierbei wird ein partielles Differenzialgleichungssystem, welches die Literatur zur Verfügung stellt, diskretisiert und somit der Preis näherungsweise berechnet. Dieses Verfahren wird mittels der Software *Octave* in zwei Schritten implementiert: (i) Vorerst eine Routine, welche die kontinuierliche Stichprobennahme umsetzt. (ii) Danach eine Erweiterung der Routine, welche der diskreten Stichprobennahme ebenfalls Rechnung trägt. Für (i) wurde basierend auf den Resultaten der Closed-Form-Lösung eine Konvergenz im quadratischen Bereich nachgewiesen, bei (ii) musste für die Verifizierung der Resultate hingegen auf die Literatur ausgewichen werden.

# Inhaltsverzeichnis

<b>Inhaltsverzeichnis</b>	<b>i</b>
<b>I Abbildungsverzeichnis</b>	<b>iii</b>
<b>II Tabellenverzeichnis</b>	<b>iii</b>
<b>III Algorithmenverzeichnis</b>	<b>iii</b>
<b>1 Einleitung</b>	<b>1</b>
1.1 Bewertung von Derivaten – die Problemstellung . . . . .	1
1.2 Aufbau der Arbeit . . . . .	3
<b>2 Was ist eine Lookback-Option?</b>	<b>4</b>
2.1 Plain Vanilla versus Exotische Optionen . . . . .	4
2.2 Floating Strike versus Fixed Strike . . . . .	5
2.3 Amerikanische und Russische Lookback-Optionen . . . . .	6
2.4 Lookback-Perioden und $\lambda_\bullet$ . . . . .	7
<b>3 Methode</b>	<b>9</b>
3.1 Die Ausgangslage . . . . .	9
3.2 $\eta$ – Vereinfachung des Gleichungssystems . . . . .	11
3.3 Finite Differenzen Methode . . . . .	13
3.4 Das $\theta$ -Verfahren . . . . .	18
3.5 Randdaten . . . . .	20
3.6 Intrinsische Randbedingung . . . . .	22
3.7 Variable Koeffizienten . . . . .	24

3.8	Berechnung der Extrema – kontinuierlich versus diskret . . . . .	25
<b>4</b>	<b>Anwendung</b>	<b>27</b>
4.1	Kontinuierliche Stichprobennahme . . . . .	27
4.2	Konvergenzstudie . . . . .	33
4.3	Put-Optionen . . . . .	37
4.4	Diskrete Stichprobennahme . . . . .	38
4.5	Fehler der Approximation . . . . .	40
<b>5</b>	<b>Schlusswort</b>	<b>43</b>
	<b>Literaturverzeichnis</b>	<b>45</b>
<b>A</b>	<b>Anhang</b>	<b>A-i</b>
A.1	Code LBO_float.m . . . . .	A-i
A.2	Code LBO_float_BS.m . . . . .	A-ii
A.3	Code Konvergenzpruefung.m . . . . .	A-iii
A.4	Code LBO_float_diskret_put.m . . . . .	A-iv

## I Abbildungsverzeichnis

4.1	$u$ sowie $g$ in Abhängigkeit von $\eta$ . . . . .	32
4.2	$u$ in Abhängigkeit von $\eta$ in der Zeit zurückgerechnet. . . . .	32
4.3	$\ e_j\ _\infty$ in Abhängigkeit von $N$ . . . . .	37
4.4	$u_j$ an den jeweiligen <i>Sampling-Zeitpunkten</i> . . . . .	42

## II Tabellenverzeichnis

4.1	Veränderung des Fehlers mit zunehmendem $N$ . . . . .	37
4.2	Veränderung des Fehlers mit zunehmendem $N$ und $M$ . . . . .	41

## III Algorithmenverzeichnis

4.1	Routine <code>parabol_dgl_BC.m</code> (Hilber, 2015: S. 171) . . . . .	29
A.1	Routine <code>LBO_float.m</code> . . . . .	A-i
A.2	Routine <code>LBO_float_BS.m</code> . . . . .	A-ii
A.3	Routine <code>Kovertgenzpruefung.m</code> . . . . .	A-iii
A.4	Routine <code>LBO_float_diskret_put.m</code> . . . . .	A-iv

# 1 Einleitung

Optionen, Futures genauso wie andere Derivate sind heute wohlbekannter Finanzjargon. Erste Finanzderivate datieren dabei in der Geschichte schon Jahrzehnte zurück (Poitras, 2009: S. 1-3). Diese Art von “abgeleiteten”<sup>1</sup> Finanzprodukten bietet den Käufern, welche diese Produkte als Absicherungen gegen aversive Marktentwicklungen wie auch aus pur spekulativen Gründen erwerben, vielerlei Möglichkeiten. Lange Zeit hielten sich aber die gehandelten Volumina in einem – im heutigen Vergleich – geringen Rahmen (MacKenzie, 2006: S. 4).

MacKenzie (2006: S. 5, 6) beschreibt wie es an der Akademie war, Finanzen als “Thema von Interessen” zu entdecken. Financial Economics, ein Forschungsgebiet, welches sich in den 1950er-Jahren in den Vereinigten Staaten beging zu etablieren, ist von den bis dahin gewählten primär deskriptiven Vorgehensweisen abgewichen und hat angefangen die Finanzmärkte mit mathematischen Modellen zu analysieren. Eine wichtige Publikation aus diesem Bereich ist das Black-Scholes-Merton-Modell aus dem Jahr 1973, welches einen entscheidenden Beitrag zur Entwicklung des modernen Optionshandels geleistet hat.

Verständlicherweise war der Handel in diesen Produkten nur beschränkt möglich, wenn der Verkaufspreis nicht klar eruiert werden konnte. Warum Optionen einen komplexen Fall darstellen und wie sich die Bewertung einer spezifischen Optionsart abspielen kann, wird im Detail auf den folgenden Seiten zu sehen sein.

## 1.1 Bewertung von Derivaten – die Problemstellung

Die Funktionsweise von Optionen ist nicht allzu schwer zu verstehen: Eine Option (oder zu deutsch eine Möglichkeit) eine Aktie, einen Gegenstand oder dergleichen (zu einem bestimmten Zeitpunkt) zu vordefinierten Bedingungen zu kaufen oder verkaufen. Kommt es jedoch zur Frage, was für ein solches Finanzinstrument ein fairer Preis sein würde, gestaltet sich die Angelegenheit schon ein wenig komplizierter. In diesem Abschnitt wird deshalb summarisch auf die Schwierigkeiten der Optionsbewertung sowie auf die Zielsetzung dieser Arbeit eingegangen.

---

<sup>1</sup>“ab- bzw. hergeleitet” von einem Basiswert



Wenn der Preis einer Option bestimmt werden soll, lässt sich – wie auch bei der Bewertung von anderen Objekten – die Frage stellen, welcher Wert das Gekaufte künftig zu generieren vermag. Der Present Value<sup>2</sup> davon wäre dann ein *fairer Preis* dafür. Das Problem der Option besteht jedoch darin, dass sich der Preis – wie Hilber (2015: S. 43) am Fall einer *Europäischen Plain Vanilla Option* zeigt – durchaus als diskontierter Erwartungswert darstellen lässt, es jedoch schwierig ist, zu bestimmen, welcher Wert genau “erwartet” werden kann. Der Wert, welchen eine Option bei der Ausübung für den Käufer haben wird, ist zwar mittels der Auszahlungsfunktion angeben. Dieser hängt aber vom Basiswert zur Zeit der Ausübung ab und lässt sich daher nicht ohne Weiteres beziffern.

Methoden zur Bewertung gibt es heute viele, so existiert für Europäische Plain Vanilla Optionen eine sogenannte Closed-Form-Lösung, welche als Black-Scholes-Formel weitgehend bekannt ist. Optionen sind jedoch in ihrer Vielfältigkeit gewachsen, so dass es heute in vielen Fällen nicht mehr möglich ist, den Preis anhand von (einfachen) Formeln “von Hand” zu berechnen.

Diese Arbeit befasst sich mit den *Floating Strike Lookback-Optionen*, welche es einem Käufer ermöglichen, einen Basiswert zum minimalen (maximalen) Preis zu kaufen (verkaufen). Lookback-Optionen finden sich beispielsweise als Bestandteil von strukturierten Produkten wie dem Lookback PERLES Plus der UBS AG (2014). Es handelt sich hierbei um sogenannte exotische Optionen, von welchen über die Jahre viele entwickelt worden sind. Lookback-Optionen sind ein Beispiel eines Produkts, welches sich unter Umständen nur schwer anhand einer Formel bewerten lässt und für welches es sinnvoll ist auf numerische Verfahren zurückzugreifen.

Für Lookback-Optionen existiert eine partielle Differenzialgleichung, welche in ihrer Art der ursprünglichen Black-Scholes-Merton-Gleichung sehr ähnlich ist: In gewissen Fällen lassen sich solche Gleichungen für den Preis der Option auflösen, wie dies Black und Scholes für *Europäische Plain Vanilla Optionen* gemacht haben. Ist das nicht möglich lässt sich die Black-Scholes-Gleichung mittels der Finiten Differenzen Methode<sup>3</sup> diskretisieren und ein Näherungswert des Preises berechnen.

Ziel dieser Arbeit ist es, Routinen für die Bewertung von Lookback-Optionen<sup>4</sup> mittels der

---

<sup>2</sup>der Gegenwartswert

<sup>3</sup>Andere Methoden wie zum Beispiel *Monte-Carlo-Simulationen* werden ebenfalls zur Bewertung von Optionen verwendet; diese werden aber im Rahmen dieser Arbeit nicht angeschaut.

<sup>4</sup>Wird auf den folgenden Seiten Bezug auf *Lookback-Optionen* genommen, so bezieht sich dies jeweils auf *Europäische Floating Strike Lookback-Optionen*, sofern nicht ausdrücklich auf andere Arten von (Lookback-)Optionen hingewiesen wird.

Software *Octave* zu entwickeln. Diese sollen es ermöglichen Floating Strike Lookback-Optionen sowohl unter Annahme von kontinuierlicher Stichprobennahme wie auch unter Annahme von diskreter Stichprobennahme zu bewerten. Diese Bewertung soll unter Anwendung der Finiten Differenzen Methode approximativ erfolgen.

## 1.2 Aufbau der Arbeit

Nachdem in den letzten Absätzen die Problemstellung der Bewertung von Derivaten sowie die Zielsetzung der vorliegenden Arbeit genauer erläutert wurde, soll in diesem Abschnitt das weitere Vorgehen und der Aufbau der Arbeit beschrieben werden.

Im Anschluss hieran soll in Kapitel 2 erklärt werden, wie Lookback-Optionen funktionieren, wie sie sich von anderen Optionen abheben und wie sich Lookback-Optionen untereinander abgrenzen lassen.

Kapitel 3 geht auf *theoretischer Ebene* auf die Methodik ein, welche verwendet wird, um die Floating Strike Lookback-Optionen zu bewerten. Es wird somit der Prozess vom partiellen Differenzialgleichungssystem hin zum *volldiskreten Verfahren* beschrieben. Dies wird wiederum in Kapitel 4 aufgegriffen und mittels *Octave* Schritt für Schritt implementiert.

Zu guter Letzt werden im 5. Kapitel die Resultate der vorliegenden Arbeit nochmals zusammengefasst und abschliessend kommentiert.

## 2 Was ist eine Lookback-Option?

Wie im vorangehenden Kapitel erwähnt, ist es Ziel dieser Arbeit, eine *Floating Strike Lookback-Option* zu bewerten. Bevor jedoch auf die Berechnung des Preises eingegangen werden kann, stellt sich die Frage, was eine Lookback-Option ist und wie sich die *Floating Strike* von anderen Lookback-Optionen abhebt.

### 2.1 Plain Vanilla versus Exotische Optionen

Lookback-Optionen gehören zu den exotischen pfadabhängigen Finanzderivaten. Um genauer zu verstehen, was ein exotisches Finanzderivat ist, wird im Folgenden der Unterschied zwischen Plain Vanilla und exotischen Optionen erläutert.

Eine Call-(Put-)Option<sup>1</sup> gesteht dem Käufer das Recht zu, einen Basiswert (in vielen Fällen Aktien oder Indices) zu einem Zeitpunkt in der Zukunft zu kaufen (verkaufen). Da eine Option ein Vertrag ist, welcher sich auf einen Basiswert bezieht, und der Wert sich somit von diesem herleitet, gehört sie zu den Finanzderivaten<sup>2</sup>.

Um eine “herkömmliche” Plain Vanilla<sup>3</sup> Option handelt es sich, wenn bei der Ausübung jeweils die Wahl zwischen dem aktuellen Marktpreis und einem fixen Ausübungspreis besteht, sowie keine weiteren Sonderkonditionen anwendbar sind<sup>4</sup>. Dies bedeutet, dass die Auszahlungsfunktion ( $g$ ) durch

$$g(S) = \max(S - X, 0)$$

im Falle einer Call-Option beziehungsweise durch

$$g(S) = \max(X - S, 0)$$

im Falle einer Put-Option gegeben ist (Zhu et al., 2010: S. 10). Die Auszahlungsfunktion  $g$  definiert die Auszahlung beziehungsweise den Wert der Option bei Ausübung und hängt

---

<sup>1</sup>Option bedeutet zu Deutsch “Wahl” oder “Möglichkeit”.

<sup>2</sup>“Derive” ist das englische Wort für “ableiten”.

<sup>3</sup>Der Begriff Plain Vanilla kommt aus dem Englischen und beschreibt umgangssprachlich ein Objekt, welches keine speziellen Merkmale aufweist (Plain Vanilla, 2000).

<sup>4</sup>Die Subsumtion von Optionen unter dem Begriff “Plain Vanilla” ist nicht trennungsscharf, die Begriffe werden jedoch analog zu Zhu, Wu, Chern und Sun (2010: S. 159) verwendet.

vom Basiswert  $S$  sowie vom Ausübungspreis  $X$  ab.

Plain Vanilla Optionen lassen sich weiter nach Ausübungszeitpunkt unterscheiden. Die ersten Optionen waren in ihrer Art vergleichbar mit den heute als Europäisch bezeichneten Optionen, das heißt, dass sie nur zu einem vordefinierten Zeitpunkt (bei Maturität) ausgeübt werden konnten. Heutzutage sind die meisten gehandelten Optionen dahingegen von amerikanischer “Natur”, was bedeutet, dass sie zu jedem beliebigen Zeitpunkt zwischen Emission und Ablauf der Option ausgeübt werden können (Poitras, 2009: S. 1-3, 19).

Im Verlaufe der Zeit haben Finanzinstitute neue Derivate entwickelt. Somit haben sich neben den Plain Vanilla Optionen auch neue, sogenannte “exotische” Optionen, herausgebildet. Diese Optionen werden vorwiegend OTC (over the counter) gehandelt, was bedeutet, dass sie nicht an der Börse erhältlich sind, sondern vielmehr direkt zwischen den Banken und Firmen gehandelt werden (Zhu et al., 2010: S. 159). Wie Zhu et al. (2010: S. 159) erläutern sind die meisten exotischen Optionen pfadabhängig, was bedeutet, dass die Auszahlungsfunktion bei der Ausübung nicht nur vom Endkurs des Basiswertes und einem fixen Ausübungspreis abhängt, sondern auch vom Preisverlauf des Basiswertes. Hierbei kann es sein, dass der “Pfad des Kurses” über die ganze Laufdauer des Derivats angeschaut wird, oder auch nur über einen beschränkten Zeitraum.

Neben anderen Optionsarten sowie Asiatischen Optionen oder Barrier Optionen ist auch die Lookback-Option eine sogenannte exotische pfadabhängige Option, da die Auszahlungsfunktion bei der Ausübung vom Minimum/Maximum des Basiswertes (je nach Fall über die ganze Laufdauer oder nur einem Teil davon gemessen) abhängt. Diese Charakteristik stellt – wie in Kapitel 3 diskutiert wird – für die Preisberechnung einige Schwierigkeiten dar.

## 2.2 Floating Strike versus Fixed Strike

Lookback-Optionen zeichnen sich dadurch aus, dass der “Payoff” (das heißt die Auszahlung) bei der Ausübung vom Minimum/Maximum des Basiswertes abhängt. Nun lassen sich vielerlei Produkte mit dieser Charakteristik konstruieren, weshalb im Folgenden genauer auf die Unterschiede zwischen den verschiedenen Lookback-Optionen eingegangen wird.

Im Vergleich zur erwähnten *Europäischen Plain Vanilla Option* kann das Minimum/Ma-

ximum in der Auszahlungsfunktion an die Stelle des “Strike-Preises” oder des Basiswertes treten. Ersteres wird als Floating Strike Lookback-Option beziehungsweise Lookback Strike Option bezeichnet und zahlt bei Ausübung

$$g(S, L) = \max(\lambda S - L, 0)$$

respektive

$$g(S, H) = \max(H - \lambda_p S, 0)$$

aus. Hierbei wird das Maximum mit  $H$  für *high* und das Minimum mit  $L$  für *low* bezeichnet;  $\lambda$  steht für eine Konstante aus dem Intervall  $]0, 1]$  und  $\lambda_p \geq 1$ . Auf  $\lambda$  wird in den folgenden Abschnitten nochmals eingegangen. Mit  $\lambda_\bullet = 1$  ermöglicht es die Floating Strike Lookback-Option dem Käufer den Basiswert zum Maximal-(Minimal-)Preis zu verkaufen (kaufen).

Im Gegensatz zur Floating Strike Lookback-Option ist die Auszahlungsfunktion der Fixed Strike Lookback- beziehungsweise Lookback Price Option wie folgt definiert:

$$g(H) = \max(H - X, 0)$$

$$g(L) = \max(X - L, 0).$$

Der Kauf einer solcher Option würde Sinn machen, wenn sich ein Käufer über den “Bereich”, in welchem sich der Basiswert bewegen wird, sicher ist aber nicht über das Timing. Das Risiko des Timings besteht nicht, da die Auszahlungsfunktion nicht bloss vom Preis zu einem bestimmten Zeitpunkt abhängt, sondern vielmehr von einem Extremwert über eine längere Periode gemessen (Kwok, 2008: S. 202).

Die vorliegende Arbeit wird sich ausschliesslich mit Floating Strike Lookback-Optionen befassen; die entsprechenden Auszahlungsfunktionen werden auf den folgenden Seiten wieder anzutreffen sein.

### 2.3 Amerikanische und Russische Lookback-Optionen

Wie bereits in Abschnitt 2.1 erwähnt, gibt es Europäische sowie auch Amerikanische Plain Vanilla Optionen, welche sich im Hinblick auf ihre Ausübungsmöglichkeit unterscheiden: Erstere lässt sich nur bei Maturität ausüben, letztere schon früher. Wohingegen bei den Ausführungen soweit jeweils implizit angenommen wurde, dass es sich um Europäische Lookback-Optionen handelt, lassen sich Lookback-Optionen auch mit den fle-

xibleren Ausübungsmöglichkeiten einer Amerikanischen Option kombinieren. Dies wirkt sich nicht auf die Auszahlungsfunktion aus, sondern nur auf die Ausübungsmöglichkeiten.

Ist nun eine Lookback-Option Amerikanischer Natur und somit jederzeit ausübbar, jedoch in der Laufzeit unbeschränkt, so handelt es sich um eine Russische (Lookback-) Option (Zhu et al., 2010: S. 210). Diese Arbeit befasst sich ausschliesslich mit Europäischen Lookback-Optionen und wird im weiteren Verlauf nicht weiter auf Amerikanische und Russische (Lookback-)Optionen eingehen.

### 2.4 Lookback-Perioden und $\lambda_\bullet$

Die Auszahlungsfunktionen der *Floating Strike Lookback-Optionen* sind für die Käufer sehr attraktiv, was dazu führt, dass ein potentieller Emittent ein entsprechend hohes Premium verlangen muss. Sollte jedoch das Bedürfnis bestehen, ein günstigeres Derivat zu emittieren, so ist dies beispielsweise über die Wahl der *Lookback-Periode* oder der Einführung eines *Multiplikators* in der Auszahlungsfunktion möglich. Diese Möglichkeiten werden im Folgenden genauer erläutert:

Ersteres würde bedeuten, dass das Maximum/Minimum nicht über die gesamte Laufdauer gemessen wird, sondern nur während eines Teils derer. Solche Optionen können auch als Limited Period Lookback-Optionen bezeichnet werden und würden sich weiter danach unterscheiden lassen, welche Periode innerhalb der Laufzeit effektiv als Lookback-Periode verwendet wird. Für einen Investor, der sich beispielsweise über das optimale Timing für den Markteintritt unsicher ist, kann eine Limited Period Lookback-Option mit Lookback-Periode früh in der Laufdauer interessant sein. So sichert sich der Investor gegen den Fall ab, in welchem er den Basiswert zu früh kauft, da der Preis nochmals fällt, bevor er gegen Ende der Laufzeit wieder zunimmt (Kwok, 2008: S. 202).

Ähnlich lässt sich auch mit einem  $\lambda_\bullet \neq 1^5$  die Auszahlungsfunktion so "steuern", dass sie für den Käufer nicht mehr gleich vorteilhaft ist und somit das nötige Premium tiefer angesetzt werden könnte, das heisst, die Option günstiger wird. Dies lässt sich insofern nachvollziehen, als dass die Auszahlungsfunktion mit der Differenz<sup>6</sup> zwischen  $\lambda S$  und dem minimalen Preis des Basiswertes  $S$ , also  $L$ , gegeben ist<sup>7</sup>. Wird nun  $\lambda = 1$  gesetzt, kommt  $\lambda$  nicht zum Tragen und die Auszahlungsfunktion bräuchte das  $\max(\bullet, 0)$  nicht,

---

<sup>5</sup>Vergleiche hierzu die entsprechenden Auszahlungsfunktionen.

<sup>6</sup>Es wird an dieser Stelle exemplarisch eine Lookback-Call-Option angeschaut.

<sup>7</sup>Sofern dieser Wert  $\geq 0$  ist.

## 2 Was ist eine Lookback-Option?

---

da ein Wert keine tieferen Werte als sein eigenes Minimum annehmen kann<sup>8</sup>. In anderen Worten, es wird eine Lookback-Option betrachtet, welche nicht wirklich eine Option ist: Die Frage der Ausübung stellt sich in diesem Fall nicht, da sie immer sinnvoll ist. Eine solche Option ist entsprechend teuer (Kwok, 2008: S. 202).

Wird nun jedoch der Basiswert in der Auszahlungsfunktion mit einem  $\lambda$  im Intervall  $]0,1[$  multipliziert, besteht die Möglichkeit, dass die Differenz zum Minimum  $L$  tatsächlich negativ sein kann. Abgesehen davon ist der *Payoff* für jeden Basiswert  $S$  kleiner als mit einem  $\lambda = 1$  und der Preis der Option wird entsprechend geringer ausfallen. Das  $\lambda$  lässt sich durch den Emittenten wählen.

Im Rahmen der vorliegenden Arbeit, wird die Auszahlungsfunktion in Abhängigkeit von  $\lambda_{\bullet}$  für die Bewertung verwendet, jedoch für Berechnungen jeweils  $\lambda_{\bullet} = 1$  gewählt. Auf Limited Period Lookback-Optionen wird nicht eingegangen.

Ist somit im weiteren Verlaufe dieser Arbeit von einer (Floating Strike) Lookback-Option die Rede, bezieht sich dies jeweils auf eine Europäische Floating Strike Lookback-Option, deren Preis von einem Maximum beziehungsweise Minimum abhängt, welches über die gesamte Laufzeit berechnet wird. Zudem wird jeweils darauf hingewiesen, ob es sich bei den Rechenbeispielen um Call- oder Put-Optionen handelt und der Einfachheit halber ein  $\lambda_{\bullet}$  von 1 gewählt. Des Weiteren wird angenommen, dass die Bewertung zur Zeit der Emission erfolgt. Basierend auf diesen Angaben lassen sich nun im Rahmen der nächsten Kapiteln die theoretischen und technischen Aspekte der Preisfindung diktuieren.

---

<sup>8</sup>Sofern die selben Zeitspannen angeschaut werden: Ein Wert kann natürlich jeder Zeit einen Wert annehmen, der geringer ist als das Minimum bis anhin, in diesem Fall entsteht jedoch ein neues Minimum.

## 3 Methode

Im Folgenden wird nun die Methodik, mithilfe welcher die Bewertung der Lookback-Optionen durchgeführt wird, vorgestellt. Somit wird anhand dem von Zhu et al. (2010: S. 185ff) hergeleiteten partiellen Differenzialgleichungssystem und der durch Hilber (2015) beschriebenen Methode vorgegangen.

### 3.1 Die Ausgangslage

Im Rahmen dieser Bachelorarbeit wird die Finite Differenzen Methode zur Bewertung von Optionen verwendet. Hierbei wird von einer partiellen Differenzialgleichung, wie beispielsweise der Black-and-Scholes-Gleichung ausgegangen und aufgrund derselben eine approximative Berechnung des Preises vorgenommen.

Dies ist nicht in jedem Fall notwendig, da für einige Optionen sogenannte Closed-Form-Lösungen vorhanden sind. Beispielsweise zeigen Zhu et al. (2010: S. 76-80), wie es möglich ist, die Black-Scholes(-Differenzial)-Gleichung für Europäische Optionen zu lösen, woraus die wohlbekannte Black-Scholes-Formel resultiert. Diese ermöglicht es, den Preis von Europäischen Optionen direkt zu berechnen, was eine Approximation überflüssig macht. Dieselbe ist durch

$$V_c(S, t) = S e^{-q(T-t)} \Phi_{0,1}(d_1) - X e^{-r(T-t)} \Phi_{0,1}(d_2) \quad (3.1)$$

mit

$$d_1 := \frac{\ln\left(\frac{S}{X}\right) + \left(r - q + \frac{\sigma^2}{2}\right)(T-t)}{\sigma\sqrt{T-t}}$$

und

$$d_2 := d_1 - \sigma\sqrt{T-t}$$

gegeben. Hierbei wird der Einheitlichkeit halber die Notation nach Hilber (2015: S. 5) gewählt und somit  $\Phi_{\mu, \sigma}$  als Bezeichnung der kumulierten Häufigkeitsverteilung der Normalverteilung verwendet. Des Weiteren entspricht  $r$  dem risikolosen Zinssatz,  $\sigma$  der Volatilität des Basiswertes,  $T$  der Laufzeit der Option,  $X$  dem Ausübungspreis<sup>1</sup> und  $q$  der stetigen Dividendenrendite.  $t$  steht für die bereits “vergangene” Laufzeit so, dass  $T - t$  die

---

<sup>1</sup>Auch “Strike-Preis” genannt.



Restlaufzeit bezeichnet<sup>2</sup>.

Solche Closed-Form-Lösungen existieren nur für gewisse Optionen, sind unter Umständen weitaus komplexer als Gleichung (3.1) und/oder decken nur Spezialfälle ab. Für Lookback-Optionen existiert beispielsweise eine Closed-Form-Lösung. Wird jedoch von einer diskreten Stichprobennahme für die Feststellung des Minimums/Maximums ausgegangen, bedarf es der  $n$ -variater Normalverteilung (wobei  $n$  für die Anzahl Sampling-Punkte steht).

Wird wiederum eine entsprechende Differenzialgleichung mit der Finiten Differenzen Methode diskretisiert, lassen sich Approximationen auch in einem Szenario mit diskreten Stichprobennahmen mit verhältnismässig geringer Rechenleistung durchführen. Kwok (2008: S. 209ff) zeigt wie sich die Black-Scholes-Gleichung für den Fall einer Lookback-Option erweitern lässt. Das partielle Differenzialgleichungssystem in (3.2) entspricht der durch Kwok hergeleiteten Gleichung mit der Erweiterung für  $q^3$  (Zhu et al., 2010: S. 190). Diese wird im folgenden Kapitel in angepasster Form für die Approximation des Lookback-Optionspreises verwendet.

$$\begin{cases} \partial_t V + \frac{1}{2} \sigma^2 S^2 \partial_{SS}^2 V + (r - q) S \partial_S V - rV = 0, & 0 \leq L \leq S, 0 \leq t \leq T \\ V(S, L, T) = \max(\lambda S - L, 0), & 0 \leq L \leq S \\ \partial_L V(S, S, t) = 0, & 0 \leq S, 0 \leq t \leq T \end{cases} \quad (3.2)$$

$\lambda \in [0, 1]$  steht an dieser Stelle für den Multiplikator, welcher bereits in Kapitel 2 andiskutiert wurde.

Die entsprechende Closed-Form-Lösung ist für Lookback-Optionen durch

$$\begin{aligned} V_{C,fl}(S, t, m) = & S \Phi_{0,1}(d_m) - e^{-r(T-t)} m \Phi_{0,1}(d_m - \sigma \sqrt{T-t}) \\ & + e^{-r(T-t)} \frac{\sigma^2}{2r} S \left[ \left( \frac{S}{m} \right)^{-\frac{2r}{\sigma^2}} \Phi_{0,1} \left( -d_m + \frac{2r}{\sigma} \sqrt{T-t} \right) \right. \\ & \left. - e^{r(T-t)} \Phi_{0,1}(-d_m) \right] \end{aligned} \quad (3.3)$$

---

<sup>2</sup>Die Bedeutung dieser Parameter wird für den weiteren Verlauf dieser Arbeit bestand haben, sofern nicht ausdrücklich auf Änderungen der Definitionen hingewiesen wird.

<sup>3</sup> $q$  entspricht der stetigen Dividendenrendite, welche von einer kontinuierlich-ausbezahlten Dividende ausgeht. Diese Annahme ist zwar nicht realistisch, ermöglicht jedoch Dividenden-Auszahlungen leicht einzubeziehen. Eine Umrechnung von diskreten Dividendenzahlungen zu kontinuierlichen ist möglich (Hilber, 2015: S.5).

mit

$$d_m := \frac{\ln\left(\frac{S}{m}\right) + \left(r + \frac{\sigma^2}{2}\right)(T - t)}{\sigma\sqrt{T - t}}$$

(für den Fall einer Call-Option<sup>4</sup>) gegeben (Kwok, 2008: S. 205). Variable  $m$  steht für das *Minimum bis anhin*, das heisst das Minimum des Basiswertes gemessen für die Periode von der Emission bis zum Zeitpunkt der Bewertung. Da im vorliegenden Fall eine Bewertung zum Zeitpunkt der Emission angeschaut wird, entspricht  $m$  dem Basiswert  $S$ . Gleichung (3.3) geht von einer kontinuierlichen Stichprobennahme für das Minimum<sup>5</sup> des Basiswertkurses aus und bezieht die Dividendenrendite  $q$ <sup>6</sup> nicht mit ein.

Gleichung (3.3) wird es ermöglichen, die mit der Finiten Differenzen Methode implementierte approximative Lösung der Gleichung (3.2) auf Konvergenz (Konvergenz[-raten] werden im Folgenden genauer angeschaut) zu prüfen. In anderen Worten, ermöglicht sie es, festzustellen, ob die approximative Lösung sich dem genauen Wert annähert und wie schnell dies der Fall ist.

## 3.2 $\eta$ – Vereinfachung des Gleichungssystems

Bevor im Rahmen von 3.3 die Finite Differenzen Methode genauer vorgestellt wird, sollen an dieser Stelle zwei Anpassungen am partiellen Differenzialgleichungssystem (3.2) vorgenommen werden, welche es überhaupt (technisch) ermöglichen mit der durch Hilber (2015: S. 93-181) beschriebenen Methode eine approximative Lösung für  $V$  zu berechnen.

Bei der Betrachtung von (3.2) wird klar, dass einer der Hauptunterschiede zur “*ordinären*” Europäischen Option darin besteht, dass eine Abhängigkeit nicht nur von  $t$  und  $S$ , sondern gleichfalls von  $L$  besteht. Nun ist es Variable  $L$ <sup>7</sup>, welche es verunmöglichen würde, das partielle Differenzialgleichungssystem (3.2) mit Hilber (2015: S. 93-181)’s Methode<sup>8</sup> ohne weiteres zu approximieren. Das Problem besteht darin, dass  $L$  eine dritte Variable darstellt, welche ebenfalls zu “modellieren” wäre. Die Methode in Abschnitt 3.3 und 3.4 befasst sich in der angegebenen Form aber nur mit einer Variable für den Basis-

<sup>4</sup>Es wird im Folgenden exemplarisch der Fall einer Lookback-Call-Option angeschaut; die Bewertung einer Put-Option lässt sich analog dazu bewerkstelligen. Die wichtigsten Unterschiede werden fortzu erläutern.

<sup>5</sup>Minimum, da es sich um die Formel für eine Call-Option handelt. Im Falle einer Put-Option wäre die Auszahlungsfunktion vom Maximum abhängig.

<sup>6</sup>Eine Closed-Form-Lösung, welche die Dividendenrendite berücksichtigt, wird durch Zhu et al. (2010: S. 207) bereitgestellt. Auf diese wird im Rahmen dieser Arbeit jedoch nicht weiter eingegangen.

<sup>7</sup>Das heisst der Minimalpreis des Basiswertes während der Laufdauer des Derivates.

<sup>8</sup>Dieselbe wird unter 3.3 und 3.4 genauer vorgestellt.

wert und der zusätzlichen Variable  $t$  für die Zeit. Es wäre demnach von Vorteil, wenn das Problem vereinfacht werden könnte.

Zhu et al. (2010: S. 191) lösen diese Problematik mittels Einführung der Variablen  $\eta$ , so dass

$$\eta = \frac{L}{S} \quad (3.4)$$

und

$$W = \frac{V}{S}. \quad (3.5)$$

Die Einführung der Variable  $\eta$  führt dazu, dass eine neue Funktion  $W$  nun zu approximieren ist, welche jedoch nur von den zwei Variablen  $\eta$  und  $t$  abhängt und somit durch die in den nächsten Abschnitten vorgestellten Methode berechnet werden kann. Es ist jedoch zu beachten, dass  $W$  nicht mehr den Optionspreis darstellt, sondern das Verhältnis desselben zum Preis. Es gilt nun (3.2) entsprechend zum partiellen Differenzialgleichungssystem

$$\begin{cases} \partial_t W + \frac{1}{2} \sigma^2 \eta^2 \partial_{\eta\eta}^2 W + (q - r) \eta \partial_{\eta} W - qW = 0, & 0 \leq \eta \leq 1, 0 \leq t \leq T, \\ W(\eta, T) = \max(\lambda - \eta, 0), & 0 \leq \eta \leq 1, \\ \partial_{\eta} W(1, t) = 0, & 0 \leq t \leq T \end{cases} \quad (3.6)$$

zu erweitern, für welches  $W$  die Lösung ist (Zhu et al., 2010: S. 191f).

An dieser Stelle ist es nun aus technischen Gründen notwendig die ‘‘Richtung der Zeit’’ umzukehren, beziehungsweise die Gleichung in Abhängigkeit der *Restlaufzeit* umzuformulieren. Hierzu wird Variable  $\tau := T - t$  eingeführt, wobei  $\tau = 0$  bei Maturität und  $\tau = T$  bei Emission. Da somit  $t = T - \tau$ , lässt sich

$$w(\eta, \tau) := W(\eta, T - \tau) \quad (3.7)$$

definieren, was dazu führt, dass  $-\partial_t W = \partial_{\tau} w$  (vergleiche Kettenregel) und somit

$$\begin{cases} \partial_{\tau} w - \frac{1}{2} \sigma^2 \eta^2 \partial_{\eta\eta}^2 w - (q - r) \eta \partial_{\eta} w + qw = 0, & 0 \leq \eta \leq 1, 0 \leq \tau \leq T, \\ w(\eta, 0) = \max(\lambda - \eta, 0), & 0 \leq \eta \leq 1, \\ \partial_{\eta} w(1, \tau) = 0, & 0 \leq \tau \leq T. \end{cases} \quad (3.8)$$

Der Einfachheit halber, wird an dieser Stelle  $\tau$  wieder in  $t$  umbenannt. Somit bezeichnet  $t$  fortan die Restlaufzeit der jeweiligen Option.

### 3.3 Finite Differenzen Methode

Wie in Abschnitt 3.1 erwähnt, würde es sich unter Umständen schwierig gestalten, Lookback-Optionen mit der Closed-Form-Formel zu bewerten: Andere Methoden erscheinen sinnvoll. Im Rahmen der vorliegenden Arbeit wird anhand von Gleichungssystem (3.8) der Preis approximativ bestimmt. Im Folgenden wird hierzu die Methodik genauer erläutert.

Diese Arbeit bezweckt, den Wert von Lookback-Optionen zu berechnen. Derselbe ist in (3.2) und (3.3) mit  $V$  beziehungsweise in (3.5) mit  $S_w$  bezeichnet<sup>9</sup>. Das intuitivste Vorgehen wäre deshalb, Gleichung (3.2) nach  $V$  aufzulösen, was im Fall von (3.3) gemacht wurde. Für Lookback-Optionen führt dies aber dazu, dass – sobald für das Extremum diskrete Stichproben genommen werden – die multivariate Normalverteilung zu verwenden ist. Dies gestaltet die Berechnung wiederum schwierig.

Es bietet sich an, auf die Differenzialgleichung in (3.2) zurückzugreifen. Da diese jedoch als partielle Differenzialgleichung nicht nur von  $V$  selbst, sondern genauso von den partiellen Ableitungen ( $\partial V$ ) erster und zweiter Ordnung derselben abhängt, lässt sich  $V$  nicht ohne weiteres berechnen. Aus diesem Grund wird die Finite Differenzen Methode gewählt, um die gesuchten  $V$ -Werte aus Gleichung (3.2) beziehungsweise  $S_w$ -Werte aus der transformierten Differenzialgleichung in (3.8) zu “eruieren”.

Bei der finiten Differenz handelt es sich – allgemein ausgedrückt – um die Steigung einer Sekanten, welche die Tangente approximiert. Die Steigung dieser Tangente ist wiederum die Ableitung einer Funktion an einer bestimmten Stelle (Hilber, 2015: S. 96). Die Finite Differenzen Methode ist also eine Methodik, um Ableitungen annäherungsweise zu berechnen. So kann beispielsweise die erste Ableitung einer Funktion  $f(x)$ <sup>10</sup>, welche durch

$$f'(x) = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h} \quad (3.9)$$

gegeben ist, anhand von

$$f'(x) \approx \frac{f(x+h) - f(x-h)}{2h} \quad (3.10)$$

approximiert werden. Dies weist im Gegensatz zum Differenzenquotienten  $\frac{f(x+h)-f(x)}{h}$  eine quadratische Konvergenz<sup>11</sup> aus und führt somit schneller zu Annäherungen mit der gewünschten Präzision.  $\frac{f(x+h)-f(x)}{h}$  wird in diesem Zusammenhang auch als Vorwärtsdif-

<sup>9</sup>Subskripte werden hierbei der Einfachheit halber ignoriert.

<sup>10</sup>Es werden der Einfachheit halber “allgemeine Notationen” für diese Ausführung verwendet.

<sup>11</sup>Auf die Konvergenzrate der Finiten Differenzen Methode wird an dieser Stelle nicht weiter eingegangen, anstatt dessen wird auf Hilber (2015: S. 98) verwiesen.

ferenzenquotient und (3.10) als zentraler Differenzenquotient an der Stelle  $x$  bezeichnet. Die beiden unterscheiden sich dadurch, dass beim Vorwärtsdifferenzenquotienten  $x$  selbst sowie ein zweite Stelle (um  $h$  entfernt von  $x$ ) die Punkte sind, an welchen die Sekante den Funktionsgraphen für  $f(x)$  schneidet. Beim zentralen Differenzenquotienten stellt  $x$  kein Schnittpunkt der Sekante mit dem Funktionsgraphen dar: Es werden zwei Punkte je um  $h$  entfernt von  $x$  gewählt.

Hilber (2015: S. 99) zeigt, wie sich (3.10) aus den Taylorentwicklungen von  $f$  an den Stellen  $x \pm h$  ergibt, welche durch

$$f(x \pm h) \approx f(x) \pm hf'(x) + \frac{1}{2}(\pm h)^2 f''(x) \quad (3.11)$$

gegeben sind. Wird nun die Taylorentwicklung für  $x + h$  von derjenigen für  $x - h$  subtrahiert, so resultiert

$$f(x + h) - f(x - h) \approx 2hf'(x). \quad (3.12)$$

(3.12) kann nun wiederum nach  $f'(x)$  aufgelöst werden, was zu Approximation (3.10) führt. Hilber (2015: S. 99) zeigt zudem, wie die Approximation für  $f''(x)$  auf die gleiche Weise hergeleitet werden kann. Dieselbe ist durch (3.13) gegeben.

$$f''(x) \approx \frac{f(x - h) - 2f(x) + f(x + h)}{h^2} \quad (3.13)$$

Es ist mit den eingeführten zentralen Differenzenquotienten nun also möglich  $\partial_{\eta\eta}^2 w$  und  $\partial_{\eta} w$  an einer bestimmten Stelle  $x$  beziehungsweise im vorliegenden Fall  $\eta$  zu approximieren. Da es sich bei (3.8) um ein partielles Differenzialgleichungssystem handelt, gilt es jedoch gleichermassen  $\partial_t w$  zu approximieren. Zudem soll die Funktion nicht nur an *einem* bestimmten Punkt näherungsweise berechnet werden, sondern an einer Vielzahl von Punkten<sup>12</sup>, welche jeweils  $h$  voneinander entfernt sind.

Dies lässt sich mit einem sogenannten äquidistanten Gitter bewerkstelligen<sup>13</sup>, welches  $N$  Punkte zwischen zweier Randpunkten (in diesem Beispiel  $\eta_l$  und  $\eta_r$ ) so wählt, dass sie jeweils  $h$ <sup>14</sup> voneinander entfernt sind. Wobei die Maschenweite mit

$$h := \frac{\eta_r - \eta_l}{N + 1} \quad (3.14)$$

<sup>12</sup>Der Wert, welchen der Basiswert bei der Maturität annehmen wird, ist zu diesem Zeitpunkt noch nicht bekannt. Somit wird die Funktion an vielerlei Stellen approximiert und dann in der Zeit zurückgerechnet.

<sup>13</sup>Äquidistant, da die Maschenweite von  $h$  konstant bleibt.

<sup>14</sup>Da der *Vorwärtsdifferenzenquotient*  $\rightarrow f'(x)$  wenn  $h \rightarrow 0$  [vergleiche (3.9)], ist anzunehmen, dass eine Annäherung mit kleineren  $h$ -Werten beziehungsweise geringerer Maschenweite immer genauer wird.

gegeben ist und die Punkte im äquidistanten Gitter im Intervall  $G = [\eta_l, \eta_r]$  sich wie folgt darstellen lassen (Hilber, 2015: S. 100):

$$\eta_l = \eta_0 < \eta_1 < \eta_2 < \dots < \eta_N < \eta_{N+1} = \eta_r \quad (3.15)$$

In diesem Zusammenhang wird von einer *Lokalisierung* gesprochen, das heisst, ein unendlicher Bereich (zum Beispiel der Menge der reellen Zahlen  $\mathbb{R}$ ) wird auf das Intervall  $G$  eingeschränkt, da sich eine Approximation aus offensichtlichen Gründen nicht für eine unbeschränkte Zahl an Werten vornehmen lässt. Das Intervall  $G$  ist frei wählbar, es ist jedoch zu beachten, dass die Funktionswerte an den Stellen  $\eta_l$  und  $\eta_r$  bekannt sein müssen für alle  $t \geq 0$ . Im Fall von (3.8) findet sich die vereinfachte Bedingung vor, dass  $\eta$  bereits auf das Intervall  $[0, 1]$  beschränkt ist, was jedoch für *Lookback-Put-Optionen* nicht zutrifft<sup>15</sup>.

Häufig ist es nicht möglich, den Funktionswert an den Randpunkten zu kennen, weshalb beispielsweise ein Wert von 0 angenommen werden kann. Es liesse sich dann jedoch nicht mehr von der selben Funktion sprechen, da nunmehr eine Funktion gesucht würde, welche an den Randstellen den Wert 0 annimmt. Dies ist bei  $w$  nicht der Fall. Der guten Ordnung halber wird (3.8) deshalb zu

$$\begin{cases} \partial_t u - \frac{1}{2} \sigma^2 \eta^2 \partial_{\eta\eta}^2 u - (q - r) \eta \partial_{\eta} u + qu = 0, & \text{in } G \times [0, T], \\ u(\eta, 0) = \max(\lambda - \eta, 0), & \text{in } G, \\ \partial_{\eta} u(1, t) = 0, & \text{in } [0, T], \end{cases} \quad (3.16)$$

transformiert, wobei  $u$  eine Funktion ist, welche von  $\eta \in G = [\eta_l, \eta_r]$  abhängt.

Auf die Randbedingungen wird erst unter 3.5 genauer eingegangen; es wird für die Ausführungen in 3.3 und 3.4 vereinfacht angenommen, dass die Funktionswerte in  $\eta_l$  und  $\eta_r$  jeweils 0 sind (Hilber, 2015: S. 117f).

Die Differenzialgleichung in (3.16) lässt sich wie folgt umschreiben<sup>16</sup> (Hilber, 2015: S.

---

<sup>15</sup>Für Put-Optionen ist nur eine untere Grenze gesetzt, nach “oben” kann  $\eta$  unendlich hohe Werte annehmen.

<sup>16</sup>Es wurde schon einige Vorarbeit für die nicht-konstanten Koeffizienten geleistet, welche erst in 3.7 besprochen werden.

120ff):

$$\begin{aligned} \partial_t u(\eta_i, t) + a(\eta_i) \frac{u(\eta_{i-1}, t) - 2u(\eta_i, t) + u(\eta_{i+1}, t)}{h^2} \\ + b(\eta_i) \frac{-u(\eta_{i-1}, t) + u(\eta_{i+1}, t)}{2h} + c(\eta_i) u(\eta_i, t) \approx 0 \end{aligned} \quad (3.17)$$

Der Einfachheit und Übersichtlichkeit halber sowie aus Gründen der Konsistenz mit Hilber (2015: S. 109) und dem Octave Skript<sup>17</sup> wurden die Koeffizienten

$$a(\eta_i) := -\frac{1}{2} \sigma^2 \eta^2, \quad b(\eta_i) := -(q - r)\eta, \quad c(\eta_i) := q$$

gesetzt<sup>18</sup>. An dieser Stelle lässt sich nun mit Hilfe von  $u_i(t) \approx u(\eta_i, t)$  in der Approximation (3.17) Gleichheit erzwingen:

$$\begin{aligned} \partial_t u_i(t) + a(\eta_i) \frac{u_{i-1}(t) - 2u_i(t) + u_{i+1}(t)}{h^2} \\ + b(\eta_i) \frac{-u_{i-1}(t) + u_{i+1}(t)}{2h} + c(\eta_i) u_i(t) = 0 \end{aligned} \quad (3.18)$$

Gleichung (3.18) wird nun nach  $u_i$  geordnet [mit Vereinfachung  $\partial_t u_i(t) = u_i'(t)$ ], was zu

$$u_i'(t) + \alpha_i u_{i-1}(t) + \beta_i u_i(t) + \gamma_i u_{i+1}(t) = 0 \quad (3.19)$$

führt, wobei

$$\alpha_i := \frac{a(\eta_i)}{h^2} - \frac{b(\eta_i)}{2h}, \quad \beta_i := -\frac{2a(\eta_i)}{h^2} + c, \quad \gamma_i := \frac{a(\eta_i)}{h^2} + \frac{b(\eta_i)}{2h}$$

gesetzt werden.

Wird nun an dieser Stelle vereinfacht angenommen, dass die Randbedingungen  $u_0(t) = u(\eta_0, t) = u(\eta_l, t) = 0$  und  $u_{N+1}(t) = u(\eta_{N+1}, t) = u(\eta_r, t) = 0$ <sup>19</sup>, so lassen sich die  $N$  Gleichungen in (3.19) in Abhängigkeit von  $t$  als

$$\begin{aligned} u_1'(t) + \beta_1 u_1(t) + \gamma_1 u_2(t) &= 0, \\ u_i'(t) + \alpha_i u_{i-1}(t) + \beta_i u_i(t) + \gamma_i u_{i+1}(t) &= 0, \\ u_N'(t) + \alpha_N u_{N-1}(t) + \beta_N u_N(t) &= 0, \end{aligned} \quad (3.20)$$

schreiben. Dabei läuft der Index  $i$  in der zweiten Gleichung von 2 bis  $N - 1$ . Alternativ

<sup>17</sup>Welches in Kapitel 4 weiter diskutiert wird.

<sup>18</sup>Es ist zu beachten, dass – strikte genommen – nur  $a$  und  $b$  von  $\eta$  abhängen.

<sup>19</sup>Eine Annahme, welche in 3.5 wieder revidiert wird, es aber vorerst ermöglicht auf den generellen Fall einzugehen, bevor die “Sonderheiten” der Lookback-Option weiter diskutiert werden.

dazu, lässt sich dies in Matrix-Schreibweise ausdrücken als

$$u'(t) + Au(t) = 0 \quad (3.21)$$

wobei

$$u(t) := \begin{pmatrix} u_1(t) \\ u_2(t) \\ \vdots \\ u_N(t) \end{pmatrix}, \quad u'(t) := \begin{pmatrix} u'_1(t) \\ u'_2(t) \\ \vdots \\ u'_N(t) \end{pmatrix}$$

und

$$A := \begin{pmatrix} \beta_1 & \gamma_1 & & & & \\ \alpha_2 & \beta_2 & \gamma_2 & & & \\ & \alpha_3 & \beta_3 & \gamma_3 & & \\ & & & \ddots & & \\ & & & & \alpha_{N-1} & \beta_{N-1} & \gamma_{N-1} \\ & & & & & \alpha_N & \beta_N \end{pmatrix}. \quad (3.22)$$

Es ist zu beachten, dass zur eindeutigen Lösbarkeit des Problems die Anfangsbedingung  $u(0)$  bekannt sein muss, welche im vorliegenden Fall durch die Matrix  $g$  gegeben ist. Diese entspricht der Auszahlungsfunktion an den jeweiligen Gitterpunkten ausgewertet (Hilber, 2015: S. 119ff), welche es nun in der Zeit zurückzurechnen gilt. Somit ist

$$u(0) = g, \quad g := \begin{pmatrix} g(\eta_1) \\ g(\eta_2) \\ \vdots \\ g(\eta_N) \end{pmatrix}. \quad (3.23)$$

Hilber (2015: S. 252ff) zeigt, dass (3.21) in Verbindung mit (3.23) gelöst werden kann, was in  $u(t) = e^{At}g$  resultiert: Eine Lösung, welche – wenn der skalare Fall (das heisst  $N = 1$ ) betrachtet wird – einfach nachvollzogen werden kann. Da sich jedoch  $e^{Matrix}$  nur sehr schwer<sup>20</sup> auswerten lässt (Hilber, 2015: S. 120f), wird wiederum auf die Finite Differenzen Methode zurückgegriffen. In Abschnitt 3.4 wird gezeigt, wie sich eine Approximation in diesem Sinne mit Hilfe des  $\theta$ -Verfahrens durchführen lässt.

<sup>20</sup>Besonders für grosse Matrizen.



### 3.4 Das $\theta$ -Verfahren

Um nun das Differenzialgleichungssystem, welches sich aus den Gleichungen (3.21) in Verbindung mit (3.23) ergibt, approximativ zu lösen, wird der skalare Fall (das heisst  $N = 1$ )

$$\begin{cases} u'(t) + Au(t) & = 0 \\ u(0) & = g \end{cases} \quad (3.24)$$

betrachtet, wobei<sup>21</sup>  $A := -\frac{2a(\eta_1)}{h^2} + c$  und  $g := g(\eta_1)$  (Hilber, 2015: S. 121). Wiederum wird ein äquidistantes Gitter gewählt, welches durch

$$0 = t_0 < t_1 < \dots < t_{M-1} < t_M = T$$

gegeben ist und aus  $M + 1$  Gitterpunkten  $t_j = jk$  im Intervall  $[0, T]$  besteht, wobei Zeitschritt respektive Maschenweite  $k := \frac{T}{M}$ .

Nun lässt sich  $u'(t)$  zu jedem Zeitpunkt  $t_j$  durch den Vorwärtsdifferenzenquotienten ersetzen, welcher schon im Zusammenhang mit (3.9) und (3.10) erwähnt wurde (Hilber, 2015: S. 122f).

$$\underbrace{\frac{u(t_j + k) - u(t_j)}{k}}_{\text{Vorwärtsdifferenzenquotient}} + Au(t_j) \approx 0, \quad j = 0, \dots, M - 1 \quad (3.25)$$

Wird nun mit  $u_j \approx u(t_j)$  Gleichheit erzwungen sowie nach  $u_j$  sortiert, resultiert

$$u_{j+1} = u_j + k(-A)u_j, \quad j = 0, \dots, M - 1. \quad (3.26)$$

(3.26) wird als *explizites Euler-Verfahren* bezeichnet und ermöglicht es mittels Addition des vertikalen Zuwachses den nächsten Wert in der Folge zu erhalten, wobei sich dieser aus dem horizontalen Zuwachs  $k$  multipliziert mit der Steigung  $-Au(t_j)$  ergibt. Es handelt sich dabei um eine geometrische Folge, in welcher sich ein  $u_{j+1}$  sowohl rekursiv wie auch direkt berechnen lässt. Wie jedoch Hilber (2015: S. 123f) ausführt, hat das *explizite Euler-Verfahren* die Eigenschaft der *Instabilität*, was zu einer divergierenden Approximation führt, wenn Zeitschritt  $k$  nicht genügend klein gewählt wird.

Eine solche *bedingte Stabilität* ist nicht erwünscht, lässt sich aber mit dem Rückwärtsdif-

<sup>21</sup>Der Index 1 wurde weggelassen.

ferenzenquotienten verhindern.

$$\underbrace{\frac{u(t_j) - u(t_j - k)}{k}}_{\text{Rückwärtsdifferenzenquotient}} + Au(t_j) \approx 0, \quad j = 1, \dots, M \quad (3.27)$$

Es lässt sich nun analog zu (3.26) Gleichheit erzwingen sowie die Gleichung umstellen; des Weiteren wird die Gleichung nun mit dem Index beginnend bei 0 geschrieben:

$$u_{j+1} = u_j + k(-A)u_{j+1}, \quad j = 0, \dots, M - 1 \quad (3.28)$$

(3.28) wird als *implizites Euler-Verfahren* bezeichnet und weist eine *unbedingte Stabilität* auf (Hilber, 2015: S. 126).

Es lassen sich nun die beiden Verfahren zum Zweck einer besseren Konvergenzrate mit Hilfe des Parameters  $\theta \in [0, 1]$  kombinieren: das sogenannte  *$\theta$ -Verfahren*.

$$\frac{u_{j+1} - u_j}{k} + \underbrace{(1 - \theta)Au_j}_{\text{vergleiche (3.25)}} + \underbrace{\theta Au_{j+1}}_{\text{vergleiche (3.27)}} = 0, \quad j = 0, \dots, M - 1 \quad (3.29)$$

Für  $\theta = 0$  resultiert wiederum das *explizite Euler-Verfahren* und für  $\theta = 1$  das *implizite Euler-Verfahren*. Somit ist das Verfahren für  $\theta > 0$  ein implizites, da der vertikale Zuwachs, welcher zur Bestimmung von  $u_{j+1}$  benötigt wird, von sich selbst abhängt<sup>22</sup>. Wird (3.29) nun nach  $u_{j+1}$  aufgelöst, resultiert

$$u_{j+1} = \frac{1 - k(1 - \theta)A}{1 + k\theta A} u_j, \quad j = 0, \dots, M - 1. \quad (3.30)$$

Hilber (2015: S. 127-132) zeigt, dass das  $\theta$ -Verfahren für  $\theta$ -s im Intervall  $[1/2, 1]$  unbedingt stabil ist, sowie für  $\theta = \frac{1}{2}$  eine quadratische Konvergenz aufweist. Das  $\theta$ -Verfahren für  $\theta \neq \frac{1}{2}$  sowie auch der *Vorwärts-*<sup>23</sup> und *Rückwärtsdifferenzenquotient* selbst konvergiert linear. Das  $\theta$ -Verfahren für  $\theta = \frac{1}{2}$  wird als *Crank-Nicolson-Verfahren* bezeichnet.

Nachdem das  $\theta$ -Verfahren besprochen wurde, soll dieses an dieser Stelle für den nicht-skalare Fall (das heisst  $N > 1$ ) erweitert werden. Es wurde aufgezeigt, wie sich das Gleichungssystem (3.6) diskretisieren lässt. Hierbei ist das Gleichungssystem anhand von *Finiten Differenzen* sowohl hinsichtlich  $\eta$  wie auch in der Zeit zu diskretisieren. Beide Male

<sup>22</sup>Der vertikale Zuwachs entspricht wiederum dem horizontalen Zuwachs  $k$ , multipliziert mit der Steigung [hier nunmehr  $-(1 - \theta)Au_j - \theta Au_{j+1}$ ].

<sup>23</sup>Wie bereits im Zusammenhang mit (3.9) erwähnt.

wurde dazu ein äquidistantes Gitter eingeführt. Es entsteht also das Produktgitter

$$\mathcal{G}_{\eta,t} := \mathcal{G}_\eta \times \mathcal{G}_t = \{(\eta_i, t_j) \mid i = 0, \dots, N+1, j = 0, \dots, M\} \quad (3.31)$$

im Produkteintervall  $[\eta_l, \eta_r] \times [0, T]$ . Dieses besteht aus  $(N+2)(M+1)$  Gitterpunkten  $(\eta_i, t_j)$ , wobei in  $(\eta_i, t_0)$ ,  $(\eta_l, t_j)$  und  $(\eta_r, t_j)$  die Funktionswerte bekannt sind (vergleiche hierzu Hilber, 2015: S.133).

Hilber (2015: S. 133-136) weist nach, dass Matrix (3.23), welche die Anfangsbedingungen  $(\eta_i, t_0)$  angibt, analog zum skalaren Fall mit  $N = 1$  in der Zeit zurückgerechnet werden kann, was im Gegensatz zu (3.30) nicht mehr in einem einzelnen Wert, sondern wie folgt in einer Matrix resultiert<sup>24</sup>.

$$u_{j+1} = (I + k\theta A)^{-1}(I - k(1 - \theta)A)u_j, \quad j = 0, \dots, M-1. \quad (3.32)$$

Es ist an dieser Stelle zu bemerken, dass der Index  $j$  für den nicht skalaren Fall nicht mehr wie für (3.24) weggelassen werden kann, und dass  $A$  nun wiederum die tridiagonale  $N \times N$ -Matrix (3.22) bezeichnet. Bei  $u_j$  handelt es sich um  $M+1$  Vektoren, welche je  $N$  lang sind, wobei Vektor  $u_{i,M}$  die näherungsweise berechneten  $\eta$  zum Zeitpunkt  $t_0$  beinhaltet und somit von primärem Belang ist. Welcher der  $N$  Werte in Vektor  $u_{i,M}$  interessant ist, hängt vom gesuchten  $\eta$  ab, welches zur Zeit der Emission 1 sein muss, da der Preis bei Emission dem Minimal-Preis entspricht und  $\eta$  das Verhältnis zwischen Preis und Minimum bezeichnet. Um den Wert der Option aus den gegebenen  $\eta$ -s zu errechnen, müssen diese mit dem aktuellen Basiswert  $S$  multipliziert werden.

Hilber (2015: S. 139-147) zeigt, dass das *volldiskrete Verfahren* für  $\theta$ -s im Intervall  $[1/2, 1]$  unbedingt und für jene im Intervall  $[0, 1/2]$  nur bedingt (in der 2-Norm) stabil ist. Des Weiteren wird erläutert, dass eine quadratische Konvergenz für alle  $\theta$  erhalten werden kann, wenn  $M$  und  $N$  entsprechend gewählt werden (darauf wird in Kapitel 4 nochmals – auf den Fall bezogen – eingegangen).

### 3.5 Randdaten

In den vorangehenden Abschnitten wurde hinsichtlich der Randdaten jeweils hierhin verwiesen und für Ausführungen soweit immer davon ausgegangen, dass die Approximation anhand von *homogenen Randdaten* durchgeführt werden soll. Homogene Randdaten wür-

<sup>24</sup>Hierbei ist zu beachten, dass  $I$  für die *Einheitsmatrix* steht und  $(\bullet)^{-1}$  gewählt wurde, da Matrixdivision nicht definiert ist (Hilber, 2015: S. 136).

de bedeuten, dass die Funktionswerte an den Randstellen  $\eta_l$  und  $\eta_r$  dem Wert 0 entsprechen, das heisst  $u(\eta_l, t) = u(\eta_r, t) = 0$ . Eine solche Annahme führt jedoch zu schlechten Approximationen nahe des Randes des Intervalls  $G$  (Hilber, 2015: S. 161).

Es ist zu beachten, dass es verschiedene Arten von Randbedingungen gibt. So handelt es sich im erwähnten Fall, in welchem der Funktionswert bekannt ist, um eine *Dirichlet*-Randbedingung. Ist die erste Ableitung am Rand von  $G$  bekannt, wird dies hingegen als *Neumann*-Randbedingung bezeichnet; Randbedingungen basierend auf Ableitungen zweiter Ordnung sind ebenfalls möglich.

Im vorliegenden Fall ist mit  $[0, 1]$  für die *Floating Strike Lookback-Call-Option*<sup>25</sup> das Intervall, in welchem sich  $\eta$  bewegen kann, bereits gegeben. Es macht deshalb Sinn,  $\eta_l = 0$  und  $\eta_r = 1$  zu setzen. Somit ist mit

$$\partial_\eta u(1, t) = 0 \quad (3.33)$$

für  $\eta_r = 1$  eine erste Neumann-Randbedingung gegeben. Für den linken Rand des Intervalls  $G$  wird hingegen mit sogenannten *intrinsic Randdaten* gearbeitet. Beide Konzepte sollen im Folgenden in mehr Detail erläutert werden:

In einem ersten Schritt wird die *Neumann*-Randbedingung am rechten Rand des Intervalls betrachtet. Zu diesem Zweck muss die letzte der  $N$  Differenzialgleichungen angeschaut werden, welche mit finiten Differenzen diskretisiert worden ist (Hilber, 2015: S. 163).

$$\begin{aligned} & \partial_t u_N(t) + a(\eta_N) \frac{u_{N-1}(t) - 2u_N(t) + u_{N+1}(t)}{h^2} \\ & + b(\eta_N) \frac{-u_{N-1}(t) + u_{N+1}(t)}{2h} + c(\eta_N) u_N(t) = 0 \end{aligned} \quad (3.34)$$

Es stellt sich das Problem, dass  $u_{N+1}(t)$  nicht bekannt ist. Dies lässt sich aber mittels der Neumann-Randbedingung umgehen. Hierzu muss die Ableitung in der Randbedingung wiederum diskretisiert werden, was sich jedoch nicht mehr mit dem zentralen Differenzenquotienten (3.10) machen lässt, da derselbe von  $u_{N+2}(t)$  abhängen würde: Ein Punkt ausserhalb von Intervall  $G$ .

Hilber (2015: S. 164) zeigt, wie sich  $f'(x)$  anhand von

$$f'(x) \approx \frac{\pm 3f(x) \mp 4f(x \mp h) \pm f(x \mp 2h)}{2h} \quad (3.35)$$

---

<sup>25</sup>Bei *Lookback-Put-Optionen* verhält es sich anders so, dass nur  $\eta_l$  gegeben ist;  $\eta_r$  lässt sich jedoch genügend gross wählen, dass die Gitterpunkte nahe  $\eta_r$  nicht mehr von Interesse sind.

approximieren lässt, womit

$$\begin{aligned}
 & \partial_\eta u(\eta_r, t) = u_r(t), \\
 \xrightarrow{(3.35)} & \frac{u(\eta_{N+1} - 2h, t) - 4u(\eta_{N+1} - h, t) + 3u(\eta_{N+1}, t)}{2h} \approx u_r(t), \quad (3.36) \\
 \xrightarrow{u(\eta_{N+1} - h, t) \approx u_N(t)} & \frac{u_{N-1}(t) - 4u_N(t) + 3u_{N+1}(t)}{2h} = u_r(t).
 \end{aligned}$$

Dies lässt sich nun nach  $u_{N+1}(t)$  auflösen, so dass

$$u_{N+1}(t) = -\frac{1}{3}u_{N-1}(t) + \frac{4}{3}u_N(t) + \frac{2}{3}hu_r(t). \quad (3.37)$$

in (3.34) eingesetzt werden kann, wobei die zusätzliche Unbekannte  $u_{N+1}(t)$  wegfällt.

Wird zudem nach  $u_i$  sortiert, resultiert

$$u'_N(t) + \alpha_N u_{N-1}(t) + \beta_N u_N(t) = -\gamma_N u_r(t) \quad (3.38)$$

mit

$$\begin{aligned}
 \alpha_N &:= \frac{2a(\eta_N)}{3h^2} - \frac{2b(\eta_N)}{3h}, \\
 \beta_N &:= -\frac{2a(\eta_N)}{3h^2} + \frac{2b(\eta_N)}{3h} + c(\eta_N), \\
 \gamma_N &:= \frac{2a(\eta_N)}{3h} + \frac{b(\eta_N)}{3}.
 \end{aligned}$$

Es ist zu beachten, dass  $u_r(t)$  im vorliegenden Fall durch 0 gegeben ist [vergleiche (3.33)], womit  $-\gamma_N u_r(t) = 0$ . Zudem werden die Differenzialgleichungen für die Funktionen  $u_i(t)$  für  $i = 1, \dots, N - 1$  nicht tangiert. Das Einzige, was somit zu tun bleibt, ist die Werte  $\alpha_N$  und  $\beta_N$  in der zuvor entwickelten Matrix (3.22) zu "überschreiben". Hilber (2015: S. 161ff) zeigt wie für Randbedingungen basierend auf Ableitungen zweiter Ordnung analog verfahren werden kann.

### 3.6 Intrinsische Randbedingung

Für den rechten Rand des Intervalls wurde eine Neumann-Randbedingung durch (3.33) gegeben. Dies ist jedoch für den linken Rand des Intervalls nicht der Fall, weshalb auf die eingangs erwähnten *intrinsischen Randbedingungen* zurückgegriffen wird.

Hilber (2015: S. 173-175) zeigt, dass die Funktionswerte am Rand von Intervall  $G$  nicht

unbedingt bekannt sein müssen, da diese mit den Differenzialgleichungen (*mit-*)diskretisiert werden können. Es werden im Wesentlichen die Funktionswerte  $u_0(t)$  ebenfalls approximiert, was bedeutet, dass die Differenzialgleichung nicht nur im Intervall  $]0, 1[$ , sondern  $[0, 1[$  gelöst wird; die linke Randbedingung dann aber nicht gesetzt werden muss.

Es stellt sich an dieser Stelle ein ähnliches Problem wie bei den Neumann-Randbedingungen: Die zentralen Differenzenquotienten, welche bis anhin für die näherungsweise Lösung der Differenzialgleichungen verwendet wurden, bedürfen für Funktion  $u_i(t)$  mit  $i = 0$  dem Gitterpunkt  $u_{-1}(t)$ . Dieser liegt jedoch ausserhalb des Intervalls  $G$ . Um diesem Problem Abhilfe zu schaffen, wird ein Differenzenquotient gewählt, welcher nicht von Gitterpunkt  $u_{-1}(t)$  abhängt. Hierfür wird zur Approximation der Ableitung erster Ordnung (3.35) und für die Ableitung zweiter Ordnung

$$f''(x) \approx \frac{2f(x) - 5f(x+h) + 4f(x+2h) - f(x+3h)}{h^2} \quad (3.39)$$

herangezogen. Werden nun die zentralen Differenzenquotienten in der Differenzialgleichung in (3.17) für  $i = 0$  durch (3.39) respektive (3.35) ersetzt, sowie mit  $u(\eta_0, t) \approx u_0(t)$  Gleichheit erzwungen, resultiert

$$\begin{aligned} u'_0(t) + a(\eta_0) \frac{2u_0(t) - 5u_1(t) + 4u_2(t) - u_3(t)}{h^2} + \\ b(\eta_0) \frac{-3u_0(t) + 4u_1(t) - u_2(t)}{2h} + c(\eta_0)u_0(t) = 0, \end{aligned} \quad (3.40)$$

welche nicht mehr von  $u_{-1}(t)$  abhängt. Dies wird nun wiederum nach  $u_i$  sortiert,

$$u'_0(t) + \beta_0 u_0(t) + \gamma_0 u_1(t) + \delta_0 u_2(t) + \varepsilon_0 u_3(t) = 0 \quad (3.41)$$

wobei

$$\begin{aligned} \beta_0 &:= \frac{2a(\eta_0)}{h^2} - \frac{3b(\eta_0)}{2h} + c(\eta_0), \\ \gamma_0 &:= -\frac{5a(\eta_0)}{h^2} + \frac{4b(\eta_0)}{2h}, \\ \delta_0 &:= \frac{4a(\eta_0)}{h^2} - \frac{b(\eta_0)}{2h}, \\ \varepsilon_0 &:= -\frac{a(\eta_0)}{h^2}. \end{aligned}$$

gesetzt wird. Im Gegensatz zu den erläuterten Neumann-Randbedingungen reicht es nun nicht mehr, einfach eine Zeile der tridiagonale  $N \times N$ -Matrix (3.22), welche wir bis anhin mit  $A$  bezeichnet haben, zu überschreiben. Vielmehr muss eine neue  $(N + 1) \times (N + 1)$ -Matrix (nun nicht mehr tridiagonal) eingeführt werden, welche mit

$$\tilde{A} := \begin{pmatrix} \beta_0 & \gamma_0 & \delta_0 & \varepsilon_0 & & & & & & & \\ & \alpha_1 & \beta_1 & \gamma_1 & & & & & & & \\ & & \alpha_2 & \beta_2 & \gamma_2 & & & & & & \\ & & & \alpha_3 & \beta_3 & \gamma_3 & & & & & \\ & & & & & \ddots & & & & & \\ & & & & & & \alpha_{N-1} & \beta_{N-1} & \gamma_{N-1} & & \\ & & & & & & & \alpha_N & \beta_N & & \end{pmatrix} \quad (3.42)$$

gegeben ist und Teil des Differentialgleichungssystems

$$u'(t) + \tilde{A}u(t) = 0 \quad (3.43)$$

für die nunmehr  $N + 1$  Funktionen  $u(t) = (u_0(t), \dots, u_N(t))^T \in \mathbb{R}^{(N+1)}$  ist. Die Maschenweite  $h$  ist nunmehr durch

$$h = \frac{|G|}{N + 1} \quad (3.44)$$

gegeben, wobei  $|G|$  die Länge von Intervall  $G$  angibt (Hilber, 2015: S. 166-168).

### 3.7 Variable Koeffizienten

Im Rahmen der letzten Seiten wurde erklärt, wie es von einem partiellen Differentialgleichungssystem zu einer approximativen Bewertung einer Floating Strike Lookback-Call-Option kommt. Dabei wurde ausführlich diskutiert, wie die Differenzialquotienten anhand der Finiten Differenzen näherungsweise berechnet werden können. Was bei diesen Ausführungen nur am Rande angeschaut wurde, sind die Koeffizienten, welche mit  $a$ ,  $b$  und  $c$  bezeichnet werden.

Im Zusammenhang mit (3.17) wurde erwähnt, dass die Koeffizienten im vorliegenden Fall nicht konstant sind. Das ist daher der Fall, weil mit

$$a(\eta_i) := -\frac{1}{2}\sigma^2\eta^2$$

und

$$b(\eta_i) := -(q - r)\eta$$

zwei der Koeffizienten von  $\eta_i$  selbst abhängen und somit für alle  $N + 1$   $i$ -en andere sind. Der Einheitlichkeit halber wurden alle Koeffizienten in Abhängigkeit von  $\eta_i$  ausgedrückt, obwohl dies effektiv nur für  $a$  und  $b$  der Fall ist. Da sowohl  $\alpha$ ,  $\beta$  wie auch  $\gamma$  von den Koeffizienten  $a$  und/oder  $b$  abhängen, sind wiederum alle drei keine Konstanten und werden deshalb auch in (3.19) mit Subskript versehen. Das heisst für jede Zeile der Matrix  $A$ , wie durch (3.22) gegeben, ist beispielsweise  $\alpha$  ein anderer Wert. Darüber hinaus verändern sich die Werte für jeden Zeitschritt [vergleiche hierzu das *Produktgitter* gegeben durch (3.31)].

Es scheint also logisch, dass sich der Sachverhalt vereinfachen würde, wenn nicht nur  $c$  konstant wäre, sondern auch gleich  $a$  und  $b$ . Hilber (2015: S. 156) beschreibt die Möglichkeit, die Differenzialgleichung zu transformieren und in Abhängigkeit von

$$x := \ln(\eta)$$

auszudrücken. Da somit die erste Ableitung von  $x$  mit  $\frac{1}{\eta}$  und die zweite mit  $\frac{1}{\eta^2}$  gegeben wäre, lässt sich anhand der *Kettenregel* nachvollziehen, dass die  $\eta$ -s in den Koeffizienten wegfallen würden.

Im Rahmen der vorliegenden Arbeit wird diese Herangehensweise jedoch nicht gewählt, da sich – wie im nächsten Kapitel 4 gezeigt wird – die variablen Koeffizienten einfach implementieren lassen.

### **3.8 Berechnung der Extrema – kontinuierlich versus diskret**

Bis anhin ist davon abgesehen worden, dass für die Bestimmung der Extrema verschiedene Vorgehensweisen in Frage kommen. Das partielle Differenzialgleichungssystem (3.2), welches durch Zhu et al. (2010: S. 190) bereitgestellt wird, geht von einer kontinuierlichen Berechnung des minimalen Basiswertes während der Laufdauer des Derivates (mit  $L$  für *low* bezeichnet) aus. In diesem Fall würde eine kontinuierliche Messung bedeuten, dass



der Basiswert jederzeit (also kontinuierlich) auf ein neues “low” geprüft werden muss.

Die kontinuierliche Stichprobennahme lässt sich anhand der bis anhin beschriebenen Methode einfach berechnen. Es besteht jedoch der Nachteil, dass es sich um eine Annahme theoretischer Natur handelt, da eine Stichprobennahme realistisch gesehen nur diskret erfolgen kann<sup>26</sup>. Aus diesem Grund wird in Kapitel 4 nicht nur die kontinuierliche Stichprobennahme angeschaut, sondern die Routine anhand von

$$\begin{cases} \partial_t W + \frac{1}{2} \sigma^2 \eta^2 \partial_{\eta\eta}^2 W + (q - r) \partial_{\eta} W - rW = 0, & \text{falls } T \geq t \neq t_i, i = 1, \dots, K, \\ W(\eta, t_i^-) = W(\max(1, \eta), t_i^+), & \text{falls } t = t_i, i = 1, \dots, K, \\ W(\eta, T) = \max(\eta - \lambda_p, 0), \end{cases} \quad (3.45)$$

erweitert<sup>27</sup> (wobei  $\eta \geq 0$ ; Zhu et al., 2010: S. 190)<sup>28</sup>. Der wesentliche Unterschied besteht darin, dass die Basiswerte bzw.  $\eta$ -Werte im Intervall  $G$  nicht einfach in der Zeit zurückgerechnet werden, ausgehend von einem Vektor, welcher durch die Auszahlungsfunktion gegeben ist. Vielmehr wird im Fall der diskreten Stichprobennahme die Approximation bis zum ersten Stichprobennahmepunkt durchgeführt. Die Resultate werden dann für die Anfangsbedingung des nächsten Durchgangs verwendet. Wird dieser Schritt mehrmals durchgeführt, lässt sich der Preis zum Zeitpunkt  $t_0$  diskret berechnen.

---

<sup>26</sup>Vergleiche hierzu Kwok (2008: S. 211).

<sup>27</sup> $t$  steht an dieser Stelle nun nicht für die *Restlaufzeit*, da die “Richtung der Zeit” noch nicht umgekehrt wurde.

<sup>28</sup>Es handelt sich hierbei um die Formel für eine *Floating Strike Lookback-Put-Option*.

## 4 Anwendung

Nachdem in Kapitel 3 die Finite Differenzen Methode vorgestellt sowie das Vorgehen beziehungsweise die Besonderheiten der Bewertung einer Floating Strike Lookback-Option genauer erläutert wurden, soll die entsprechende Routine im Rahmen dieses Abschnittes erstellt werden.

Zu diesem Zweck wird das Programm Octave herangezogen. Wie schon zuvor wird auch in diesem Kapitel die Implementierung nur entweder anhand einer Put oder einer Call-Option erläutert – für die jeweils andere sind lediglich minimale Anpassungen erforderlich.

In Kapitel 3 wurde erwähnt, dass der Preis einer Lookback-Option sowohl anhand von kontinuierlichen wie auch von diskreten Stichprobennahmen (für die Extrema) bestimmt werden kann. Im Folgenden wird vorerst die kontinuierliche Stichprobennahme implementiert, welche dann sogleich auf Konvergenz getestet wird. In einem zweiten Schritt wird der Code für die diskrete Stichprobennahme erweitert.

Es ist zudem zu erwähnen, dass für die Implementierung (sowohl für die diskrete, wie auch für die kontinuierliche) auf die Routinen `Matrixgenerator_1.m` und `parabol_dgl_BC.m` – welche durch Hilber (2015: S. 169-171) zur Verfügung gestellt werden – zurückgegriffen wird.

### 4.1 Kontinuierliche Stichprobennahme

Es wird nun also die Routine `LBO_float.m` zur Bewertung einer Europäischen Floating Strike Lookback-Call-Option entwickelt oder wie im Kommentar der Routine:

```
% w = LBO_float(s,sigma,r,q,T,tj,N,M,lambda) gibt den  
% Wert einer europaeischen lookback strike call option  
% mit Auszahlungsfunktion: g=max(0,lambda-eta)
```

Die Routine erfordert die folgenden Inputparameter, welche es zu setzen gilt:

```
% Input:  
% r=Zinssatz  
% T=Maturitaet  
% q=Dividende
```

```
% sigma=Volatilitaet
% s=Basiswert
% lambda=Multiplikator
```

Das % gibt an, dass es sich bei den Ausführungen nur um Kommentare zum eigentlichen Code handelt. Die Bezeichnungen werden Analog zu den Formeln in Kapitel 3 gewählt, wobei  $\lambda := \lambda$ ,  $\eta := \eta$  und  $\sigma := \sigma$ .

```
function w = LBO_float(s,sigma,r,q,T,N,M,lambda,theta)
```

definiert die “Funktion” in Abhängigkeit von den genannten Parametern, wobei zusätzlich die Modellparameter  $M$ ,  $N$  und  $\theta$  gewählt werden müssen.

Es gilt, die Koeffizienten, welche im Zusammenhang mit (3.17) gegeben wurden, mit

```
a=@(eta)-0.5.*sigma.^2.*eta.^2;
b=@(eta)-(q-r)*eta;
c=@(eta)q*eta.^0;
```

für die Routine zu definieren. Dabei ist zu beachten, dass  $\text{@}(\eta)$  die Abhängigkeit von der Variablen  $\eta$  ausdrückt, das heisst angibt, dass die Koeffizienten nicht konstant sind. In einem nächsten Schritt wird mit

```
g=@(eta)max(0,lambda-eta);
```

die Anfangsbedingung mit der Auszahlungsfunktion gemäss des partiellen Differenzialgleichungssystems (3.8) gleichgesetzt (wiederum in Abhängigkeit von  $\eta$ ) sowie mit

```
wl=@(t)0*t;
wr=@(t)0*t;
```

die “Funktionswerte an den Rändern”<sup>1</sup> mit 0 angegeben. In diesem Fall wird mit  $\text{@}(t)$  die Abhängigkeit von  $t$  ausgedrückt, wobei die Randbedingungen im vorliegenden Fall für alle  $t$  mit 0 gegeben sind. Aus diesem Grund wird die Funktion aus technischen Gründen  $0*t=0$  gesetzt. Zudem werden mit

```
etal=0;
etar=1;
```

$\eta_l$  und  $\eta_r$  festgesetzt, wobei bis anhin noch nicht angegeben wurde, dass es sich bei den

---

<sup>1</sup>Im vorliegenden Fall handelt es sich am rechten Rand um die Ableitung und nicht den Funktionswert.

Randbedingungen um Neumann- beziehungsweise intrinsische Randbedingungen handelt – dies wird jedoch in

```
[eta,u] = parabol_dgl_BC(a,b,c,T,etal,wl,3,etar,wr,1,g,N,M,theta);
```

nachgeholt, wenn die definierten Parameter an die Routine `parabol_dgl_BC.m` weitergegeben werden. So steht die 3 für intrinsische Randdaten am linken Rand des Intervalls  $G$  sowie 1 für Neumann-Randbedingungen am rechten Rand.

```
function [s,w] = parabol_dgl_BC(a,b,c,T,sl,wl,nl,sr,wr,nr,g,N,M,
    theta)
2 % Gitter definieren
h = (sr-sl)/(N+1); s =(sl:h:sr)'; k = T/M;
4 if nl < 3; s(1) = []; end, if nr < 3; s(end)= []; end
% Matrizen A und I
6 [M2,M1,M0,M2bc,M1bc] = matrixgenerator_1({'M2',a},{'M1',b},{'M0',c
    }},[nl nr],sl,sr,N);
A = M2+M1+M0; I = speye(length(A));
8 % Hilfsmatrizen definieren
B = I + k*theta*A; C = I - (1-theta)*k*A;
10 % Start-Vektor w0 definieren
w = g(s);
12 % Loop ueber alle j, (der Vektor wj wird laufend ueberschrieben)
for j = 1:M
14     tj = (j-1+theta)*k; wbc = sparse(length(A),1);
    wbc(1) = wl(tj); wbc(end) = wr(tj);
16     f = -(M2bc+M1bc)*wbc; w = B\(C*w+k*f);
end
```

Algorithmus 4.1: Routine `parabol_dgl_BC.m` (Hilber, 2015: S. 171)

Die Routine 4.1 nimmt die übergebenen Parameter und berechnet die Maschenweite  $h$ , die durch (3.14) definiert wurde, wobei die vorkommenden  $s$ -en im vorliegenden Fall als  $\eta$ -s zu verstehen sind. In einem zweiten Schritt werden die Gitterpunkte im äquidistanten Gitter im Intervall  $G$  analog zu (3.15) in einem Vektor abgelegt. Es ist zu beachten, dass der Vektor die Randpunkte vorerst auch mitenthält. Mit if-Bedingungen wird danach geprüft, ob es sich bei einer der Randbedingungen um eine Bedingung intrinsischer Natur handelt. Sollte dies nicht der Fall sein, wird entsprechend der erste beziehungsweise letzte Wert im Vektor gelöscht: In casu wird also der Wert  $\eta_r = 1$  wieder aus dem Vektor gelöscht, da es sich am rechten Rand nicht um intrinsische Randdaten handelt.

An dieser Stelle wird mit den Parametern  $N$ ,  $\eta_l$  und  $\eta_r$  sowie den Angaben, um wel-

che Art Randbedingungen es sich handelt<sup>2</sup>, über die Routine `matrixgenerator_1.m` die Matrizen  $M2$ ,  $M1$  sowie  $M0$  bezogen. Zugleich wird vorgeschrieben, dass es sich bei der ersten Matrix um eine Matrix von ‘‘Typ’’  $M2$  für Koeffizient  $a$ , bei der zweiten um einen Typ  $M1$  für  $b$  sowie bei der dritten um eine von Typ  $M0$  für  $c^3$  handelt.

Die Matrizen sind durch Hilber (2015: S. 176) mit

$$M0 := \begin{pmatrix} y_0 & & & & \\ & y_1 & & & \\ & & \ddots & & \\ & & & y_N & \\ & & & & y_{N+1} \end{pmatrix}, \quad (4.1)$$

$$M1 := \frac{1}{2h} \begin{pmatrix} -3y_0 & 4y_0 & -y_0 & & \\ -y_1 & & y_1 & & \\ & \ddots & & \ddots & \\ & & -y_N & & y_N \\ & & y_{N+1} & -4y_{N+1} & 3y_{N+1} \end{pmatrix} \quad (4.2)$$

und

$$M2 := \frac{1}{h^2} \begin{pmatrix} 2y_0 & -5y_0 & 4y_0 & -y_0 & \\ y_1 & -2y_1 & y_1 & & \\ & \ddots & & \ddots & \\ & & y_N & -2y_N & y_N \\ -y_{N+1} & 4y_{N+1} & -5y_{N+1} & 2y_{N+1} & \end{pmatrix}, \quad (4.3)$$

definiert, wobei die Routine, sollten keine intrinsischen Randdaten vorliegen, wieder einen Teil der jeweiligen Matrix löscht: In casu werden, da es sich am rechten Rand nicht um intrinsische Randdaten handelt, die letzten Zeilen sowie letzten Spalten aller Matrizen ‘‘gelöscht’’, sowie die neuen letzten Zeilen entsprechend der Art der Randbedingung wiederum überschrieben. Da es sich am rechten Rand um Neumann-Randbedingungen handelt, werden in  $M2$  auf der letzten Zeile noch bestehende Einträge  $y_N$  und  $-2y_N$  mit  $\frac{2}{3}y_N$  respektive  $-\frac{2}{3}y_N$  überschrieben sowie in  $M1$   $-y_N$  und  $0$  mit  $-\frac{4}{3}y_N$  respektive  $\frac{4}{3}y_N$ .

Es gilt zu beachten, dass das  $y$  jeweils für die Koeffizienten  $a$ ,  $b$  oder  $c$  steht, für welchen

<sup>2</sup>In der Routine mit  $n_l$  und  $n_r$  bezeichnet, wobei es sich bei  $n_l = 3$  um intrinsische Randbedingungen und bei einem Wert von  $n_l = 1$  um Neumann-Randbedingungen am linken Rand handelt

<sup>3</sup>Vergleiche hierzu (3.17).

die Matrix generiert wurde<sup>4</sup> (letztere hängen teilweise von  $\eta$  ab). Werden die drei Matrizen addiert, entsteht die Matrix  $\tilde{A}$ , welche in (3.42) eingeführt wurde.  $M2bc$  und  $M1bc$  – ebenfalls durch den `matrixgenerator_1.m` generiert – werden nicht im Detail erklärt, da im vorliegenden Fall entweder mit intrinsischen Randdaten oder einem Wert von 0 “am Rand” gearbeitet wird und diese deshalb nicht von Belang sind.

Es soll nun wieder die Routine 4.1 betrachtet werden: Diese addiert die drei Matrizen zu  $\tilde{A}$  und generiert eine *Einheitsmatrix*  $I$  entsprechend der Länge von  $\tilde{A}$ . Danach wird  $B$  und  $C$  so definiert, dass

$$\stackrel{(3.32)}{\implies} u_{j+1} = B^{-1}Cu_j, \quad j = 0, \dots, M-1. \quad (4.4)$$

Die Auszahlungsfunktion, welche zuvor definiert wurde, wird sogleich mit  $u_0$ <sup>5</sup> gleichgesetzt. Zu guter Letzt wird mittels eines `for-end`-Loops für  $j = 1, \dots, M$  “in der Zeit zurückgerechnet”. Es gilt zu berücksichtigen, dass im Vergleich zu (4.4) der Index um 1 verschoben ist. Für dies wird bei der Definition von  $t_j$  mit  $-1$  kompensiert.

Wird in Betracht gezogen, dass  $f$  im vorliegenden Fall ein Vektor von Nullen ist, wird klar, dass der `for-end`-Loop für  $j = 1, \dots, M$  schlussendlich (4.4) implementiert.

Somit ist durch `LBO_float.m` lediglich noch mit

```
w=s.*interp1(eta,u,1,'pchip','extrap');
```

dasjenige  $\eta$  aus dem übergebenen Vektor  $u_M$  zu “suchen”, welches dem aktuellen  $\eta$  entspricht. Dieses muss, vorausgesetzt die Bewertung findet zur Zeit der Emission statt, 1 sein. Was – wie in Kapitel 3 bereits angesprochen – der Fall ist, da am Anfang der Laufzeit der *Basiswert* auch gleich dem *Minimum* entspricht und somit das Verhältnis  $\frac{L}{S} = 1$  ist. Anhand des Wertes 1 kann der entsprechende Wert  $u_{1,M}$  extrapoliert werden, welcher  $W(1, M)$  approximiert und mit dem aktuellen Basiswert  $S$  multipliziert den approximierten Wert der Lookback-Call-Option bei der Emission darstellt ( $V$ ).

**Beispiel:** Nachdem nun eine Routine zur Bewertung von Europäischen Floating Strike Lookback-Call-Optionen vorliegt, kann diese wie folgt zur Bewertung einer fiktiven Lookback-Option verwendet werden:

```
>> s=100; sigma=.25; r=.03; q=0; T=1; lambda=1; theta=.5; N=1000;
    M=10000;
>> w = LBO_float(s,sigma,r,q,T,N,M,lambda,theta)
```

<sup>4</sup>Das heisst,  $y$  steht für Koeffizient  $a$  in Matrix  $M2$ , für  $b$  in Matrix  $M1$  sowie  $c$  in Matrix  $M0$

<sup>5</sup>In Routine 4.1 als  $w_0$  bezeichnet.

w = 19.688

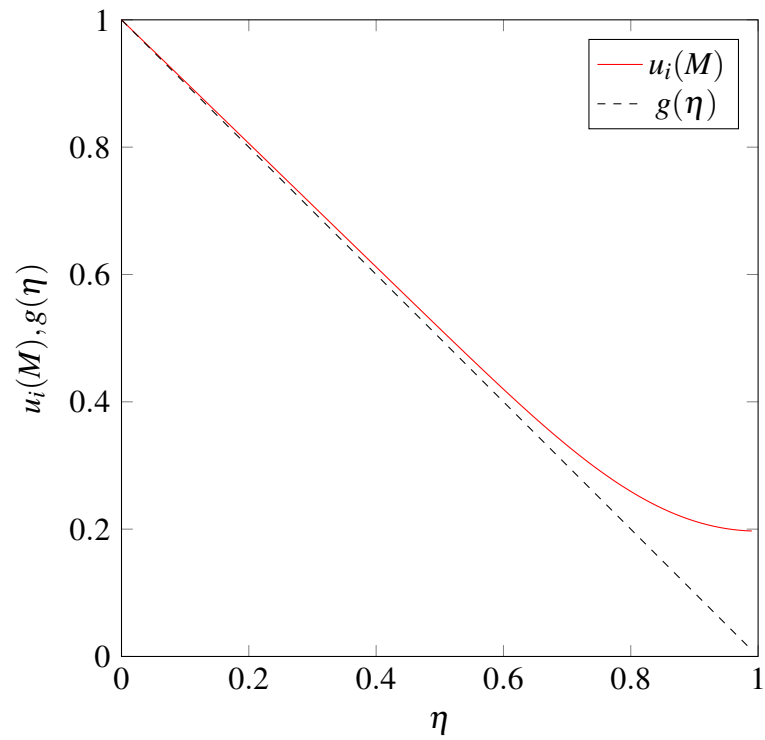


Abbildung 4.1:  $u$  sowie  $g$  in Abhängigkeit von  $\eta$ .

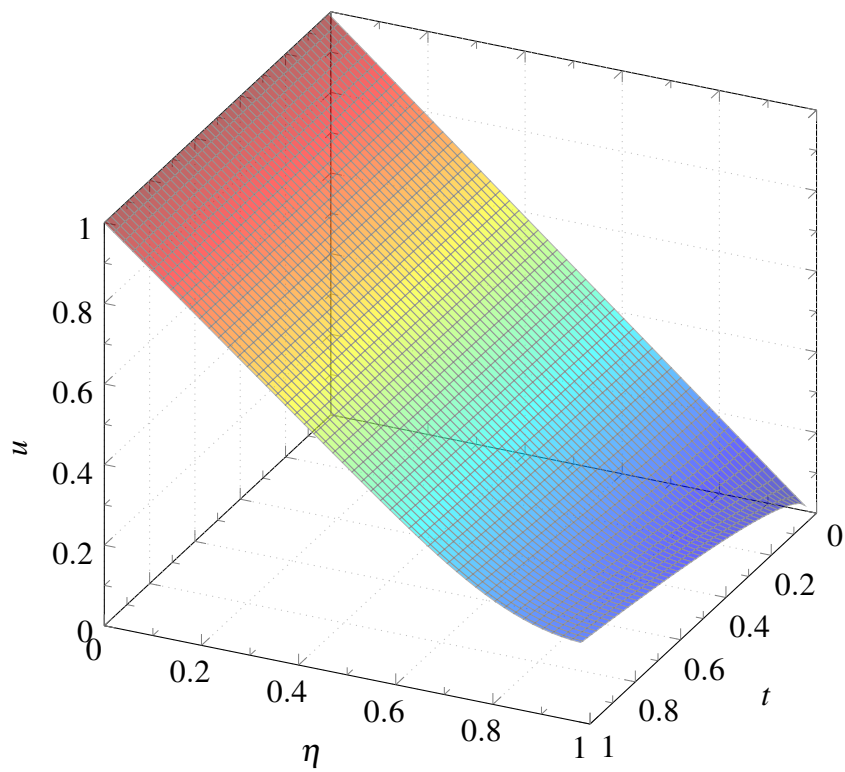


Abbildung 4.2:  $u$  in Abhängigkeit von  $\eta$  in der Zeit zurückgerechnet.

Abbildung 4.1 plotted hierzu die Werte aus dem Vektor  $u_M$  in Abhängigkeit von  $\eta$ , welche durch Routine 4.1 ausgegeben werden (—); des Weiteren wird die Auszahlungsfunktion  $g$  – wie in (3.16) gegeben – ebenfalls in Abhängigkeit von  $\eta$  geplotted (- - -). Abbildung 4.2 zeigt zudem die “Entwicklung über die Zeit”.

## 4.2 Konvergenzstudie

Folgend der Entwicklung der Routine über die letzten Seiten, scheint es wichtig, zu prüfen, ob die Resultate überhaupt sinnvoll sind. Da es sich bei der Finiten Differenzen Methode jedoch um eine näherungsweise Berechnung eines Wertes – in unserem Fall dem Preis einer Option – handelt, kann nicht erwartet werden, dass der “errechnete” Wert exakt ist; die Approximation muss aber konvergieren. Dies bedeutet im vorliegenden Fall, dass der Fehler<sup>6</sup>

$$\|e_j\| := \|V(t_j) - V_j\| \quad (4.5)$$

gegen Null streben muss, wenn  $M$  und  $N$  immer höhere Werte annehmen (Hilber, 2015: S. 45).

Bevor dies jedoch möglich ist, muss die Closed-Form-Gleichung (3.3) implementiert werden, um den genauen Wert der Lookback-Option bestimmen zu können, was mit Routine A.2 `LB0_float_BS.m` realisiert wurde.

Werden die gleichen Werte wie zuvor für Routine 4.1 verwendet sowie  $m=s$  gesetzt (was zur Zeit der Emmission der Realität entspricht), resultiert mit

```
>> s=100; sigma=.25; r=.03; T=1; m=s;
>> cfl = LB0_float_BS(s,sigma,r,T,m)
cfl = 19.688
```

der “gleiche” Preis wie schon zuvor. Dies ist jedoch nur für die ersten drei Dezimalstellen der Fall. Werden mittels `format long` zusätzliche Dezimalstellen ausgegeben, wird offensichtlich, dass die Werte nur näherungsweise gleich sind.

```
>> format long
>> cfl = LB0_float_BS(s,sigma,r,T,m)
cfl = 19.6879351990616
>> w = LB0_float(s,sigma,r,q,T,N,M,lambda,theta)
w = 19.6880580199321
>> e = abs(w-cfl)
```

<sup>6</sup>Auf Normen wird im Rahmen dieser Arbeit nicht weiter eingegangen, stattdessen wird in diesem Zusammenhang auf Hilber (2015: S. 140) verwiesen.



$e = 1.22820870526397e-04$

Es kann also gezeigt werden, dass der Wert, welcher mit der Finiten Differenzen Methode approximiert wurde, eine relativ genaue ‘‘Schätzung’’ des exakten Wertes<sup>7</sup> darstellt. Nun soll jedoch: (i) nachgewiesen werden, dass die Approximationen gegen den exakten Wert konvergieren, wenn  $M$  und  $N$  gegen unendlich streben, und (ii) die Konvergenzrate<sup>8</sup> bestimmt werden.

Hilber (2015: S. 147) zeigt, dass fur das  $\theta$ -Verfahren

$$\|e_j\| = \begin{cases} \mathcal{O}(h^2 + k), & \text{falls } \theta \in [0, 1] \setminus \{\frac{1}{2}\} \\ \mathcal{O}(h^2 + k^2), & \text{falls } \theta = \frac{1}{2} \end{cases} \quad (4.6)$$

gilt, was bedeutet, dass fur  $\theta = \frac{1}{2}$  von einer quadratischen Konvergenz gesprochen werden kann. Eine quadratische Konvergenz zeichnet sich dadurch aus, dass sich bei einer Verdoppelung von  $M$  und  $N$  der Fehler – im Gegensatz zur linearen Konvergenz – nicht nur halbiert, sondern nur noch ein Viertel so gross ist [da  $h = \mathcal{O}(N^{-1})$  und  $k = \mathcal{O}(M^{-1})$ ]. Ist ein Fehler also beispielsweise im einstelligen Bereich und einer im Basispunkte-Bereich soll erzielt werden, gilt es  $M$  und  $N$  hundertmal so gross zu wahlen (da  $100^2 = 10'000$ ). Es ist zu beachten, dass  $\mathcal{O}(\bullet)$  jeweils angibt, in welcher Weise (das heisst linear, quadratisch und so weiter) ein Fehler von einer gewissen Variablen abhangt.

Um dasselbe Verhalten bei  $\theta \neq \frac{1}{2}$  zu erzielen, fuhrt Hilber (2015: S. 147) weiter an, dass die Bedingung

$$M = \lceil \mu N^p \rceil \quad (4.7)$$

erfullt sein muss, wobei  $p \in \{1, 2\}$  und  $\mu > 0$  ( $p = 1$  fur  $\theta = \frac{1}{2}$  sowie  $p = 2$  wenn  $\theta \neq \frac{1}{2}$ ). Entspricht  $\theta$  also nicht 0.5, muss  $M$  jeweils vervierfacht werden, wenn  $N$  verdoppelt wird, um eine quadratische Konvergenz zu erzielen.

Es gilt nun die Routine Konvergenzpruefung.m vorzustellen. Da eine quadratische Konvergenz angestrebt wird, soll die Bedingung (4.7) gleich zu Beginn in die Routine Konvergenzpruefung.m (siehe Anhang A.3) aufgenommen beziehungsweise mittels einer if-else-Bedingung erzwungen werden. Zudem wird ein neuer (mit Nullen ‘‘gefullter’’) Vektor erstellt, in welchen spater die Fehler geschrieben werden. Mit sigma wird zudem

<sup>7</sup>Durch die Closed-Form-Losung generiert.

<sup>8</sup>Die Konvergenzrate sagt aus, wie schnell der Fehler  $\|e_j\|$  gegen 0 konvergiert, wenn  $M$  und  $N$  gegen  $\infty$  streben.

eine Anzahl an Volatilitäten gegeben, für welche die Konvergenzprüfung durchgeführt wird; die exakten Werte `wex` werden zugleich für alle Volatilitäten berechnet.

```
sigma=[0.1:.1:0.3];
wex = LBO_float_BS(s,sigma,r,T,s*1);
e = zeros(Lmax-Lmin+1,length(sigma));

if theta==.5, p=1; else p=2; end
```

Mit der “erzwungenen” Bedingung (4.7) wird nun erwartet, dass eine quadratische Konvergenz festgestellt werden kann, das heisst

$$\|e_j\| = \mathcal{O}(N^{-2}). \quad (4.8)$$

Hilber (2015: S. 147) bemerkt zudem, dass Konvergenz nicht in jeder Norm  $\|\bullet\|$  der Fall sein muss. Mit Konvergenzprüfung `.m` wird die Konvergenz in der *Maximumnorm* betrachtet, sprich

$$\|e_j\|_\infty = \mathcal{O}(N^{-n}). \quad (4.9)$$

$n$  steht hierbei für die Konvergenzordnung.

Es werden erneut mittels eines `for-end`-Loops für eine Anzahl von  $N$ -en die Preise der Lookback-Optionen entsprechend der verschiedenen Volatilitäten berechnet. Die  $N$ -en werden analog zu Hilber (2015: S. 149f) mit  $N = 2^L - 1$  in Abhängigkeit von Variable  $L$  gegeben. Es ist zu beachten, dass  $N \in \{2^{L_{min}} - 1, 2^{L_{min}+1} - 1, \dots, 2^{L_{max}} - 1\}$ , wobei  $L_{min}$  und  $L_{max}$  für die Routine gesetzt werden müssen.

```
for L = Lmin:Lmax

    N = 2^L-1; M = ceil(mu*N^p);

    w= zeros(1,length(sigma));
    for u=1:length(sigma)
        w(u) = LBO_float(s,sigma(u),r,q,T,N,M,lambda,theta);
```

Mit den erhaltenen Werten wird darauffolgend der Fehler in der *Maximumnorm* berechnet.

```
e(L-Lmin+1,u) = norm(w(u)-wex(u),'inf');
```

Anschliessend wird mittels `polyfit` eine Regression durchgeführt, um festzustellen in welcher Weise die logarithmierten  $e$ -en von den logarithmierten  $N$ -en abhängen: Es wird

die Steigung der Regressionskurve sowie eine grafische Darstellung derselben für jeden Wert aus  $\sigma$  ausgegeben.

```
n = zeros(length(sigma),1); N = 2.^(Lmin:Lmax)'.-1;
for j = 1:length(sigma)
    b = polyfit(log(N),log(e(:,j)),1); n(j) = b(1);
loglog(N,e(:,j),'k+',N,exp(b(2))*N.^n(j),'r-'), xlabel('N'), ylabel
('e_N'), hold on
end
```

Da eine Abhängigkeit des Fehlers von  $N$  in Form von

$$e = cN^{-n} \quad (4.10)$$

erwartet wird (es werden der Anschaulichkeit halber vereinfachte Notationen verwendet;  $c$  steht hierbei für eine beliebige Konstante), muss der Logarithmus verwendet werden, um mittels

$$\ln(e) = -n \ln(N) + c_l \quad (4.11)$$

die Konvergenzrate  $n$  zu erhalten, wobei  $c_l := \ln(c)$ .

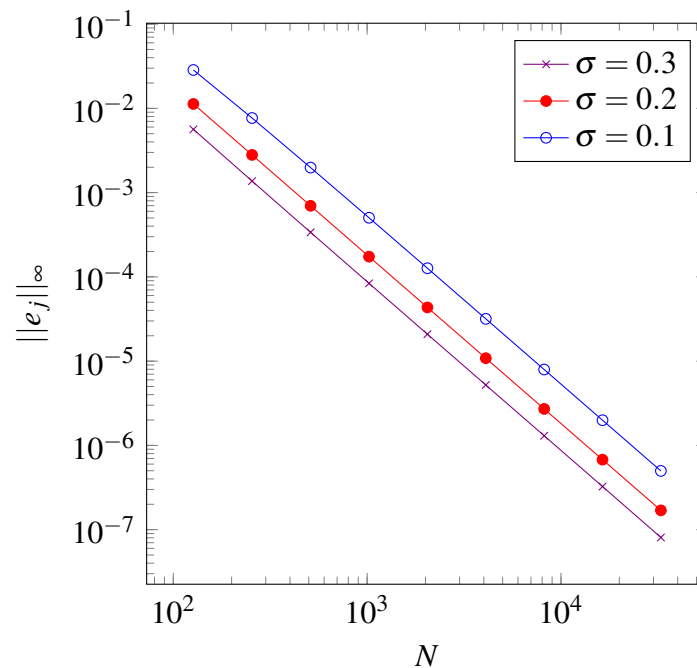
**Beispiel:** Es lässt sich nun anhand der schon zuvor gewählten Werte die Konvergenzrate der Routine `LBO_float.m` analysieren:

```
1 >> n = Konvergenzpruefung(100,.03,0,1,7,15,1,.5,.8)
n =
3 -1.97904548344186
-1.99446874371612
5 -2.00029157162604
-2.00409685324118
7 -2.00668811600383
-2.01199148546044
9 -2.33523200646917
```

Es zeigt sich, dass die Konvergenzrate in der Maximumnorm in der Tat in etwa zwei beträgt. In Tabelle 4.1 sowie Abbildung 4.3 wird zudem die Entwicklung der Fehler mit zunehmendem  $N$  dargestellt.

Tabelle 4.1: Veränderung des Fehlers mit zunehmendem  $N$ 

L	N	Maximumnorm ( $\sigma$ )		
		(0.1)	(0.2)	(0.3)
7	127	$2.85 \cdot 10^{-2}$	$1.13 \cdot 10^{-2}$	$5.63 \cdot 10^{-3}$
8	255	$7.68 \cdot 10^{-3}$	$2.8 \cdot 10^{-3}$	$1.37 \cdot 10^{-3}$
9	511	$1.98 \cdot 10^{-3}$	$6.97 \cdot 10^{-4}$	$3.38 \cdot 10^{-4}$
10	1,023	$5.03 \cdot 10^{-4}$	$1.74 \cdot 10^{-4}$	$8.38 \cdot 10^{-5}$
11	2,047	$1.27 \cdot 10^{-4}$	$4.34 \cdot 10^{-5}$	$2.09 \cdot 10^{-5}$
12	4,095	$3.18 \cdot 10^{-5}$	$1.08 \cdot 10^{-5}$	$5.21 \cdot 10^{-6}$
13	8,191	$7.95 \cdot 10^{-6}$	$2.71 \cdot 10^{-6}$	$1.3 \cdot 10^{-6}$
14	16,383	$1.99 \cdot 10^{-6}$	$6.77 \cdot 10^{-7}$	$3.25 \cdot 10^{-7}$
15	32,767	$4.98 \cdot 10^{-7}$	$1.69 \cdot 10^{-7}$	$8.07 \cdot 10^{-8}$

Abbildung 4.3:  $\|e_j\|_\infty$  in Abhängigkeit von  $N$ .

### 4.3 Put-Optionen

Bis anhin wurde die Bewertung von Europäischen Floating Strike Lookback-Optionen exemplarisch anhand einer Call-Option diskutiert. Dieser Absatz soll nun summarisch auf den Fall von Put-Optionen eingehen.

Im Wesentlichen gilt es lediglich: (i) die Auszahlungsfunktion anzupassen und (ii) die

Randbedingungen anders zu wählen.

- (i) Die Auszahlungsfunktion für Floating Strike Lookback-Put-Optionen ist durch Zhu et al. (2010: S. 192) mit

$$u(\eta, T) = \max(\eta - \lambda_p, 0), \quad 1 \leq \eta,$$

gegeben. Wobei es sich bei  $\lambda_p$  nicht mehr um das gleiche  $\lambda$  wie für die Call-Option handelt, da dieses  $\geq 1$ . Es ist jedoch nach wie vor möglich ein  $\lambda$  von 1 zu wählen, sollte kein Bedarf bestehen, Optionen mit einem niedrigeren Preis zu generieren<sup>9</sup>.

- (ii) Bezüglich der Randbedingungen ist zu beachten, dass zwar nach wie vor eine Neumann-Randbedingung vorliegt, diese sich jedoch, da  $\eta \in [1, \infty[$ , nicht mehr am rechten Rand, sondern am linken befindet. Für den rechten Rand lassen sich hingegen wiederum intrinsische Randdaten verwenden, wobei ein passendes  $\eta_r$  zu wählen ist.

Auf weitere Vorgänge, welche durch Hilber (2015: S. 169-171)'s `Matrixgenerator_1.m` und `parabol_dgl_BC.m` durchgeführt würden, soll im Rahmen dieser Arbeit nicht nochmals eingegangen werden.

## 4.4 Diskrete Stichprobennahme

An dieser Stelle soll nun die unter Punkt 3.8 angekündete diskrete Stichprobennahme angeschaut sowie Routine `LBO_float.m` zu Routine A.4 `LBO_float_diskret_put.m` erweitert werden.

Wie im Zusammenhang mit dem partiellen Differenzialgleichungssystem (3.45) erwähnt, ist im Wesentlichen die bestehende Routine dahingehend zu erweitern, dass der Preis jeweils nur bis zu einem gewissen "Sample-Punkt", durch die Routine mit `tj` bezeichnet, errechnet wird. Im Kommentar zur Routine wird ein Beispiel für solche `tj`-s gegeben.

```
%tj=[1:40]'/40*T;
```

Ausserdem ist zu beachten, dass `LBO_float_diskret_put.m` für die Bewertung einer *Put-Option* aufgesetzt wurde, was den Vorteil hat, dass die Resultate im Folgenden mit denen von Aitsahlia und Lai (1998: S. 238) verglichen werden können; entsprechend

---

<sup>9</sup>Vergleiche hierzu die Erläuterungen in Kapitel 3.

wurde die Auszahlungsfunktion für *Put-Optionen*<sup>10</sup> verwendet:

```
g=@(eta) max(0, eta - lambda);
```

In einem nächsten Schritt wird der Vektor  $t_j$  mit 0 ergänzt, was für die Berechnung der “Zeitspanne” zwischen den Sample-Punkten notwendig ist. Gleichzeitig werden die *Koeffizienten*  $a, b$ , und  $c$  gesetzt.

```
tjx=[0; tj];
```

```
a=@(eta) -0.5*sigma^2*eta.^2;
```

```
b=@(eta) -(q-r)*eta;
```

```
c=@(eta) q*eta.^0;
```

Da sich  $\eta$  nicht mehr wie im Fall der kontinuierlich berechneten *Call-Option* im Intervall  $[0, 1]$  bewegt, sondern nur mit 0 nach unten beschränkt ist, sind die Randbedingungen anders zu setzen: Für den linken Rand bietet es sich hierbei an, in  $\eta_l = 0$  eine intrinsische Randbedingung zu verwenden.  $\eta_r$  hingegen lässt sich beliebig wählen, wobei es sinnvoll ist,  $2^L$  mit  $L$  beispielsweise 2 oder 3 zu wählen. Dies wird dazu führen, dass, wenn wie im Abschnitt 4.2  $N = 2^L - 1$  (beispielsweise mit  $L = 10$ )<sup>11</sup>, einer der Punkte  $\eta_i$  dem Wert 1 entspricht. Das wird sich später als nützlich erweisen. An dieser Stelle  $\eta_r$  wird ein Wert von 0 für die *zweite* Ableitung verwendet.

```
wl=@(t) 0*t;
```

```
wr=@(t) 0*t;
```

```
etal=0;
```

```
etar=4;
```

Nachdem die Anfangs- und Randbedingungen für die Bewertung unter Annahme der diskreten Stichprobennahme gesetzt worden sind, gilt es die Routine entsprechend zu erweitern. Im Gegensatz zum kontinuierlichen Verfahren ist mit einer einmaligen Berechnung, wie durch `LBO_float.m` durchgeführt, den Anforderungen nun jedoch nicht mehr Genüge getan. Vielmehr gilt es diesen Vorgang  $t_j$ -mal zu wiederholen: Dies wird wiederum mittels eines `for-end`-Loops bewerkstelligt. In jedem Durchgang wird jeweils für die Zeitspanne zwischen zwei aufeinanderfolgenden “Sampling-Punkten” in der “Zeit zurückgerechnet”. Hierfür wird `revnum` definiert, welche es ermöglichen wird, die Werte in Vektor  $t_jx$  in “umgekehrter Reihenfolge” abzurufen.

```
[eta, u] = parabol_dgl_BC(a, b, c, tjx(length(tjx)) - tjx(length(tjx) - 1),  
    etal, wl, 3, etar, wr, 2, g, N, M, theta);
```

<sup>10</sup>Vergleiche 4.3(i).

<sup>11</sup>Wird  $\eta_r = 2^2$  gewählt, sollte für den gewünschten Effekt  $L > 2$  für  $N$  gewählt werden.

```

for j = 2:length(tj)
    idx=find(eta==1);
    u(1:idx)=u(idx);

    g=@(eta)u;
    revnum=length(tj)+2-j;

    [eta,u] = parabol_dgl_BC(a,b,c,tjx(revnum)-tjx(revnum-1),etal,wl,3,
        etar,wr,2,g,N,M,theta);
end

```

Nach jedem Durchgang ist anhand des erhaltenen Vektors  $u$  die neue *Anfangsbedingung* für den nächsten Durchgang zu definieren: Hierfür wird, wie durch Gleichungssystem (3.45) beschrieben, der erhaltene Vektor  $u$  verwendet. Mittels `idx` wird derjenige *Index*  $i$  gefunden, welcher dem Gitterpunkt  $\eta_i = 1$  entspricht<sup>12</sup>. Diejenigen  $u_i$ -s, welche aus  $\eta$ -s  $\leq 1$  resultieren, werden dann mit demjenigen für  $\eta = 1$  ersetzt. Mit dem generierten  $u$  wird die bestehende Anfangsbedingung  $g$  nun jeweils überschrieben.

Nach `tj` Durchlaufen wird dann wiederum der “passende”  $u_{i,M}$  mit  $S$  multipliziert und somit der approximierte Optionspreis berechnet.

```
w=s.*interp1(eta,u,1,'pchip','extrap');
```

## 4.5 Fehler der Approximation

Mit der entwickelten Routine lassen sich erneut approximative Bewertungen vornehmen. Es gilt jedoch vorerst zu prüfen, ob die generierten Werte eine gute Approximation darstellen. Für die Berechnung des exakten Preises unter Annahme von diskretem Sampling existiert zwar wiederum eine Closed-End-Lösung, diese bedingt jedoch die Verwendung der  $n$ -variater Normalverteilung<sup>13</sup>. Es wird auf Resultate aus der Literatur zurückgegriffen.

**Beispiel:** Wird der Wert einer Option  $V_M \approx V(t_M)$  aufgrund von den Input-Parametern  $s=100$ ,  $\sigma=0.3$ ,  $r=0.1$ ,  $q=0$  und  $T=0.5$  berechnet sowie eine Anzahl von 40 Sampling-Zeitpunkten verwendet, so lässt sich dies mit dem Wert  $V_{lit} \doteq 13.2394$  aus Aitsahlia und Lai (1998: S. 238, Tabelle 3) vergleichen.

<sup>12</sup>Dass 1 ein Gitterpunkt ist, wurde durch die Wahl der Randstellen und  $N$  sichergestellt.

<sup>13</sup> $n$  steht für die Anzahl *Sampling-Zeitpunkte* (vergleiche hierzu Kwok, 2008: S. 211)

## 4 Anwendung

---

```
>> s=100; sigma=.3; r=.1; q=0; T=0.5; lambda=1; theta=.5; L = 15;
    N=2^L-1; M=ceil(.05*N);
>> tj=[1:40]'/40*T;
>> w = LBO_float_diskret_put(s,sigma,r,q,T,tj,N,M,lambda,theta)
w = 13.2394140006233
```

Mit  $M = 32'767$  und  $N = 1'639$  stimmt der näherungsweise berechnete Wert schon bereits bis auf die ersten vier Nachkommastellen mit dem der Literatur entnommenen Wert überein. Tabelle 4.2 zeigt, wie sich der Fehler verändert, wenn  $N = 2^L - 1$  und  $M$ , wie in (4.7) beschrieben, gewählt werden. Es ist ersichtlich, dass der Fehler sich in etwa viertelt, wenn  $N$  und  $M$  verdoppelt werden. Eine detaillierte Konvergenzstudie ist hierbei schwierig, da die Werte von Aitsahlia und Lai (1998: S. 238) selbst gerundet und somit auch nur “bedingt exakt” sind. Abbildung 4.4<sup>14</sup> illustriert zudem, wie sich die Vektoren  $u_j$ , welche jeweils an den “Sampling-Zeitpunkten” ausgegeben werden, über die Zeit entwickeln.

Tabelle 4.2: Veränderung des Fehlers mit zunehmendem  $N$  und  $M$

L	N	M	$V_j$	$V_{lit}$	$e_j$
7	127	7	12.8136956	13.2394	0.4257
8	255	13	13.1437097	13.2394	0.0957
9	511	26	13.2160057	13.2394	0.0234
10	1,023	52	13.2335941	13.2394	$5.8059 \cdot 10^{-3}$
11	2,047	103	13.2379651	13.2394	$1.4349 \cdot 10^{-3}$
12	4,095	205	13.2390562	13.2394	$3.4384 \cdot 10^{-4}$
13	8,191	410	13.2393288	13.2394	$7.1194 \cdot 10^{-5}$

---

<sup>14</sup>Es ist zu beachten, dass die Achsen im Vergleich zu Abbildung 4.2 aus illustrativen Gründen umgekehrt worden sind.



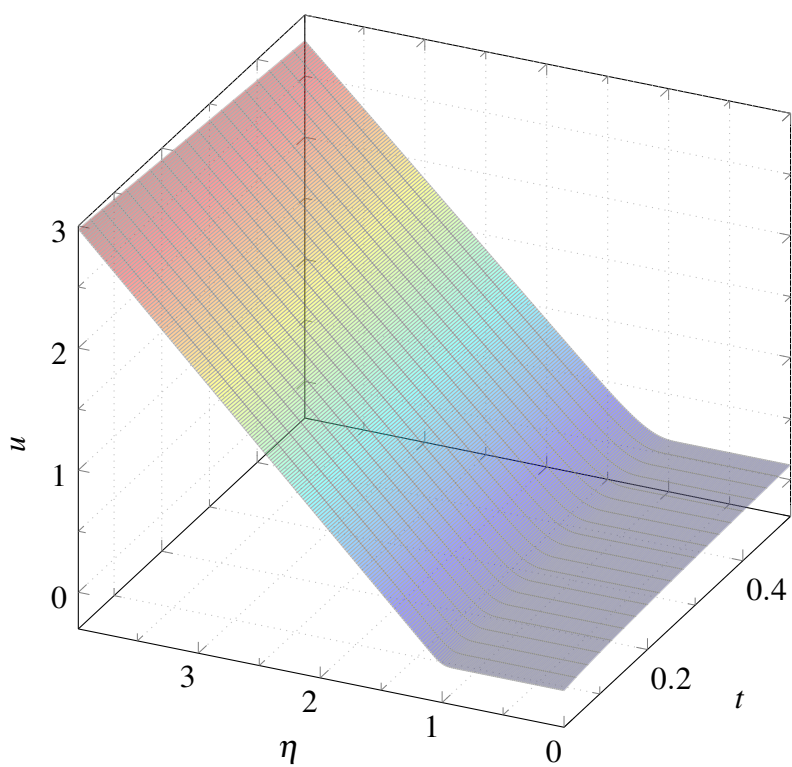


Abbildung 4.4:  $u_j$  an den jeweiligen *Sampling-Zeitpunkten*.

## 5 Schlusswort

Die vorliegende Bachelorarbeit befasst sich mit den *Floating Strike Lookback-Optionen*: eine exotische und pfadabhängige Optionsart. Es handelt sich um Optionen, bei denen die Auszahlungsfunktion nicht nur vom Basiswert bei der Ausübung abhängt, sondern gleichwohl von einem Extremwert, welcher sich während einer sogenannten *Lookback-Periode* ergibt. Bei der Floating Strike Lookback-Put-(Call-)Option tritt hierbei das Maximum (Minimum) an Stelle des *Strike-Prices* einer Europäischen Plain Vanilla Call (Put) Option.

Zur Bewertung der *Floating Strike Lookback-Optionen* wurde mittels *Octave* eine Routine erarbeitet, welche die Bewertung basierend auf der Finiten Differenzen Methode approximativ vornimmt. Dabei wurde von einem partiellen Differenzialgleichungssystem ausgegangen, welches durch Einführung einer Hilfsvariablen auf ein System abhängig von zwei Variablen vereinfacht werden konnte. Dieses wurde hinsichtlich der Hilfsvariablen und der Zeit diskretisiert und ermöglicht somit die näherungsweise Berechnung des Preises zum Zeitpunkt der Emission.

Da sich die erwähnten Extrema sowohl unter der (theoretischen) Annahme von kontinuierlicher Stichprobennahme wie auch unter der Annahme der diskreten Stichprobennahme berechnen lassen, wurden entsprechend zwei verschiedene Routinen ausgearbeitet. Die Resultate der ersteren liessen sich durch eine Closed-Form-Lösung verifizieren; die Resultate von letzteren konnten dahingegen aufgrund von Ergebnissen aus der Literatur geprüft werden.

Im Rahmen der Konvergenzstudie konnte gezeigt werden, dass die Konvergenzrate des implementierten Verfahrens in etwa zwei beträgt, wenn die Modellparameter entsprechend gesetzt werden.

Für weitere Arbeiten im Bereich der behandelten Thematik ist zu erwähnen, dass sich die vorliegende Arbeit auf eine spezifische Art von Lookback-Optionen beschränkt und nur eine mögliche Bewertungsmethode abdeckt. So sollten künftige Arbeiten auf die Bewertung von Lookback-Optionen von anderer Ausschaffung eingehen, beziehungsweise dieselbe unter Anwendung anderer Methodiken vornehmen.

Zudem wird darauf hingewiesen, dass sich die behandelten Bewertungsmodelle auch erweitern liessen: Es wäre beispielsweise möglich, Lookback-Optionen nicht nur zum Zeitpunkt der Emission zu bewerten, sondern auch zu anderen Zeitpunkten der Laufzeit. Zudem könnten Routinen, welche den Fall einer beschränkten Lookback-Periode abdecken, entwickelt werden.

## Literaturverzeichnis

- Aitsahlia, F. & Lai, T. L. (1998). Random walk duality and the valuation of discrete look-back options. In *Applied Mathematical Finance. Volume 5. S. 227–240*. Abgerufen von <http://www.tandfonline.com/doi/abs/10.1080/135048698334655>
- Hilber, N. (2015). *Vorlesungsskript Bewertung von Finanzderivaten von Dr. Norbert Hilber. Herbstsemester 2015/16*. Winterthur: Zürcher Hochschule für Angewandte Wissenschaften, School of Management and Law.
- Kwok, Y.-K. (2008). *Mathematical models of financial derivatives. 2. Auflage*. Berlin Heidelberg: Springer.
- MacKenzie, D. (2006). *An engine, not a camera: How financial models shape markets*. Cambridge, MA: MIT Press.
- Plain Vanilla. (2000). In *The American Heritage ® Dictionary of the English Language (4. Auflage)*. Abgerufen von <http://web.archive.org/web/20010224001515/http://www.bartleby.com/61/68/P0346850.html>
- Poitras, G. (2009). The early history of option contracts. In *Vinzenz Bronzin's Option Pricing Models. 1. Auflage. S. 487–518*. Berlin Heidelberg: Springer.
- UBS AG. (2014). *Lookback PERLES Plus*. Abgerufen von <http://keyinvest-ch-en.ubs.com/product/detail/documents/isin/CH0233395752>
- Zhu, Y., Wu, X., Chern, I.-L. & Sun, Z.-z. (2010). *Derivative securities and difference methods. 2. Auflage*. Berlin Heidelberg: Springer.

## A Anhang

### A.1 Code LBO\_float.m

```
2 % w = LBO_float(s,sigma,r,q,T,tj,N,M,lambda) gibt den
% Wert einer europaeischen lookback strike call option
% mit Auszahlungsfunktion: g=max(0,lambda-eta)
4 %
% Input:
6 % r=Zinssatz
% T=Maturitaet
8 % q=Dividende
% sigma=Volatilitaet
10 % s=Basiswert
% lambda=Multiplikator
12
function w = LBO_float(s,sigma,r,q,T,N,M,lambda,theta)
14
a=@(eta)-0.5.*sigma.^2.*eta.^2;
16 b=@(eta)-(q-r)*eta;
c=@(eta)q*eta.^0;
18 g=@(eta)max(0,lambda-eta);
wl=@(t)0*t;
20 wr=@(t)0*t;
22 etal=0;
etar=1;
24
[eta,u] = parabol_dgl_BC(a,b,c,T,etal,wl,3,etar,wr,1,g,N,M,theta);
26
w=s.*interp1(eta,u,1,'pchip','extrap');
```

Algorithmus A.1: Routine LBO\_float.m

## A.2 Code LBO\_float\_BS.m

```
1 % cfl = LBO_float_BS(s,sigma,r,T,m) gibt den
% Wert einer europaeischen lookback strike call option
3 % mit q=0 basierend auf kontinuierlichem Sampling.
%
5 % s=Basiswert
% sigma=Volatilität√t
7 % r=Riskfree Rate
% T=Laufzeit
9 % m=LTD Minimum

11 function cfl = LBO_float_BS(s,sigma,r,T,m)

13 dm=(log(s./m)+(r+sigma.^2./2).*T)/(sigma.*T.^5);

15
cfl=s.*normcdf(dm)-e.^(-r.*T).*m.*normcdf(dm-(sigma.*T.^5))+e.^(-r
    .*T).*(sigma.^2)/(2.*r).*s.*...
17 ((s./m).^((-2.*r)/(sigma.^2)).*normcdf(-dm+(2.*r)/sigma.*T.^5)-e
    .^(r.*T).*normcdf(-dm));
```

Algorithmus A.2: Routine LBO\_float\_BS.m

### A.3 Code Konvergenzpruefung.m

```
1 % n = Konvergenzpruefung(100,.03,0,1,7,10,1,.5,.8)
function n = Konvergenzpruefung(s,r,q,T,Lmin,Lmax,lambda,theta,mu)
3 sigma=[0.1:.1:0.3];
wex = LBO_float_BS(s,sigma,r,T,s*1);
5 e = zeros(Lmax-Lmin+1,length(sigma));

7 if theta==.5, p=1; else p=2; end

9 for L = Lmin:Lmax

11     N = 2^L-1; M = ceil(mu*N^p);

13     w= zeros(1,length(sigma));
    for u=1:length(sigma)
15         w(u) = LBO_float(s,sigma(u),r,q,T,N,M,lambda,theta);
        e(L-Lmin+1,u) = norm(w(u)-wex(u),'inf');
17     end
end

19
n = zeros(length(sigma),1); N = 2.^(Lmin:Lmax)'-1;
21 for j = 1:length(sigma)
    b = polyfit(log(N),log(e(:,j)),1); n(j) = b(1);
23 loglog(N,e(:,j),'k+',N,exp(b(2))*N.^n(j),'r-'), xlabel('N'), ylabel
    ('e_N'), hold on
end
```

Algorithmus A.3: Routine Kovergenzpruefung.m

## A.4 Code LBO\_float\_diskret\_put.m

```

1 % w = LBO_float_put(s,sigma,r,q,T,tj,N,M,lambda) gibt den
% Wert einer europaeischen lookback strike call option
3 % mit Auszahlungsfunktion: g=max(0,lambda-eta)
%
5 % Parameter:
% r=Zinssatz
7 % T=Maturitaet
% q=Dividende
9 % sigma=Volatilitaet
% tj=definierte Zeitpunkte
11 % s=Basiswert
% lambda=Multiplikator
13 %
%z.B. fuer
15 %s=100; sigma=.3; r=.1; q=0; T=0.5; lambda=1; theta=.5; N=2^10-1; M
    =100;
%tj=[1:40]'/40*T;
17 %w = LBO_float_diskret_put(s,sigma,r,q,T,tj,N,M,lambda,theta)

19 function w = LBO_float_diskret_put(s,sigma,r,q,T,tj,N,M,lambda,
    theta)

21 tjx=[0;tj];
g=@(eta)max(0,eta-lambda);
23
a=@(eta)-0.5*sigma^2*eta.^2;
25 b=@(eta)-(q-r)*eta;
c=@(eta)q*eta.^0;
27 w1=@(t)0*t;
wr=@(t)0*t;
29 etal=0;
etar=4;
31
[eta,u] = parabol_dgl_BC(a,b,c,tjx(length(tjx))-tjx(length(tjx)-1),
    etal,w1,3,etar,wr,2,g,N,M,theta);
33
for j = 2:length(tj)
35 idx=find(eta==1);
u(1:idx)=u(idx);
37
g=@(eta)u;
39 revnum=length(tj)+2-j;

```



```
41 [eta,u] = parabol_dgl_BC(a,b,c,tjx(revnum)-tjx(revnum-1),etal,wl,3,  
    etar,wr,2,g,N,M,theta);  
end  
43 w=s.*interp1(eta,u,1,'pchip','extrap');
```

Algorithmus A.4: Routine LBO\_float\_diskret\_put.m