

**UNIVERSITÄT  
BAYREUTH**

Fakultät für Mathematik, Physik und Informatik  
Mathematisches Institut

# **Multi-Level Monte-Carlo-Methoden und ihre Anwendung in der Finanzmathematik**

Masterarbeit  
von  
Sascha Pezoldt

**Aufgabenstellung und Betreuung**

Prof. Dr. Lars Grüne

Bayreuth, 29. August 2016



# Danksagung

Ich möchte mich ganz herzlich bei allen bedanken, die nicht nur zum Gelingen dieser Masterarbeit, sondern auch meines gesamten Studiums beigetragen haben.

Besonderen Dank gilt Herrn Professor Dr. Grüne, nicht nur für die interessante Themenstellung und die exzellente Betreuung, sondern auch für das Halten der Vorlesung „Numerische Methoden der Finanzmathematik“, die mein Interesse an dem gesamten Themengebiet erstmals weckte.

Weiterhin danke ich natürlich meinen Freunden, Studienkollegen und meiner Familie für all die Unterstützung, die ich während meines Studiums bekommen habe. Besonderen Dank gilt natürlich auch all den Korrekturlesern, die hoffentlich die restlichen Fehler gefunden haben...



## Zusammenfassung/Abstract

**Zusammenfassung.** Monte-Carlo Methoden werden im Bereich der Finanzmathematik sehr häufig eingesetzt und es gibt eine Vielzahl an Literatur, die sich mit den verschiedensten Verbesserungsmöglichkeiten des Verfahrens auseinandersetzt. In dieser Masterarbeit soll basierend auf der Arbeit von Prof. Dr. Giles der relativ neue Ansatz der multi-level Monte-Carlo Methoden erläutert und diskutiert werden. Der Ansatz basiert auf der Zerlegung des zu schätzenden Erwartungswerts in eine Teleskopsumme an Erwartungswerten, die einzeln mit Hilfe des Monte-Carlo Verfahrens ermittelt werden. Wir werden zeigen, dass der Rechenaufwand so erheblich reduziert werden kann, wobei die Verbesserung des Rechenaufwands abhängig von dem eingesetzten Approximationsverfahren der zugrundeliegenden stochastischen Differentialgleichung ist. Betrachtet werden vor allem europäische, asiatische und digitale Optionen, wobei die analytisch hergeleiteten Ergebnisse durch numerische Simulationen in MATLAB gestützt werden.

**Abstract.** Monte-Carlo methods are a frequently used tool in financial engineering and there is a multitude of literature dealing with the subject of improving the original method. In this paper we will introduce and discuss the rather new idea of a multilevel Monte-Carlo method, which is based on the work of Prof. Dr. Giles of Oxford University. The basic approach is similar to the one used in multigrid methods, in so far, that rather than calculating the expected value, which arises from the payoff function of a financial derivative, we use the Monte-Carlo method to calculate a telescoping series of expected values. We will show, that by doing so, the computational complexity can be significantly reduced, although the extent of the complexity reduction depends, on which discretisation scheme of the underlying stochastic differential equation is used. In this paper we will mainly focus on european, asian and digital options and we will support the analytical results with numerical simulations in MATLAB.

# Inhaltsverzeichnis

<b>Zusammenfassung/Abstract</b> .....	<b>I</b>
<b>Inhaltsverzeichnis</b> .....	<b>II</b>
<b>Tabellenverzeichnis</b> .....	<b>IV</b>
<b>Abbildungsverzeichnis</b> .....	<b>V</b>
<b>1 Grundlagen der Optionsbewertung</b> .....	<b>1</b>
1.1 Optionen .....	1
1.1.1 Auszahlungsprofile .....	2
1.1.2 Risikoneutrale Bewertung .....	4
1.1.3 Grundalgorithmus der Optionsbewertung.....	5
1.2 Simulation des Basiswerts .....	6
1.2.1 Stochastische Differentialgleichungen.....	6
1.2.2 Numerische Lösung stochastischer Differentialgleichungen .....	11
1.3 Die Monte-Carlo Methode .....	16
1.3.1 Mathematik der MC Methode.....	16
1.3.2 Numerische Umsetzung und Analyse .....	19
1.3.3 Beispiele.....	21
<b>2 Die multi-level Monte-Carlo Methode</b> .....	<b>25</b>
2.1 Herleitung der MLMC Methode.....	25
2.2 Komplexitätstheorem .....	31
2.3 Analyse und Erweiterungen.....	38
2.3.1 Optimales M.....	38
2.3.2 Bias-Schätzung und die Richardson Extrapolation.....	40
2.3.3 Anzahl der Simulationen.....	44
2.4 Implementierung und Anwendung .....	45
2.4.1 Numerischer Algorithmus .....	45
2.4.2 Beispiele.....	47
<b>3. Die MLMC Methode und das Milstein-Verfahren</b> .....	<b>54</b>
3.1 Eindimensionale SDGL.....	54
3.1.1 Optimales M.....	60
3.1.2 Europäische Optionen .....	62
3.1.3 Asiatische Optionen .....	64
3.1.4 Digitale Optionen .....	66
3.2 Mehrdimensionale SDGL.....	71
<b>4. Multi-level Quasi-Monte-Carlo Methoden</b> .....	<b>73</b>
4.1 Prinzip der Quasi-Monte-Carlo Methode .....	73
4.2 Multi-level Ansatz für QMC Methoden .....	78
4.2.1 MLQMC Algorithmus .....	78
4.2.2 Beispiel.....	79
<b>5. Fazit und Ausblick</b> .....	<b>82</b>

---

<b>Literaturverzeichnis.....</b>	<b>83</b>
<b>Inhalt der DVD-ROM.....</b>	<b>85</b>

## Tabellenverzeichnis

Tabelle 1.1: Bewertungsergebnisse einer europäischen Option mittels der MC Methode .....	23
Tabelle 1.2: Monte-Carlo Ergebnisse mit und ohne antithetische Zufallsvariablen .....	24
Tabelle 2.1: Ergebnisse einer europäischen Call Bewertung mittels der MLMC Methode.....	47

# Abbildungsverzeichnis

Abbildung 1.1: Profitfunktion eines europ. Calls und Puts am Laufzeitende .....	2
Abbildung 1.2: Darstellung zweier Pfade eines Wiener-Prozesses wie in [3] .....	8
Abbildung 1.3: Pfade der geometrischen Brownschen Bewegung und Erwartungswert wie in [5] .....	10
Abbildung 2.1: Wiener-Prozess $W_t$ in blau und $W_{t-1}$ in rot für $M = 3$ .....	27
Abbildung 2.2: Graph der Funktion $f(M)$ für die Bestimmung des optimalen $M$ ([9]) .....	40
Abbildung 2.3: Vergleich der Entwicklung der MC/MLMC Varianz- und Erwartungswertschätzer bei der europäischen Call-Option .....	48
Abbildung 2.4: $N_t$ für verschiedene $\epsilon$ beim europ. Call jeweils mit und ohne Rich. Ext. ....	49
Abbildung 2.5: Entwicklung der Rechenkosten bei MC und MLMC mit und ohne Rich. Ext. .	50
Abbildung 2.6: Vergleich der Entwicklung der MC/MLMC Varianz- und Erwartungswertschätzer bei der asiatischen Call-Option .....	51
Abbildung 2.7: Entwicklung der Simulationsanzahl und Rechenkosten der MC und MLMC mit und ohne Rich. Ext. für eine asiatische Call-Option .....	52
Abbildung 2.8: Vergleich der Entwicklung der MC/MLMC Varianz- und Erwartungswertschätzer bei der digitalen Call-Option .....	52
Abbildung 2.9: Entwicklung der Simulationsanzahl und Rechenkosten der MC und MLMC mit und ohne Rich. Ext. bei der digitalen Call-Option .....	53
Abbildung 3.1: Graph der Funktion $f(M)$ im Milstein-Verfahren .....	61
Abbildung 3.2: Varianz und Erwartungswertentwicklung der europäischen Call-Option unter Verwendung des Milstein-Verfahrens .....	63
Abbildung 3.3: Anzahl an Simulationen pro Level und Rechenkosten in Abhängigkeit der Genauigkeit beim Milstein-Verfahren .....	63
Abbildung 3.4: Varianz und Erwartungswertentwicklung der asiatischen Call-Option unter Verwendung des Milstein-Verfahrens .....	65
Abbildung 3.5: Anzahl an Simulationen pro Level und Rechenkosten für die asiatische Option beim Milstein-Verfahren .....	66
Abbildung 3.6: Entwicklung der Varianz und des Schätzers für eine digitale Option unter Verwendung des Milstein-Verfahrens und den Schätzern (3.11),(3.12) .....	69
Abbildung 3.7: Anzahl an Simulationen pro Level und Veranschaulichung der Rechenkosten für die digitale Option beim Milstein-Verfahren und Schätzer (3.11),(3.12) .....	70
Abbildung 4.1: Darstellung der ersten beiden Dimensionen einer rank-1 lattice rule im Vergleich zu pseudozufälligen Punkten wie in [21] .....	75
Abbildung 4.2: Darstellung der ersten beiden Dimensionen einer Sobol-Sequenz im Vergleich zu pseudozufälligen Punkten wie in [21] .....	76
Abbildung 4.3: Links die Entwicklung der Varianz von $Pl$ der normalen MC Methode und $Pl - 1 - Pl$ der MLMC und MLQMC Methode, rechts die Entwicklung von $E[Pl]$ und $E[Pl - 1 - Pl]$ .....	80
Abbildung 4.4: Links die Anzahl der Simulationen pro Level für verschiedene Genauigkeiten $\epsilon$ , rechts Darstellung des Rechenaufwands im Vergleich zur MLMC und MC Methode ....	80



# 1 Grundlagen der Optionsbewertung

Auf den heutigen Finanzmärkten gibt es eine Vielzahl an handelbaren Derivaten mit den unterschiedlichsten Auszahlungsstrukturen und Verwendungsmöglichkeiten. Im folgenden Kapitel soll zuerst wie in [1] und [2] die Option als bedingtes Termingeschäft vorgestellt und einige der verschiedenen Ausprägungen von Optionskontrakten definiert werden. Anschließend werden die Grundprinzipien der Optionsbewertung erläutert, sowie die später benötigten Resultate aus der numerischen Finanzmathematik formell festgehalten. Zum Ende des ersten Kapitels wird die einfache Monte-Carlo Methode eingeführt und verwendet um eine europäische Option zu bewerten. Diese Methode der Optionsbewertung wird in den nachfolgenden Kapiteln als Basis für Vergleiche bzgl. der Effizienz und Qualität der im weiteren Verlauf vorgestellten Algorithmen dienen.

## 1.1 Optionen

Eine Option ist ein Vertrag, der dem Inhaber das Recht, aber nicht die Pflicht, gibt zu einem vorher bestimmten *Ausübungspreis*  $K$  einen *Basiswert*  $S$  zu kaufen, oder zu verkaufen. Bei einer Kaufoption spricht man auch von einem *Call*, bei einer Verkaufsoption von einem *Put*. Da der Wert der Option abhängig von einem Basiswert ist, fällt sie unter die Kategorie der *Derivate*. Als Basiswert kann dabei zum Beispiel der Wert einer Aktie, eines Rohstoffs, oder sogar eine andere Option dienen.

Im Falle einer *europäischen Option* gibt es genau einen zum Vertragsabschluss festgelegten *Ausübungszeitpunkt*  $T$ , an dem der Halter sein Kauf-, bzw. Verkaufsrecht ausüben kann. Bei einer *amerikanischen Option* hingegen ist eine Ausübung jederzeit bis zum Fälligkeitszeitpunkt möglich. In beiden Fällen verfällt das Ausübungsrecht nach dem Zeitpunkt  $T$ .

Oft wird der Basiswert bei Ausübung nicht tatsächlich übergeben, sondern ein finanzieller Ausgleich, auch *glattstellen* genannt, findet statt. Dies ermöglicht ein breites Spektrum an Verwendungsmöglichkeiten, wie etwa die Spekulation auf einzelne Aktienkursverläufe. Zudem erlauben Index-Optionen, zum Beispiel basierend auf dem DAX, Spekulationen bezüglich der Gesamtmarktentwicklung. Neben dem *spekulativen* Einsatz können Optionen auch verwendet werden um sich gegen Risiken, wie zum Beispiel dem Währungsschwankungsrisiko, *abzusichern*.

Bevor wir uns der Aufgabe der fairen Bewertung von Optionen stellen, soll im Folgenden zuerst das Auszahlungsprofil einiger verschiedener Optionstypen definiert werden. Näher vorgestellt werden neben dem *europäischen Call* und *Put* auch *asiatische Optionen*, *lookback Optionen* und *digitale Optionen*.

### 1.1.1 Auszahlungsprofile

Im Folgenden sei stets

- $S(t)$  : Wert des Basiswerts zum Zeitpunkt  $t$
- $V(t, S)$  : Preis der Option zum Zeitpunkt  $t$
- $K > 0$  : vereinbarter Ausübungspreis
- $T > 0$  : Fälligkeitszeitpunkt

#### Definition 1.1

Für eine *europäische Call-Option* gilt am Laufzeitende

$$V_C(T, S(T)) = \max(S(T) - K, 0). \quad (1.1)$$

Analog formuliert man für den *europäischen Put*

$$V_P(T, S(T)) = \max(K - S(T), 0). \quad (1.2)$$

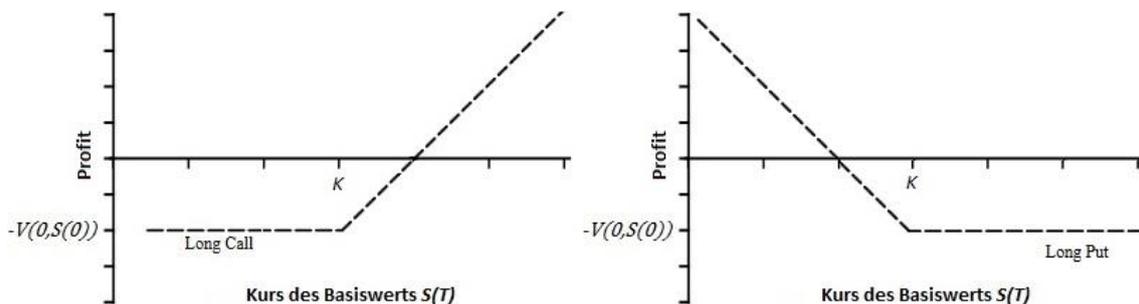


Abbildung 1.1: Profitfunktion eines europ. Calls und Puts am Laufzeitende

Die in Abbildung 1 dargestellten Profitfunktionen verdeutlichen, weshalb Optionen häufig zum Spekulieren verwendet werden: Trotz eines gedeckelten Verlustrisikos sind sehr hohe Gewinne möglich.

#### Definition 1.2

Der Wert einer *asiatischen Option* am Laufzeitende ist gegeben durch

$$V(T, S(T)) = \max\left(\frac{1}{T} \int_0^T S(\tau) d\tau - K, 0\right). \quad (1.3)$$

Im Gegensatz zur europäischen Option, bei der nur der Wert des Basiswerts zum Fälligkeitszeitpunkt  $T$  eine Rolle spielte, betrachtet man hier den gesamten Kursverlauf des Basiswerts.

Asiatische Optionen können vom Ausübungsrecht sowohl europäischer, als auch amerikanischer Natur sein. Sie eignen sich aufgrund ihrer vergleichsweise „trägen“ Art der Wertänderung besonders zum Absichern von Währungsschwankungsrisiken, sowie Rohstoffpreisrisiken.

### Definition 1.3

Die Auszahlungsfunktion einer *lookback Option* lautet

$$V(T, S(t)) = S(T) - \min_{0 \leq t \leq T} S(t). \quad (1.4)$$

Der Ausübungspreis wird erst am Ende der Laufzeit bestimmt als Minimum der tatsächlichen aufgetretenen Kurswerte. Für den Halter der Option gilt somit automatisch der bestmögliche Ausübungspreis und der Wert der Option ist dadurch immer größer gleich null.

Lookback Optionen scheinen besonders gut geeignet zu sein, wenn man einen kurzzeitigen Kurseinbruch mit anschließender Erholung vermutet. Ein absehbarer Termin für so ein Ereignis wäre zum Beispiel der Tag der Dividendenausschüttung, weswegen eine möglichst genaue Modellierung in der Praxis an dieser Stelle für den Verkäufer der Option unerlässlich ist.

### Definition 1.4

Für das Auszahlungsprofil einer *digitalen Option* gilt

$$V(T, S(T)) = H(S(T) - K), \quad (1.5)$$

wobei  $H(\cdot)$  die *Heaviside-Funktion* bzw. Treppenfunktion ist:

$$H: \mathbb{R} \rightarrow \{0, 1\}$$

$$x \mapsto \begin{cases} 0: x < 0 \\ 1: x \geq 0 \end{cases}$$

Digitale Optionen, oder auch *binäre Optionen* genannt, fallen in die Kategorie der exotischen Optionen und sind von der Auszahlungsstruktur her einer der einfachsten Kontrakte. Sie entsprechen der einfachen Wette um einen festen Betrag, ob der Basispreis am Ende der Laufzeit über einem bestimmten Wert liegt. Viele andere exotische Optionen beinhalten oft eine *knock-in* bzw. *knock-out* Bedingung, das heißt der Optionskontrakt ist nur dann überhaupt gültig, sofern ein vorgegebener Wert während der Laufzeit überschritten, bzw. nie unterschritten wird.

Die oben vorgestellten Optionen sind nur einige der vielen handelbaren Derivate, die es erlauben auf verschiedenste Marktentwicklungen zu setzen. Komplexe Finanzmarktprodukte bestehen oft aus einer Kombination verschiedener Derivate deren gemeinsame Auszahlungsstruktur es ermöglicht auf sehr spezielle Marktszenarien zu spekulieren, oder sich gegen bestimmte Entwicklungen abzusichern.

### 1.1.2 Risikoneutrale Bewertung

Da der Wert aller eben vorgestellten Optionen am Laufzeitende immer größer oder gleich null ist, ist klar, dass der Verkäufer der Option – meistens eine Bank – einen Preis für den Verkauf verlangen muss. Die Bewertungstheorie geht auf die Arbeit von Black, Scholes und Merton zurück und ist unter dem *Prinzip der risikoneutralen Bewertung* bekannt. Bevor wir uns diesem wichtigen Konzept nähern, muss als erstes geklärt werden, welche Voraussetzungen an den hier betrachteten Markt gestellt werden. Die folgenden Einschränkungen werden es uns erlauben den Markt mathematisch zu modellieren und Aussagen über den Preis von Finanzderivaten zu treffen.

Wir betrachten ein dividendenloses Marktmodell ohne Transaktionskosten in dem es zu jedem Zeitpunkt möglich ist zu einem konstanten Zinssatz  $r$  Geld als Kredit aufzunehmen, oder in einer risikolosen, festverzinslichen Anlage anzulegen. Die Zinsen werden kontinuierlich vergütet und der verzinste Wert einer Anlage in Höhe von  $x$  Euro zum Zeitpunkt  $t$  ergibt sich zu  $e^{rt}x$ . Der Markt ist zudem arbitragefrei, das heißt es ist nicht möglich ohne Risiko eine höhere Rendite als den Marktzins zu erzielen. Erweiterungen wie Dividenden oder die Einführung variabler Zinsen (Zinsstrukturkurven) lassen sich in komplexeren Modellen zum Teil verwirklichen, die Annahme über die Verfügbarkeit von Krediten ist jedoch zum Beispiel essentiell.

Die eben gestellten Annahmen lassen bereits erahnen in welche Richtung der Ansatz zur Optionsbewertung gehen wird. Angenommen man kennt den genauen Wert einer Investition  $P$  zum Zeitpunkt  $T$ , zum Beispiel eine vertraglich zugesicherte, einmalige Auszahlung in Höhe von  $P(T)$  Euro. Um den Wert dieser Auszahlung zu einem vorherigen Zeitpunkt  $t < T$  zu bestimmen muss man sich die Frage stellen, wieviel Geld man in  $t$  risikolos zum Zinssatz  $r$  anlegen muss um in  $T$  ebenfalls  $P(T)$  Euro zu haben. Da nach obigen Marktannahmen der Zinssatz über die Zeit konstant und die Verzinsung kontinuierlich ist, entspricht dieser Betrag gerade dem abgezinsten Wert der Anleihe

$$P(t) = e^{r(t-T)}P(T). \quad (1.6)$$

Jeder andere Wert würde durch ein frühzeitiges Kaufen oder Verkaufen der Anleihe eine Arbitragemöglichkeit erlauben, was wiederum den Annahmen des Marktmodells widerspricht. Formel (1.6) liefert also den *fairen Preis* der Investition in  $t$ . Dieses Prinzip der *Abzinsung*, oder auch *Diskontierung* genannt, wird auf die Bewertung von Optionen übertragen. Im Gegensatz zu einer Baranleihe, die bei Ablauf eine sichere Auszahlung liefert, hängt der Wert einer Option allerdings von der unbekanntem und vor allem zufälligen Höhe des Basiswerts ab. Die Idee der risikoneutralen Bewertung ist nun den Erwartungswert der Option zum Zeitpunkt  $T$  abzuzinsen.

$$V(t, S(t)) = e^{r(t-T)}E(V(T, S(T))) \quad (1.7)$$

Der Name „risikoneutral“ ist daher passend, da man Kursänderungsrisiken neutral gegenübersteht und den Erwartungswert wie den sicheren Endwert behandelt. Bei einer risikoaversen Bewertung hingegen würde man einen stärkeren Diskontierungsfaktor als  $e^{r(t-T)}$  verwenden, und bei einer risikofreudigen Bewertung entsprechend einen schwächeren.

Tatsächlich lässt sich das Abzinsen des Erwartungswertes insofern begründen, dass der faire Wert einer Option nicht der wirklich realisierten Auszahlung in der Zukunft entspricht. Vielmehr betrachtet man das Kapital, das der Verkäufer der Option aufbringen muss, um sich gegen die anfallende Zahlung in zufälliger Höhe abzusichern. Dieses Absichern, auch *Hedging* genannt, erfolgt durch die Investition des Kaufpreises in ein Portfolio bestehend aus dem Basiswert und der risikolosen Anleihe. Die Kursänderungen des Basiswerts müssen durch entsprechendes umschichten innerhalb des Portfolios kompensiert werden. Steigt der Kurs des Basiswerts muss dieser etwa auch einen größeren Anteil des Portfolios ausmachen um den Wertzuwachs auszugleichen. Sinkt der Kurs wird mehr in die festverzinsliche Anlage investiert.

### 1.1.3 Grundalgorithmus der Optionsbewertung

Mit Hilfe des Prinzips der risikoneutralen Bewertung können wir nun einen Grundalgorithmus zur Optionsbewertung formulieren. Der folgende Algorithmus, wie er auch in [1] zu finden ist, ist in erster Linie konzipiert für die Bewertung europäischer Optionen, er lässt sich aber mit kleinen Erweiterungen leicht auf andere Optionen übertragen.

#### Algorithmus 1.5

- (1) Bestimme eine Formel für den Wert  $V(T, S)$  zum Laufzeitende in Abhängigkeit vom Kurs  $S = S(T)$  am Laufzeitende
- (2) Bestimme ausgehend vom Basiswert  $S(t)$  zur Zeit  $t < T$  mit Hilfe eines stochastischen Modells die Zufallsvariable  $S(T) = S(T, \omega)$
- (3) Berechne den Optionswert als den abgezinsten Erwartungswert

$$V(t, S(t)) = e^{r(t-T)} E \left( V(T, S(T)) \right)$$

Schritt (1) des Algorithmus entspricht gerade den Formeln (1.1) – (1.5) am Anfang des Kapitels. Im Falle der europäischen, bzw. digitalen Option ist der Endwert tatsächlich nur abhängig vom Kurs des Basiswerts am Ende der Laufzeit. Die Auszahlungshöhe der anderen Optionen hingegen ist zudem bestimmt durch den minimal angenommenen Kurswert (lookback Option), oder den Gesamtkursverlauf (asiatische Option).

Hinter dem zweiten Schritt verbirgt sich ein Großteil der mathematischen Modellierungsarbeit. Die Kurssimulation des Basiswerts erfolgt über ein stochastisches Modell, in dem der Anwender

versucht durch geschickte Wahl der Eingabeparameter die Wirklichkeit möglichst gut nachzubilden. Zur Auswahl steht dabei eine Vielzahl an Modellen, die sich untereinander teils erheblich unterscheiden. Einer der offensichtlichsten Unterschiede ist dabei die Unterteilung in zeitdiskrete und zeitstetige Modelle. Die später verwendete stochastische Differentialgleichung, die zu den zeitstetigen Modellen zählt, ist als geometrische Brownsche Bewegung bekannt und bildet auch die Grundlage des berühmten Black-Scholes Modells. Für Optionen, deren Auszahlungsfunktion nicht nur abhängig vom Endwert  $S(T)$  ist, müssen innerhalb desselben Modells auch die restlichen relevanten Werte bestimmt werden.

Schritt (3) entspricht gerade dem Prinzip der risikoneutralen Bewertung. Die tatsächliche numerische Implementierung ist allerdings abhängig von der Umsetzung von Schritt (2). Je nach verwendetem Modell kann der Erwartungswert zum Beispiel analytisch berechenbar sein, oder etwa auch durch eine Vielzahl an Simulationen geschätzt werden. Letzteres ist gerade der Fall bei den später untersuchten Monte-Carlo Methoden.

Zu bemerken gilt zudem, dass für Optionen mit amerikanischen Ausübungsrecht der ermittelte Wert aus Schritt (3) zu jedem Zeitpunkt verglichen werden muss mit dem Resultat, welches bei sofortiger Ausübung erreichbar ist. Dies funktioniert nicht ohne weiteren Aufwand für alle Verfahren, Monte-Carlo Methoden benötigen hier zum Beispiel weitere Ansätze um eine korrekte Bewertung zu ermöglichen.

## 1.2 Simulation des Basiswerts

Für die Umsetzung von Schritt (2) im Algorithmus 1.5 wird ein stochastisches Modell benötigt um die Entwicklung des Basiswerts zu simulieren und damit die auszahlungsrelevanten Größen zu bestimmen. Um die Realität einigermaßen nachbilden zu können, sollte der Anwender dabei mindestens in der Lage sein den erwarteten Mittelwert und die Varianz des Kurses über die Eingabeparameter anzupassen. In der Finanz- und Versicherungsmathematik werden für Simulationen von unsicheren, zukünftigen Ereignissen meist stochastische Differentialgleichungen verwendet. Im folgenden Abschnitt werden diese zunächst allgemein eingeführt und anschließend zwei mögliche numerische Approximationsverfahren vorgestellt.

### 1.2.1 Stochastische Differentialgleichungen

Eine stochastische Differentialgleichung (SDGL) ist eine Differentialgleichung die zusätzlich zu ihren deterministischen Komponenten auch vom Zufall abhängige Einflüsse besitzt. Dieser stochastische Einfluss kommt dabei über eine zufällige Funktion, auch als *stochastischer Prozess* bezeichnet, in die SDGL.

**Definition 1.6**

Sei  $(\Omega, \mathcal{A}, \mathcal{P})$  ein Wahrscheinlichkeitsraum und  $(\mathbb{R}, \mathcal{B})$  der Raum der reellen Zahlen mit der dazugehörigen Borelschen  $\sigma$ -Algebra. Die Funktion

$$X : \mathbb{R} \times \Omega \rightarrow \mathbb{R}$$

heißt *stochastischer Prozess* und ist für jedes feste  $t \in \mathbb{R}$  eine Zufallsvariable:

$$X(t, \cdot) : \Omega \rightarrow \mathbb{R}$$

Alle Zufallsvariablen  $X(t, \cdot)$  sind dabei über denselben Wahrscheinlichkeitsraum  $(\Omega, \mathcal{A}, \mathcal{P})$  definiert. Für jedes feste  $\omega \in \Omega$  ist die Realisierung

$$X(\cdot, \omega) : \mathbb{R} \rightarrow \mathbb{R}$$

eine reelle Funktion und wird als *Pfad des stochastischen Prozesses* bezeichnet.

Stochastische Prozesse sind keine kompletten „black boxes“, sondern lassen sich – zum Beispiel über ihre Verteilung – mathematisch beschreiben und dadurch, je nach gewünschten Anwendungsgebiet, entsprechend auswählen. In der Schadensversicherungsmathematik werden etwa Poisson-Prozesse eingesetzt, die im Wesentlichen Treppenfunktionen sind, die zu zufälligen Zeitpunkten um genau eins steigen. Die Erhöhung um eins soll dabei das zufällige Eintreten eines Versicherungsfalls beschreiben. Besonders wichtig in der Finanzmathematik ist der *Wiener-Prozess*, der die *Brownsche Bewegung*, also die zufällige Bewegung von Teilchen auf der Wasseroberfläche, simuliert.

**Definition 1.7**

Der stochastische Prozess  $W(t, \omega)$  für  $t \geq 0$  heißt *Wiener-Prozess*, wenn er die folgenden Bedingungen erfüllt:

- (1)  $W(t)$  ist eine normalverteilte Zufallsvariable mit  $E(W(t)) = 0$  und  $Var(W(t)) = t$ , wobei die Dichtefunktion der Normalverteilung mit Erwartungswert  $\mu$  und Varianz  $\sigma^2$  gegeben ist durch

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

- (2) Für die Zeiten  $t_1 \geq t_0 \geq 0$  sind die Inkremente  $W(t_1) - W(t_0)$  normalverteilte Zufallsvariablen mit  $E(W(t_1) - W(t_0)) = 0$  und  $Var(W(t_1) - W(t_0)) = t_1 - t_0$
- (3) Für  $s_1 \geq s_0 \geq t_1 \geq t_0 \geq 0$  sind die Inkremente  $W(t_1) - W(t_0)$  und  $W(s_1) - W(s_0)$  unabhängige Zufallsvariablen.

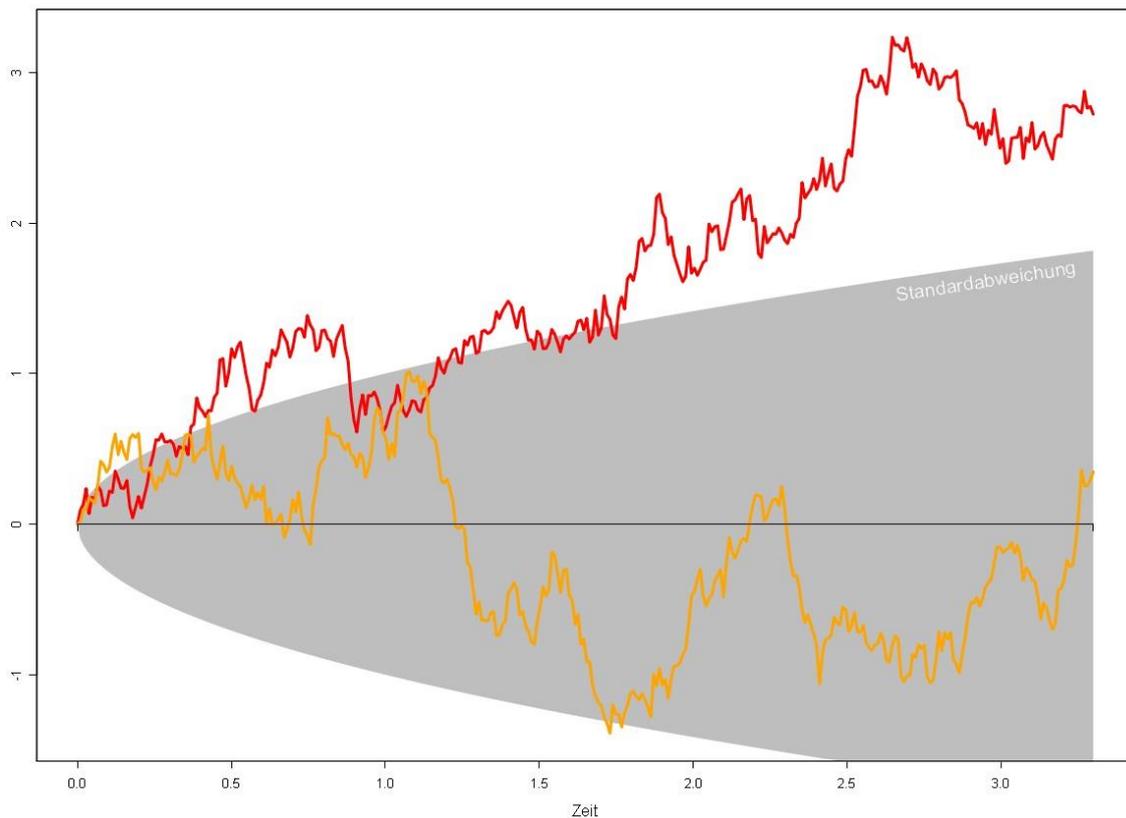


Abbildung 1.2: Darstellung zweier Pfade eines Wiener-Prozesses wie in [3]

Die erste Bedingung aus der Definition des Wiener-Prozesses besagt, dass der Erwartungswert zu jedem Zeitpunkt gleich null ist, die Streuung der Pfade aber mit der Zeit ansteigt. Der grau hinterlegte Bereich der Standardabweichung in Abbildung 1.2 wird mit fortschreitender Zeit größer und der Rand wird beschrieben durch  $\pm\sqrt{t}$ , da die Varianz ja zu jedem Zeitpunkt gerade  $t$  ist. Bedingung (2) hat zur Folge, dass egal welchen Punkt eines Wiener-Prozesses man auswählt, der resultierende Pfad ab diesem Zeitpunkt durch Verschiebung in den Ursprung ein komplett neuer, eigenständiger Pfad des stochastischen Prozesses ist. Die dritte Bedingung über die Unabhängigkeit der Inkremente bedeutet, dass man zu keinem Zeitpunkt vorhersagen kann, ob der Kurs fallen oder steigen wird, auch wenn der vorherige Kursverlauf bekannt ist. Die zukünftige Entwicklung ist also vollkommen unabhängig von den vorhergehenden Werten.

Es lässt sich zeigen, dass die Pfade eines Wiener-Prozesses stetig aber nirgends differenzierbar sind. Dies wirft einige Fragen auf, betrachtet man etwa die Differentialgleichung

$$\frac{d}{dt}X(t) = \frac{d}{dt}W(t)$$

bzw. die Darstellung in Integralform

$$X(t) = X_0 + \int_0^t \frac{d}{d\tau}W(\tau)d\tau$$

scheint zunächst unklar wie das Integral überhaupt zu interpretieren ist. Die Antwort darauf fand der japanische Mathematiker Kiyosi Itô in den 1940er Jahren. Die Lösung verwendet das nach ihm benannte Itô-Integral, für dessen genauere Interpretation und Herleitung hier nur auf [1] Kapitel 4 und [4] verwiesen sei. Dieses neue Integral erlaubt es uns nun *Itô-stochastische Differentialgleichungen* zu definieren.

### Definition 1.8

Eine *Itô-stochastische Differentialgleichung* mit den Funktionen  $a, b : \mathbb{R} \times \mathbb{R}_{+,0} \rightarrow \mathbb{R}$  und einem Wiener-Prozess  $(W_t)_{t \geq 0}$  ist gegeben durch

$$X(t) = X(t_0) + \int_{t_0}^t a(\tau, X(\tau))d\tau + \int_{t_0}^t b(\tau, X(\tau))dW_\tau. \quad (1.8)$$

Alternativ schreibt man kürzer in Differentialschreibweise:

$$dX(t) = a(t, X(t))dt + b(t, X(t))dW_t \quad (1.9)$$

Der deterministische Anteil der SDGL wird auch als *Drift* bezeichnet, der stochastische Term als *Diffusion*.

Die eben eingeführte Definition von SDGLen ermöglicht die Formulierung des in Schritt (2) des Grundalgorithmus geforderten stochastischen Modells. Das folgende Modell ist trotz seiner Simplität weit verbreitet und erlaubt, wie gefordert, über die Parameter  $\mu$  und  $\sigma$  die Rendite, bzw. Volatilität des Kurses zu kalibrieren. In der Literatur zur Finanzmathematik ist es üblich bei der Formulierung solcher Gleichungen  $S(t)$  statt  $X(t)$  zu schreiben.

### Definition 1.9

Das Modell, dass durch die stochastischen Differentialgleichung

$$dS(t) = \mu S(t)dt + \sigma S(t)dW_t \quad (1.10)$$

mit  $\mu, \sigma > 0$  beschrieben ist, heißt *geometrische Brownsche Bewegung*.

Wie bereits erwähnt ist dieses Modell eher simpel und wird in der Praxis zum Großteil durch komplexere Modelle ersetzt. Vorteilhaft ist allerdings, dass man für Gleichung (1.10) mithilfe des Itô-Lemmas die Lösung der SDGL ermitteln und damit sowohl Erwartungswert, als auch Varianz angeben kann.

### Satz 1.10

Für die Lösung  $S(t; S_0)$  der geometrischen Brownschen Bewegung gilt

$$S(t; S_0) = S_0 \exp\left(\left(\mu - \frac{1}{2}\sigma^2\right)t + \sigma W(t)\right).$$

Weiterhin gelten für den Erwartungswert und die Varianz die Gleichungen:

$$E(S(t; S_0)) = S_0 e^{\mu t} \quad (1.11)$$

$$\text{Var}(S(t)) = S_0^2 e^{2\mu t} (e^{\sigma^2 t} - 1) \quad (1.12)$$

*Beweis*

Siehe [1] Kapitel 4.

□

In der risikoneutralen Bewertungstheorie gilt die Gleichung:

$$E(S(T)) = e^{rT} S(0)$$

Daraus folgt sofort dass im Modell der geometrischen Brownschen Bewegung der Parameter  $\mu$  gleich dem risikolosen Marktzins  $r$  gesetzt werden muss. Als einstellbarer Parameter bleibt also nur  $\sigma$  zur Kalibrierung der Volatilität übrig.

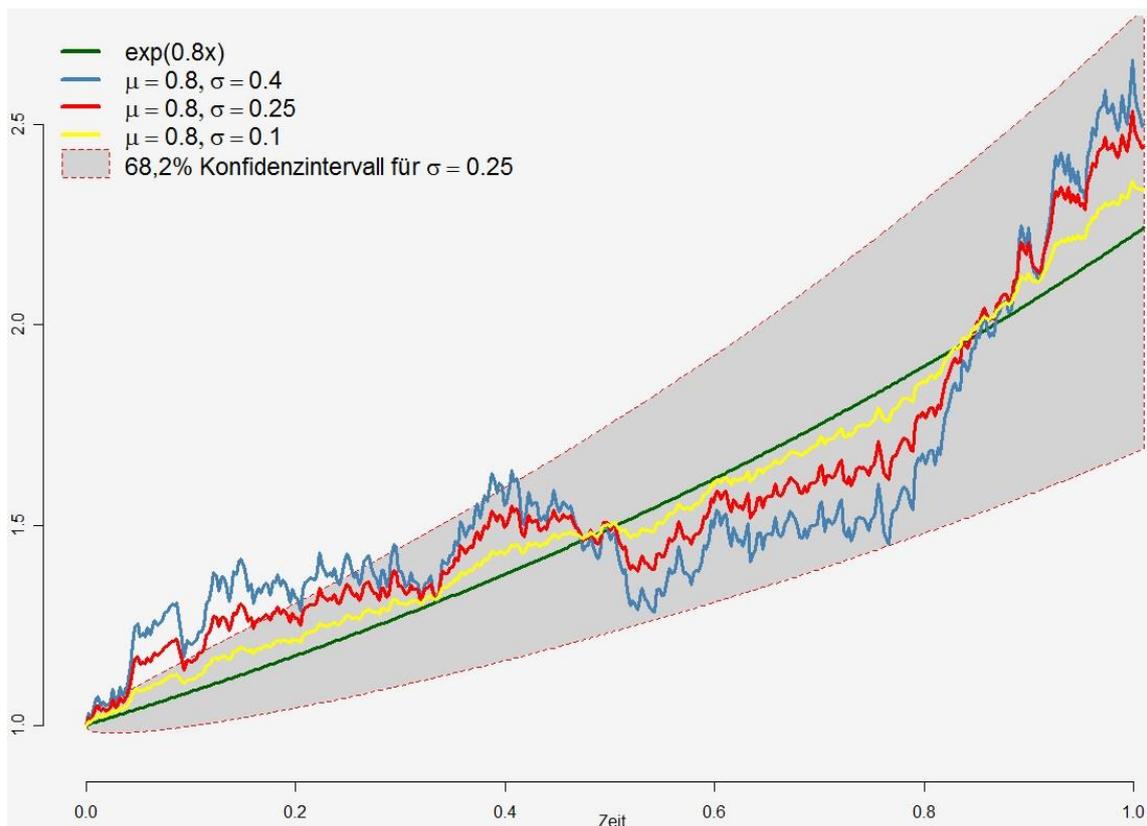


Abbildung 1.3: Pfade der geometrischen Brownschen Bewegung und Erwartungswert wie in [5]

Abbildung 1.3 zeigt eine mögliche Kursentwicklung für  $\mu = 0.8$  und den Sigma Werten 0.1, 0.25 und 0.4. Grün eingezeichnet ist der erwartete Kurswert gegeben durch  $e^{\mu t}$ .

## 1.2.2 Numerische Lösung stochastischer Differentialgleichungen

Komplexere Modelle als die geometrische Brownsche Bewegung lassen sich meist nicht mehr analytisch lösen, weswegen numerische Verfahren zur Lösung eingesetzt werden. Da die einzelnen Pfade eines stochastischen Prozesses jedoch in kurzer Zeit stark schwanken können, kann im Allgemeinen nicht vorausgesetzt werden, dass jede auftretende Realisation gut approximiert wird. Für die meisten Anwendungen in der Finanzmathematik genügt es jedoch die Pfade im Mittel gut zu approximieren, was uns zu den folgenden Konvergenzbegriffen aus der stochastischen Numerik führt.

### Definition 1.11

(i) Die Folge  $\tilde{X}_j$  von stochastischen Prozessen heißt *starke Approximation* für  $X$  zur Zeit  $T$  bzgl. einer Funktion  $g : \mathbb{R}^n \rightarrow \mathbb{R}^m$ , falls die Bedingung

$$\lim_{j \rightarrow \infty} E \left( \left\| g(X(T)) - g(\tilde{X}_j(T)) \right\| \right) = 0$$

gilt. Sie heißt *starke Approximation der Ordnung  $\gamma > 0$* , falls für alle  $j \geq j_0$  zusätzlich die Abschätzung

$$E \left( \left\| g(X(T)) - g(\tilde{X}_j(T)) \right\| \right) \leq C h_j^\gamma$$

für ein  $C > 0$  gilt.

(ii) Die Folge  $\tilde{X}_j$  von stochastischen Prozessen heißt *schwache Approximation* für  $X$  zur Zeit  $T$  bzgl. einer Funktion  $g : \mathbb{R}^n \rightarrow \mathbb{R}^m$ , falls die Bedingung

$$\lim_{j \rightarrow \infty} \left\| E \left( g(X(T)) \right) - E \left( g(\tilde{X}_j(T)) \right) \right\| = 0$$

gilt. Sie heißt *schwache Approximation mit Ordnung  $\beta > 0$* , falls zusätzlich für alle  $j \geq j_0$  die Abschätzung

$$\left\| E \left( g(X(T)) \right) - E \left( g(\tilde{X}_j(T)) \right) \right\| \leq C h_j^\beta$$

für ein  $C > 0$  gilt.

Die Bedingung der starken Approximation stellt eine Verallgemeinerung der deterministischen Definition dar und obwohl nicht ausgeschlossen werden kann, dass einzelne Pfade schlecht approximiert werden, werden die Pfade zumindest im Mittel gut getroffen. Die schwache Approximation erlaubt keine solche Aussage über den mittleren Fehler der Pfade, ist aber dennoch nützlich wenn es um die Approximation der statistischen Eigenschaften geht. Da in der Finanzmathematik häufig der Erwartungswert eines Modells eine zentrale Rolle spielt und nicht die einzelnen

Pfade, ist eine numerisch schwache Approximation aufgrund des deutlich geringeren Rechenaufwands in vielen Fällen lohnenswert. Dass die Bezeichnung der Approximationen als stark und schwach gerechtfertigt ist, zeigt das folgende Lemma.

**Lemma 1.12**

Wenn  $\tilde{X}_j(T)$  eine starke Approximation von  $X(T)$  bzgl.  $g : \mathbb{R}^n \rightarrow \mathbb{R}^m$  ist, so ist  $\tilde{X}_j(T)$  auch eine schwache Approximation von  $X(T)$  bzgl.  $g$ . Hierbei bleibt die Konvergenzordnung erhalten, d.h. es gilt  $\beta \geq \gamma$ .

*Beweis*

Siehe [1], Kapitel 7.

□

Bevor man sich Gedanken über numerische Verfahren für stochastische Differentialgleichungen macht, muss man sich überlegen, wie der eingehende Wiener-Prozess numerisch approximiert werden soll. Unabhängig davon, ob eine starke oder schwache Approximation des Wiener Prozesses gesucht ist, betrachtet man ein Gitternetz  $\mathcal{T} = \{t_0, \dots, t_N\}$  mit  $t_i = ih$ , wobei die Konstante  $h$  der äquidistanten Schrittweite entspricht.

Um eine schwache Approximation zu erreichen betrachtet man die Verteilung

$$X(\Omega) = \{-\sqrt{h}, \sqrt{h}\}; \quad \mathbb{P}_X(\{-\sqrt{h}\}) = \mathbb{P}_X(\{\sqrt{h}\}) = \frac{1}{2},$$

und erzeugt davon  $N$  unabhängige Zufallsvariablen  $\Delta W_0, \dots, \Delta W_{N-1}$ . Der Wert des schwach approximierten Wiener Prozesses zum Zeitpunkt  $t_i$  ist dann gerade durch die Iterationsvorschrift

$$\tilde{W}(t_0, \omega) = 0, \quad \tilde{W}(t_{i+1}, \omega) = \tilde{W}(t_i, \omega) + \Delta W_i \quad (1.13)$$

gegeben. Obwohl die so erzeugten Pfade optisch sehr unterschiedlich zu den richtigen Pfaden sein können, lässt sich zeigen, dass es sich tatsächlich um eine schwache Approximation bzgl. der Funktionen  $g(x) = x$  und  $g(x) = x^2$  handelt.

Um eine starke Approximation zu erhalten, wird anstatt der obigen zweipunktverteilten Zufallsvariable die Normalverteilung mit Erwartungswert 0 und der Varianz  $h$  verwendet. Die erzeugten unabhängigen  $N(0, h)$ -verteilten Zufallsvariablen  $\Delta W_0, \dots, \Delta W_{N-1}$  ergeben durch einsetzen in Formel (1.13) den Wert des Wiener-Prozesses zu den jeweiligen diskreten Zeitpunkten. Bei den so erzeugten Pfaden handelt es sich sogar um „echte“ Pfade des Wiener Prozesses, so dass für jedes  $g : \mathbb{R} \rightarrow \mathbb{R}$  und jeden Zeitpunkt  $t_i$  die Gleichung

$$E \left\{ \left| g(\tilde{W}(t_i, \omega)) - g(W(t_i, \omega)) \right| \right\} = 0$$

erfüllt ist.

Die eben eingeführten Simulationsmethoden für den Wiener-Prozess erlauben nun die Formulierung von Algorithmen zur Approximation von stochastischen Differentialgleichungen. Die aus der Numerik für gewöhnliche DGLen bekannten Klassen der *Einschritt-, Mehrschritt- und Taylorverfahren* lassen sich auch im Bereich der SDGLen wiederfinden. In einigen Fällen, wie zum Beispiel dem Euler-Verfahren, ist es möglich bekannte Verfahren mit kleinen Erweiterungen auch für SDGLen einzusetzen. Das Pendant zum Euler-Verfahren wird *stochastisches Euler-Verfahren*, oder Euler-Maruyama-Verfahren, genannt.

**Definition 1.13**

Die Menge  $\mathcal{T} = \{t_0, \dots, t_n\}$  sei ein Gitter mit  $t_i = ih$  bzw. der Schrittweite  $h$ . Weiterhin sei  $\Delta W(t) = W(t+h) - W(t)$ . Die Approximation durch das *stochastische Euler-Verfahren* für eine SDGL der Form (1.9) ist gegeben durch:

$$\Phi(t, X, h, W, \omega) = X(\omega) + ha(t, X(\omega)) + \Delta W(t, \omega)b(t, X(\omega)) \quad (1.14)$$

**Beispiel 1.14**

Die Approximation  $\tilde{S}$  der geometrischen Brownschen Bewegung mittels des stochastischen Euler-Verfahrens ist gegeben durch die Iterationsvorschrift

$$\tilde{S}_{n+1} = \tilde{S}_n + h\mu\tilde{S}_n + \sigma\tilde{S}_n\Delta W_n \quad (1.15)$$

und dem Startwert  $\tilde{S}_0 = S_0$ .

Untersucht man das stochastische Euler-Verfahren bzgl. der Konvergenzbegriffe aus Definition 1.11, muss man an die Funktionen  $a(\cdot)$  und  $b(\cdot)$  der SDGL zusätzliche Bedingungen stellen um die starke, bzw. schwache Approximationsordnung zu bestimmen. Die Bedingungen und die resultierenden Ordnungen wollen wir im folgenden Satz festhalten.

**Satz 1.15**

Angenommen es gelten für geeignete Konstanten  $K_i$  und alle  $t, s \in [0, T]$ ,  $x, y \in \mathbb{R}^n$  die folgenden Ungleichungen:

$$\|a(t, x) - a(t, y)\| + \|b(t, x) - b(t, y)\| \leq K_1|x - y|$$

$$|a(t, x)| + |b(t, x)| \leq K_2(1 + |x|)$$

$$\|a(s, x) - a(t, x)\| + \|b(s, x) - b(t, x)\| \leq K_3(1 + |x|)|s - t|^{1/2}$$

Dann gilt unter Verwendung der starken Approximation des Wiener-Prozesses:

- (i) Das stoch. Euler-Verfahren besitzt die starke Approximationsordnung  $\gamma = \frac{1}{2}$  bzgl. der Funktion  $g(x) = x$

(ii) Es gilt die Ungleichung

$$E \left( \|X(T) - \tilde{X}(T)\|^2 \right) \leq K_4^2 h. \quad (1.16)$$

Seien  $a(\cdot)$  und  $b(\cdot)$  zudem unabhängig von  $t$  und  $C_p^4$  in  $x$ , dann gilt außerdem unter Verwendung der schwachen Approximation des Wiener-Prozesses:

(iii) Das stoch. Euler Verfahren hat die schwache Approximation der Ordnung  $\beta = 1$  bzgl. jeder  $C_p^4$ -Funktion  $g$ .

*Beweis*

Siehe [1] Kapitel 7 und [4] Kapitel 10. □

Während Teil (i) und (iii) des Satzes Aussagen über die Ordnung machen, ist in (ii) eine Ungleichung festgehalten die während des Beweises in [4] gezeigt wird. Diese Ungleichung spielt normalerweise keine besondere Rolle, wird aber später im Kontext mit der multi-level Monte-Carlo Methode noch einmal benötigt.

Es lässt sich leicht sehen, dass die geometrische Brownsche Bewegung mit  $a(t, S) = \mu S$  und  $b(t, S) = \sigma S$  die Bedingungen aus Satz 1.15 erfüllt und damit die starke Approximationsordnung  $\gamma = 1/2$  erreicht wird für  $g(x) = x$ . Natürlich besitzen die vorgestellten Optionen ein Auszahlungsprofil, welches nicht nur der einfachen Identitätsfunktion entspricht. Dass im Falle einer global Lipschitz-stetigen Auszahlungsfunktionen das Ergebnis trotzdem gültig ist, zeigt das folgende Lemma.

**Lemma 1.16**

Sei  $h : \mathbb{R}^m \rightarrow \mathbb{R}^l$  eine global Lipschitz-stetige Funktion und die Folge  $\tilde{X}_j$  eine starke Approximation der Ordnung  $\gamma > 0$  bzgl. der Funktion  $g : \mathbb{R}^n \rightarrow \mathbb{R}^m$ . Dann ist  $\tilde{X}_j$  auch eine starke Approximation selber Ordnung für die Verknüpfung  $h \circ g$ .

*Beweis*

Sei  $L > 0$  die globale Lipschitz-Konstante der Funktion  $h$ . Für  $h \circ g$  gilt dann:

$$E \left( \left\| h \circ g(X(T)) - h \circ g(\tilde{X}_j(T)) \right\| \right) \leq E \left( L \left\| g(X(T)) - g(\tilde{X}_j(T)) \right\| \right)$$

Und damit aufgrund der Linearität des Erwartungswertes und der gegebenen starken Approximationsordnung von  $\tilde{X}_j$  für  $g$  das gewünschte Ergebnis. □

Eine analoge Aussage für die schwache Approximation lässt sich leider nicht formulieren. Dies ist schade, da wir für die Bewertung von europäischen Optionen primär nur am Erwartungswert

interessiert sind und somit eine schwache Approximation ausreichen würde. Zudem liefert das Euler-Maruyama-Verfahren eine höhere schwache Konvergenzordnung von  $\beta = 1$ , wobei hier zusätzlich gefordert ist, dass die Funktion  $g \in C_p^4$  ist. Allerdings ist keine der Auszahlungsfunktionen der vorgestellten Optionen auch nur einmal stetig differenzierbar, im Falle der digitalen Option ist sie noch nicht einmal stetig. Dass dennoch schwache Konvergenz vorliegt, wird am Beispiel der europäischen Option in [1] Kapitel 7 gezeigt. Dazu betrachtet man für die Analyse eine 4-mal stetig differenzierbare Funktion, die den Knick der Funktion  $g(x) = (x - K)^+$  glättet. Diese, zum Beispiel durch Interpolation erzeugte Funktion erfüllt dann die Bedingungen der schwachen Konvergenz. Das Argument lässt sich auf asiatische und lookback-Optionen verallgemeinern, im Falle der binären Option wird an der Sprungstelle eine  $C_p^4$ -Verstetigung eingeführt. Es gilt zu bemerken, dass in der eigentlichen Implementierung die richtige Auszahlungsfunktion verwendet wird und nicht die zur Analyse eingeführte.

Das stochastische Euler-Verfahren entspricht dem allgemeinen Itô-Taylor-Verfahren mit  $\gamma = 1/2$  und  $\beta = 1$ . Itô-Taylor-Verfahren lassen sich durch die Itô-Taylor-Entwicklung – eine auf SDGLen angepasste Taylor-Entwicklung – herleiten und ermöglichen es Verfahren höherer Ordnung zu konstruieren. Das Verfahren mit der starken und schwachen Approximationsordnung  $\gamma = \beta = 1$  heißt *Milstein-Verfahren* und unterscheidet sich vom stochastischen Euler-Verfahren, für die in Definition 1.8 definierte skalare SDGL mit nur einem stochastischen Prozess, lediglich durch die Addition eines neuen Terms. In diesem Fall lautet die Iterationsvorschrift:

$$\begin{aligned} \Phi(t, X, h, W, \omega) &= X(\omega) + ha(t, X(\omega)) + \Delta W(t, \omega)b(t, X(\omega)) \\ &\quad + \frac{1}{2} \frac{\partial}{\partial X} b(t, X(\omega))b(t, X(\omega))(\Delta W(t, \omega)^2 - h) \end{aligned} \quad (1.17)$$

Die tatsächliche Herleitung dieses Verfahrens, sowie der zu Satz 1.15 analoge Satz über Konvergenzbedingungen für Itô-Taylor-Verfahren würden hier zu weit führen und an dieser Stelle sei nur auf [1] Kapitel 7 und [4] Kapitel 10 verwiesen. Das Milstein-Verfahren und dazu erzeugte numerische Resultate werden im Zusammenhang mit der multi-level Monte-Carlo Methode noch einmal in Kapitel 3 genauer untersucht.

### Beispiel 1.17

Für die geometrische Brownsche Bewegung gilt unter Anwendung des Milstein-Verfahrens die Iterationsvorschrift

$$\tilde{S}_{n+1} = \tilde{S}_n + h\mu\tilde{S}_n + \sigma\tilde{S}_n\Delta W_n + \frac{1}{2}\sigma^2\tilde{S}_n(\Delta W_n^2 - h), \quad (1.18)$$

sowie dem Startwert  $\tilde{S}_0 = S_0$ .

**Bemerkung 1.18**

In Definition 1.8 wurde die Itô-stochastische Differentialgleichung in ihrer eindimensionalen Form eingeführt. Natürlich ist es auch möglich eine mehrdimensionale SDGL aufzustellen und diese numerisch zu approximieren. Im Falle des Milstein-Verfahrens ergeben sich dann allerdings aus der Konstruktion des Verfahrens mehrere Mehrfachintegrale. Diese Mehrfachintegrale, auch bekannt als „Lévy Areas“ sind für Dimensionen größer zwei nur mit enormen Aufwand numerisch zu berechnen. In Kapitel 3 wird die Problemstellung, sowie ein möglicher Lösungsansatz in Bezug zur multi-level Monte-Carlo Methode, kurz skizziert.

### 1.3 Die Monte-Carlo Methode

Bisher wurde gezeigt, wie man den Kursverlauf von Basiswerten mit Hilfe von stochastischen Differentialgleichungen simuliert und die entstehenden Pfade numerisch approximiert. Um die risikoneutrale Bewertungstheorie anwenden zu können, müssen wir allerdings den Erwartungswert am Ende der Laufzeit ermitteln. Da dieser im Allgemeinen unbekannt ist, benötigen wir eine Methode um diesen zu schätzen. Die Monte-Carlo (MC) Methode eignet sich einerseits sehr gut dafür, da sie unabhängig vom eingesetzten Modell anwendbar ist, andererseits werden wir sehen, dass sie nur langsam konvergiert und präzise Ergebnisse daher nur unter großem Rechenaufwand erzeugt werden können. Die Grundidee der Monte-Carlo Methode ist denkbar einfach und lässt sich am folgenden Beispiel erläutern.

Angenommen man hält einen Würfel mit verschiedenen schweren Seiten und möchte den Erwartungswert ermitteln. Da der Würfel aufgrund der ungleichmäßigen Gewichtsverteilung kein Laplace-Würfel ist wird der Erwartungswert natürlich von 3,5 abweichen. Die vielleicht intuitivste Herangehensweise entspricht bereits dem Monte-Carlo Verfahren. Man würfelt ausreichend oft, notiert sich die Ergebnisse und berechnet anschließend das arithmetische Mittel, welches dann einer Näherung an den tatsächlichen Erwartungswert entspricht. Das Problem ist allerdings – wie oft ist „ausreichend“ oft? Den mathematischen Hintergrund, sowie die Güte der Schätzung werden wir im Folgenden erörtern.

#### 1.3.1 Mathematik der MC Methode

Allgemeiner lässt sich das obige Würfelbeispiel so formulieren: Sei  $X$  eine Zufallsvariable mit unbekanntem Erwartungswert  $\mu$  und Varianz  $\sigma^2$ . Um den Erwartungswert zu bestimmen simuliert man  $n$  Realisationen des Zufallsexperiments, meist durch ziehen von computergenerierten Zufallszahlen. Die gezogenen Zufallszahlen  $x_1, x_2, x_3, \dots$  lassen sich sowohl als unabhängige Reali-

sierungen  $x_i = X(\omega_i)$  desselben Zufallsexperiments interpretieren, wie auch als die Realisierungen von unabhängig identisch verteilten (u.i.v.) Zufallsvariablen  $X_i \sim X$ , also  $x_i = X_i(\omega)$ . Dass dann für die neue Zufallsvariable

$$\bar{Y}_n := \frac{1}{n} \sum_{i=1}^n X_i \approx \mu = E(X) \quad (1.19)$$

gilt, folgt aus dem *zentralen Grenzwertsatz der Stochastik*.

### Satz 1.19

Sei  $X_1, X_2, \dots, X_n$  eine Folge von u.i.v. Zufallsvariablen, definiert über denselben Wahrscheinlichkeitsraum  $\mathcal{P}$  und mit Erwartungswert  $\mu$  und Varianz  $\sigma^2$ . Die Zufallsvariable  $Y_n = \sum_{i=1}^n X_i$  besitzt dann den Erwartungswert  $n\mu$ , bzw. die Varianz  $n\sigma^2$ . Die standardisierte Zufallsvariable

$$Z_n = \frac{Y_n - n\mu}{\sigma\sqrt{n}}$$

konvergiert zudem für  $n \rightarrow \infty$  punktweise gegen die Verteilungsfunktion der Standardnormalverteilung. Genauer gilt:

$$\lim_{n \rightarrow \infty} P\left(\frac{Y_n - n\mu}{\sigma\sqrt{n}} \leq x\right) = F(x)$$

für alle  $x \in \mathbb{R}$  und  $F(x)$  ist die Verteilungsfunktion der Standardnormalverteilung.

Das nun für große  $n$  mit hoher Wahrscheinlichkeit Formel (1.19) stimmt, lässt sich über zwei Abschätzungen und dem zentralen Grenzwertsatz zeigen. Dafür geben wir uns ein  $\epsilon > 0$  vor und betrachten die Wahrscheinlichkeit, dass die Abweichung von  $\bar{Y}_n$  vom Erwartungswert im  $\epsilon$ -Intervall liegt. Es gilt:

$$(i) \quad P(\bar{Y}_n - \mu \leq \epsilon) = P(Y_n - n\mu \leq n\epsilon) = P\left(\frac{Y_n - n\mu}{\sigma\sqrt{n}} \leq \frac{\sqrt{n}\epsilon}{\sigma}\right) \approx F\left(\frac{\sqrt{n}\epsilon}{\sigma}\right) \rightarrow 1$$

$$(ii) \quad P(\bar{Y}_n - \mu \leq -\epsilon) = P(Y_n - n\mu \leq -n\epsilon) = P\left(\frac{Y_n - n\mu}{\sigma\sqrt{n}} \leq -\frac{\sqrt{n}\epsilon}{\sigma}\right) \approx F\left(-\frac{\sqrt{n}\epsilon}{\sigma}\right) \rightarrow 0$$

Wobei die Konvergenzaussage am Ende gerade für  $n \rightarrow \infty$  gilt und damit für sehr große  $n$  angenommen werden kann, dass  $\bar{Y}_n$  den Erwartungswert  $\mu$  approximiert.

Die Größe  $\bar{Y}_n$  bezeichnet man auch als Monte-Carlo Schätzer und obwohl wir wissen, dass er im Mittel richtig liegt, haben wir noch keine Aussage über die Höhe des Fehlers, oder der Konvergenzgeschwindigkeit. Dazu betrachtet man die Varianz der Abweichung zwischen dem MC Schätzer und  $\mu$ .

$$\text{Var}(\bar{Y}_n - \mu) = \text{Var}(\bar{Y}_n) = \frac{1}{n^2} \sum_{i=1}^n \text{Var}(X_i) = \frac{\sigma^2}{n}$$

Die letzten beiden Gleichheitszeichen gelten aufgrund der u.i.v. Eigenschaft der Zufallsvariablen  $X_i$ . Die Standardabweichung hat also die Ordnung  $\mathcal{O}(1/\sqrt{n})$ , was bedeutet, dass um eine Nachkommastelle an Genauigkeit zu gewinnen, man einhundertmal mehr Simulationen benötigt. Um die Konvergenzgeschwindigkeit der Monte-Carlo Methode zu erhöhen, gibt es verschiedene Ansätze zur Varianzreduktion, wie zum Beispiel *stratified sampling*, oder die Verwendung von *antithetischen Zufallsvariablen*. Auf letzteres wollen wir hier etwas genauer eingehen, da diese Methode später im Zusammenhang mit dem multi-level Monte-Carlo Verfahren und den bereits erwähnten Lévy Areas noch einmal relevant wird.

Sei  $X$  die durch MC-Simulationen zu ermittelnde Zufallsvariable mit  $X = f(Z)$ , wobei die Abbildung  $f : \mathbb{R} \rightarrow \mathbb{R}$  zum Beispiel die Auszahlungsfunktion einer Option ist und  $Z$  eine standardnormalverteilte Zufallsvariable, also  $Z \sim N(0,1)$ . Ziel ist es nun die Funktion  $f$  durch eine neue Funktion  $\hat{f}$  zu ersetzen, für die  $E(\hat{f}(Z)) = E(f(Z))$  und  $\text{Var}(\hat{f}(Z)) < \text{Var}(f(Z))$  gilt. In [1] Kapitel 5 wird gezeigt, dass die Funktion

$$\hat{f}(Z) = \frac{f(Z) + f(-Z)}{2} \quad (1.20)$$

diese Bedingungen erfüllt. Genauer lässt sich zeigen, dass das Ersetzen des MC Schätzers

$$\frac{1}{n} \sum_{i=1}^n f(Z(\omega_i))$$

durch den neuen Schätzer

$$\frac{1}{n} \sum_{i=1}^n \frac{f(Z(\omega_i)) + f(-Z(\omega_i))}{2}$$

mindestens eine Halbierung der Varianz zur Folge hat. Es gilt also  $E(\hat{f}(Z)) = E(f(Z))$  und  $\text{Var}(\hat{f}(Z)) \leq \text{Var}(f(Z))/2$ .

Diese recht simple Methode erlaubt bereits eine wesentliche Verbesserung der einfachen Monte-Carlo Methode wie in dem Beispiel am Ende des Kapitels gezeigt wird. Dem großen Nachteil der MC Methode – die langsame Konvergenz – kann damit etwas entgegengewirkt werden, als besonders effizient kann die Monte-Carlo Methode dennoch nicht bezeichnet werden. Die numerische Umsetzung sowie eine Gesamtfehleranalyse soll im Folgenden etwas genauer betrachtet werden.

### 1.3.2 Numerische Umsetzung und Analyse

Mit Hilfe der bisher gewonnenen Ergebnisse ist es nun möglich Schritt (2) und (3) des Algorithmus 1.5 umzusetzen. Für die Bewertung einer europäischen Option basierend auf dem Modell der geometrischen Brownschen Bewegung gilt dann zum Beispiel:

$$(2) \text{ Setze } S(T) = S_0 \exp\left(\left(\mu - \frac{1}{2}\sigma^2\right)T + \sigma W(T)\right) \text{ mit } W(T) \sim N(0, T)$$

(3) Simuliere  $n$  Kurswerte  $S(T, \omega_1), \dots, S(T, \omega_n)$  numerisch und approximiere den Optionswert als den abgezinsten Mittelwert

$$V(t, S(t)) = e^{r(t-T)} \frac{1}{n} \sum_{i=1}^n V(T, S(T, \omega_i))$$

Für die Formel  $V(T, S)$  wird entweder Formel (1.1) für den europäischen Call, bzw. Formel (1.2) für den Put verwendet. Die benötigten Zufallsvariablen  $W(T)$  werden am Computer gezogen, in MATLAB zum Beispiel über den `randn`-Befehl. Bei der Verwendung von antithetischen Zufallsvariablen ändert sich der obige Algorithmus geringfügig ab. Einerseits nutzen wir aus, dass  $N(0, T)$ -verteilte Zufallsvariablen auch als  $\sqrt{T}Z$  mit  $Z \sim N(0, 1)$  geschrieben werden können, wodurch die Abbildung  $f$  zum Beispiel die Form

$$f(Z) = \max\left(S_0 \exp\left(\left(\mu - \frac{1}{2}\sigma^2\right)T + \sigma\sqrt{T}Z\right) - K, 0\right)$$

für den europäischen Call annimmt. In Schritt (3) des Algorithmus wird dann wie beschrieben der Monte-Carlo Schätzer ausgetauscht:

(3) Simuliere die Ergebnisse für  $f(Z(\omega_i))$  und  $f(-Z(\omega_i))$  numerisch und approximiere den Optionswert durch den abgezinsten Wert

$$V(t, S(t)) = e^{r(t-T)} \frac{1}{n} \sum_{i=1}^n \frac{f(Z(\omega_i)) + f(-Z(\omega_i))}{2}$$

Im zweiten Schritt des Algorithmus müsste normalerweise ein Verfahren zur numerischen Approximation der verwendeten stochastischen Differentialgleichung eingesetzt werden, im Fall der geometrischen Brownschen Bewegung lässt sich das vermeiden, da diese analytisch lösbar ist. Im Allgemeinen lässt sich das für komplexere Modelle, wie etwa dem 2-dimensionalen Heston Modell, nicht annehmen. Zudem gilt obiger Algorithmus nur für die Bewertung von europäischen Optionen, da in der Auszahlungsfunktion nur der Wert bei Fälligkeit eine Rolle spielt. Bei anderen

Derivaten wie etwa der asiatischen Option sind wir allerdings auf die Kenntnis des Pfades angewiesen. In den Bewertungsbeispielen am Ende des Kapitels wird daher im zweiten Schritt des Algorithmus das Euler-Verfahren zum Einsatz kommen.

Da sowohl die Monte-Carlo Methode, wie auch jedes numerische Approximationsverfahren nicht einhundert Prozent genau ist, stellt sich die Frage wie groß der Gesamtfehler ist. Zudem gibt es zwei Möglichkeiten diesen zu verkleinern: Zum einen kann die Schrittweite  $h$  des Euler-Verfahrens in Schritt (2) verringert werden, andererseits verringert eine Erhöhung der Anzahl der Simulationen  $n$  den Monte-Carlo Fehler in Schritt (3). Beide Varianten verursachen jedoch zusätzlichen Rechenaufwand und können daher nicht beliebig hoch angesetzt werden. Bevor wir das optimale Verhältnis von Schrittweite  $h$  und Simulationsanzahl  $n$  untersuchen, führen wir eine Fehlermaß ein, um den Gesamtfehler in Abhängigkeit der Genauigkeit zu ermitteln. Eine mögliche Wahl eines Fehlermaßes dafür ist der *mean squared error* (MSE), oder zu Deutsch *mittlere quadratische Abweichung*.

**Definition 1.20**

Sei  $X$  eine Zufallsvariable und  $\tilde{Y}$  eine messbare Funktion dieser Variablen. Ist  $\tilde{Y}$  ein Schätzer für den wahren Parameter  $Y$ , dann ist die mittlere quadratische Abweichung (MSE) definiert als

$$MSE(\tilde{Y}, Y) = E[(\tilde{Y} - Y)^2].$$

Der *root mean squared error* (RSME) ist dann gegeben durch

$$RMSE(\tilde{Y}, Y) = \sqrt{E[(\tilde{Y} - Y)^2]}.$$

**Bemerkung 1.21**

Der MSE gibt den erwarteten quadratischen Fehler des Schätzers zum realen Wert an. Mittels des Verschiebungssatzes aus der Stochastik lässt sich der Fehler in zwei Terme aufspalten und erlaubt somit den Gesamtfehler genauer den einzelnen Fehlerursachen zuzuordnen.

$$\begin{aligned} MSE(\tilde{Y}) &= E[(\tilde{Y} - Y)^2] \\ &= Var(\tilde{Y} - Y) + E[(\tilde{Y} - Y)]^2 \\ &= Var(\tilde{Y}) + E[(\tilde{Y} - Y)]^2 \end{aligned}$$

Der hintere Term  $E[(\tilde{Y} - Y)]$  wird auch als *Verzerrung*, oder *Bias* bezeichnet und wird mit  $Bias(\tilde{Y}, Y)$  notiert. Bei einem erwartungstreuen Schätzer, d.h. der Erwartungswert des Schätzers ist gleich dem wahren Wert des zu schätzenden Parameters, ist der Bias null und der MSE ist gleich mit der Varianz des Schätzers.

Unter Verwendung eines einfachen Monte-Carlo Ansatzes mit  $n$  Simulationen und in Kombination mit dem stochastischen Euler-Verfahren mit äquidistanter Schrittweite  $h$  gilt dann für den ersten Summanden des MSE:

$$\begin{aligned} \text{Var}(\tilde{Y}) &= \text{Var}\left(\frac{1}{n}\sum_{i=1}^n f\left(\tilde{S}_{T/h}^{(i)}\right)\right) \\ &= \frac{1}{n^2}\sum_{i=1}^n \text{Var}\left(f\left(\tilde{S}_{T/h}^{(i)}\right)\right) \\ &\leq c_1 n^{-1} \end{aligned}$$

Wobei  $f$  eine Auszahlungsfunktion in Abhängigkeit eines Basiswerts am Laufzeitende ist. Setzt man die Konstante  $c_1$  gleich der Varianz einer einzigen Monte-Carlo Simulation  $f(\tilde{S}_{T/h}^{(i)})$ , erhält man Gleichheit.

Für den zweiten Term, den quadratischen Bias, folgt aus der schwachen Approximationsordnung  $\beta = 1$  des stochastischen Euler-Verfahrens sofort

$$E[(\tilde{Y} - Y)]^2 \leq c_2 h^2$$

für ein  $c_2 \in \mathbb{R}$  und einer hinreichend kleinen Schrittweite  $h$ . Für den Gesamtfehler gilt demnach:

$$MSE(\tilde{Y}) \approx c_1 n^{-1} + c_2 h^2 \quad (1.21)$$

Es ist leicht zu sehen, dass der erste Fehlerterm gerade aus der Verwendung der Monte-Carlo Methode zur Schätzung des Erwartungswertes stammt und durch die Erhöhung der Simulationen verringert werden kann. Der zweite Term geht aus dem stochastischen Euler-Verfahren hervor und lässt sich durch eine kleinere Schrittweite verbessern. Setzt man nun fest, dass der MSE insgesamt die Ordnung  $\epsilon^2$  haben soll, folgt sofort die Ordnung  $\epsilon^{-2}$  für  $n$ , sowie  $\mathcal{O}(\epsilon)$  für die Schrittweite. Dass dieses quadratische Verhältnis zwischen Anzahl an Simulationen und Schrittweite tatsächlich optimal bezüglich des Rechenaufwandes ist, wurde 1995 im Paper „Efficient Monte Carlo Simulation of Security Prices“ von D. Duffie und P. Glynn gezeigt ([8]).

Für den Gesamtrechenaufwand ergibt sich bei  $\mathcal{O}(\epsilon^{-2})$  Simulationen und einer Schrittweite  $h$  der Ordnung  $\epsilon$  – und damit  $\mathcal{O}(\epsilon^{-1})$  Schritten – die Ordnung  $\epsilon^{-3}$ .

### 1.3.3 Beispiele

Zum Abschluss des Kapitels sollen die bisher gezeigten Methoden und Ergebnisse verwendet werden um eine europäische Call-Option zu bewerten. Die Simulation des Basiswerts erfolgt über die geometrische Brownsche Bewegung aus Definition 1.9 mit den konstanten Parametern  $\mu$  und

$\sigma$ . Um einen realistischeren Vergleich zu komplexeren Modellen – deren analytische Lösungen im Allgemeinen nicht bekannt sind – zu ermöglichen wird anstatt der Gleichung

$$S(T) = S_0 \exp\left(\left(\mu - \frac{1}{2}\sigma^2\right)T + \sigma W(T)\right)$$

das stochastische Euler-Verfahren mit Schrittweite  $h$  für die Approximation des Wertes  $S(T)$  eingesetzt. Für die Schätzung des Erwartungswertes werden  $1/h^2$  einfache Monte-Carlo Simulationen verwendet und das Ergebnis in die risikoneutrale Bewertungsformel eingesetzt.

Das unter den gegebenen Marktannahmen und Basiswertsimulation tatsächlich auch eine analytische Lösung für die Bewertung einer europäischen Option existiert, wurde von F. Black, M. Scholes und R. Merton in den 1970er Jahren gezeigt. Dazu wird die als *Black-Scholes Gleichung* bekannte partielle DGL verwendet:

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} - rV + rS \frac{\partial V}{\partial S} = 0$$

Die partielle DGL wird vom Optionswert  $V$  erfüllt und besitzt im Allgemeinen unendlich viele Lösungen. Durch die Annahme von Randwerten ergibt sich jedoch eine eindeutige Lösung, für die im Fall von einer Call-Option gilt

$$V_{Ce}(t, S) = SP(a) - Ke^{-r(T-t)}P(b).$$

Wobei  $P$  die Verteilungsfunktion der Standard-Normalverteilung ist und  $a$  gegeben ist durch

$$a = \frac{\log \frac{S}{K} + \left(r + \frac{\sigma^2}{2}\right)(T-t)}{\sigma\sqrt{T-t}}$$

und  $b = a - \sigma\sqrt{T-t}$  gilt.

Die Black-Scholes Gleichung wird verwendet um den wahren Optionswert im folgenden Beispiel zu berechnen. Je höher die Anzahl der Monte-Carlo Simulationen und je kleiner die Schrittweite, desto näher sollten die Ergebnisse im Mittel am Black-Scholes Wert liegen.

Es gelten die folgenden Beispieldaten:

- $\mu = r = 0.05$
- $\sigma = 0.2$
- $h = 0.1, 0.01, 0.001, 0.0001$
- $n = 1/h^2$
- $T = 1$
- $K = 1$
- $S_0 = 1$

Für den wahren Optionswert nach Black-Scholes ergibt sich in diesem Fall gerundet auf vier Nachkommastellen 0.1045. Da die Schätzung über die Monte-Carlo Methode auf Zufall beruht, kann natürlich nicht davon ausgegangen werden, dass mehr Simulationen *immer* zu einem genaueren Ergebnis führen. Um zumindest ein Gefühl für die Güte der Schätzung zu bekommen, sind in folgender Tabelle für die verschiedenen Schrittweiten exemplarisch vier unabhängig ermittelte Monte-Carlo Ergebnisse abgetragen.

<b>Ergebnis</b> <b>Methode</b>	<b>Lauf 1</b>	<b>Lauf 2</b>	<b>Lauf 3</b>	<b>Lauf 4</b>	<b>#Funktionsauswertungen</b>
<b>MC</b> $h = 4^{-1}$	0.0583	0.1386	0.1727	0.0710	64
<b>MC</b> $h = 4^{-2}$	0.0926	0.1167	0.1177	0.1063	4.096
<b>MC</b> $h = 4^{-3}$	0.1067	0.1065	0.1033	0.1049	262.144
<b>MC</b> $h = 4^{-4}$	0.1040	0.1040	0.1049	0.1038	16.777.216

Tabelle 1.1: Bewertungsergebnisse einer europäischen Option mittels der MC Methode

Die Spalte „#Funktionsauswertungen“ in Tabelle 1.1 gibt die Gesamtanzahl der berechneten Punkte innerhalb des stochastischen Euler-Verfahrens wieder und entspricht gerade  $h^{-3}$ . Mit kleiner werdender Schrittweite und damit quadratisch ansteigender Simulationszahl, nähern sich die Ergebnisse wie vermutet dem Black-Scholes Wert von 0.1045 an, aber es lässt sich auch die langsame Konvergenzgeschwindigkeit der Monte-Carlo Methode erkennen. So lässt sich nur eine relativ geringe Verbesserung zwischen den letzten beiden Versuchsreihen feststellen, obwohl die Verringerung der Schrittweite von  $1/64$  auf  $1/256$  und die damit verbundene Erhöhung der Simulationen von 4096 auf 65536 zu einem erheblichen Anstieg des Rechenaufwands führt.

Als abschließendes Beispiel wollen wir noch die zur Varianzreduktion vorgestellte Methode der antithetischen Zufallsvariablen vergleichen mit der einfachen Monte-Carlo Methode. Dazu werden die in Kapitel 1.3.2 vorgestellten Algorithmen verwendet, d.h. der Basiswert  $S(T)$  am Laufzeitende wird über die analytische Lösung der geometrischen Brownschen Bewegung ermittelt und nicht über das stochastische Euler-Verfahren. Dies führt dazu, dass die Verwendung von antithetischen Zufallsvariablen zwar den Rechenaufwand erhöht, da ja neben  $f(Z)$  auch  $f(-Z)$  berechnet werden muss, der Mehraufwand sich aber vernachlässigen lässt, da die Anzahl der Rechenoperationen sich höchstens verdoppeln würde und damit Ordnungstechnisch gleich bleibt. Die Werte für  $\sigma, \mu, T, K$  und  $S_0$  entsprechen wieder den obigen Werten.

<b>Ergebnis</b> <b>Methode</b>	<b>Lauf 1</b>	<b>Lauf 2</b>	<b>Lauf 3</b>	<b>Lauf 4</b>	<b>#Simulationen</b>
MC (Standard)	0.1122	0.1295	0.1148	0.0820	100
MC (Standard)	0.1070	0.0984	0.1098	0.1094	1.000
MC (Standard)	0.1056	0.1052	0.1023	0.1061	10.000
MC (Standard)	0.1048	0.1040	0.1048	0.1043	100.000
MC (Standard)	0.1043	0.1044	0.1045	0.1043	1.000.000
MC (antithetisch)	0.0964	0.1048	0.1133	0.1200	100
MC (antithetisch)	0.1059	0.1039	0.1049	0.0998	1.000
MC (antithetisch)	0.1042	0.1019	0.1052	0.1056	10.000
MC (antithetisch)	0.1044	0.1042	0.1045	0.1044	100.000
MC (antithetisch)	0.1045	0.1045	0.1044	0.1045	1.000.000

Tabelle 1.2: Monte-Carlo Ergebnisse mit und ohne antithetische Zufallsvariablen

Die Ergebnisse untermauern die theoretische Analyse, dass die Verwendung von antithetischen Zufallsvariablen eine schnellere Konvergenz zur Folge hat. Mit einer Millionen Monte-Carlo Simulationen konnte das gewünschte Ergebnis 0.1045 fast immer getroffen werden, allerdings gilt es zu bemerken, dass die sehr hohe Simulationsanzahl sich nur aufgrund des verwendeten stochastischen Modells und der bekannten analytischen Lösung in akzeptabler Zeit durchführen lässt.

## 2 Die multi-level Monte-Carlo Methode

Im vorherigen Kapitel wurde gezeigt, wie die Entwicklung des Basiswerts mit Hilfe von stochastischen Differentialgleichungen simuliert werden kann und anschließend der Erwartungswert durch die Monte-Carlo Methode ermittelt wird. Aufgrund des recht allgemeinen Ansatzes des Monte-Carlo Verfahrens ist ein modellunabhängiger und damit flexibler Einsatz möglich. In Sachen Konvergenzgeschwindigkeit konnte die Methode jedoch wenig überzeugen, in Kombination mit dem stochastischen Euler-Verfahren ergab sich etwa eine Ordnung des Rechenaufwands von  $\mathcal{O}(\epsilon^{-3})$  bei einem MSE von  $\mathcal{O}(\epsilon^{-2})$ .

In diesem Kapitel soll gezeigt werden, dass durch einen von Michael B. Giles eingeführten multi-level Ansatz eine Verbesserung der Konvergenzordnung bei gleichbleibendem MSE möglich ist. Dazu werden wir die multi-level Monte-Carlo Methode (MLMC) zuerst in Kombination mit dem stochastischen Euler-Verfahren und einer Lipschitz-stetigen Auszahlungsfunktion herleiten und zeigen, dass der Rechenaufwand so auf  $\mathcal{O}(\epsilon^{-2} \ln(\epsilon)^2)$  verbessert werden kann. Eine wesentlich allgemeinere Formulierung der MLMC Methode wird im sogenannten Komplexitätstheorem festgehalten und bewiesen. Anschließend werden einige der auftretenden Parameter für eine genauere Analyse herangezogen und mögliche Erweiterungen der Methode vorgestellt. Zum Abschluss des Kapitels wird genauer auf die numerische Implementierung eingegangen und anhand von Beispielen die verschiedenen Vorteile der Methode aufgezeigt. Die hier vorgestellten Ergebnisse beruhen größtenteils auf dem Paper „Multilevel Monte-Carlo Path Simulation“ von Michael B. Giles ([9]).

### 2.1 Herleitung der MLMC Methode

Die Grundidee der multi-level Monte-Carlo Methode ist ähnlich zu der des Mehrgitterverfahrens (engl. multigrid method), das in der Numerik zur Lösung von Gleichungssystemen eingesetzt wird. Um das Gleichungssystem iterativ lösen zu können, muss das Problem diskretisiert werden, wobei durch die Diskretisierungspunkte ein Gitter entsteht, auf dem die Lösung approximiert wird. Anstatt ein möglichst feines Gitter zu verwenden, was zwar eine hohe Genauigkeit, aber auch einen hohen Rechenaufwand zur Folge hat, wird das Gleichungssystem auf verschiedenen feinen Gittern gelöst. Dabei wird typischerweise das Gitter der nächsten Stufe zweimal so fein wie das der vorherigen Stufe gewählt. Die ungenaueste Lösung auf dem größten Gitter lässt sich als erste Schätzung interpretieren und die nachfolgenden Lösungen auf den feineren Gittern als Korrekturterme. Insgesamt lässt sich damit ein Ergebnis erzielen, dass von der Qualität der Lösung dem feinen Gitter entspricht, aber einen deutlich geringeren Rechenaufwand erfordert.

Auf die Monte-Carlo Methode lässt sich diese Idee übertragen, indem im numerischen Approximationsverfahren für die SDGL immer kleiner werdende Schrittweiten verwendet werden. Für die Herleitung der MLMC Methode wird die folgende Notation verwendet.

**Notation 2.1**

Sei  $P$  der Payoff einer Lipschitz-stetigen Auszahlungsfunktion  $f(S_T)$ , d.h.  $f(S_T) = P$  und  $h_l$  die Schrittweite des stochastischen Euler-Verfahrens wobei gilt

$$h_l = M^{-l}T$$

für  $l = 0, 1, \dots, L$  und ein  $M \in \mathbb{N}_{\geq 2}$ .

Seien weiterhin  $\tilde{S}_{l,M^l}$  und  $\tilde{P}_l$  die Approximationen von  $S(T)$  bzw.  $P$  auf dem  $l$ -ten Level, das heißt unter Verwendung der Schrittweite  $h_l$ .

Es lässt sich leicht sehen, dass aufgrund der Linearität des Erwartungswertes die Gleichung

$$E[\tilde{P}_L] = E[\tilde{P}_0] + \sum_{l=1}^L E[\tilde{P}_l - \tilde{P}_{l-1}] \quad (2.1)$$

gilt. In der einfachen Monte-Carlo Methode wird der Term auf der linken Seite geschätzt, also die erwartete Auszahlung unter Verwendung einer möglichst kleinen Schrittweite. Die multi-level Monte-Carlo Methode dagegen schätzt alle Terme der rechten Teleskopsumme ab, wobei man – ähnlich wie beim Mehrgitterverfahren – den ersten Summanden  $E[\tilde{P}_0]$  als Initialschätzung und die folgenden  $L$  Summanden als Korrekturterme interpretieren kann. Für die Schätzung der Terme  $E[\tilde{P}_l - \tilde{P}_{l-1}]$ , welche gerade der erwarteten Abweichung des Payoffs unter Verwendung verschiedener Schrittweiten entspricht, wird die einfache MC Methode eingesetzt.

Sei dazu  $\tilde{Y}_0$  Schätzer für  $E[\tilde{P}_0]$  unter Verwendung von  $N_0$  Simulationen und  $\tilde{Y}_l$  Schätzer für  $E[\tilde{P}_l - \tilde{P}_{l-1}]$  mittels  $N_l$  Simulationen, d.h.:

$$\tilde{Y}_l = \frac{1}{N_l} \sum_{i=1}^{N_l} (\tilde{P}_l^{(i)} - \tilde{P}_{l-1}^{(i)}).$$

Bei der eigentlichen Implementierung des Verfahrens ist hier zu beachten, dass für die  $i$ -te Monte-Carlo Simulation für den Wert  $\tilde{P}_l^{(i)} - \tilde{P}_{l-1}^{(i)}$  zwar eine unterschiedliche Schrittweite eingesetzt wird, der verwendete Wiener-Prozess jedoch der Gleiche ist. Dazu zieht man zuerst die  $M^l$  Zufallszahlen für den Pfad des  $l$ -ten Levels und summiert diese dann in Gruppen der Größe  $M$  auf für die Approximation auf dem Level  $l - 1$ . Siehe hierzu auch Abbildung 2.1, in der ein Wiener-Prozess  $W_l$  und der korrespondierende Pfad des Prozesses  $W_{l-1}$  für  $M = 3$  abgebildet sind.

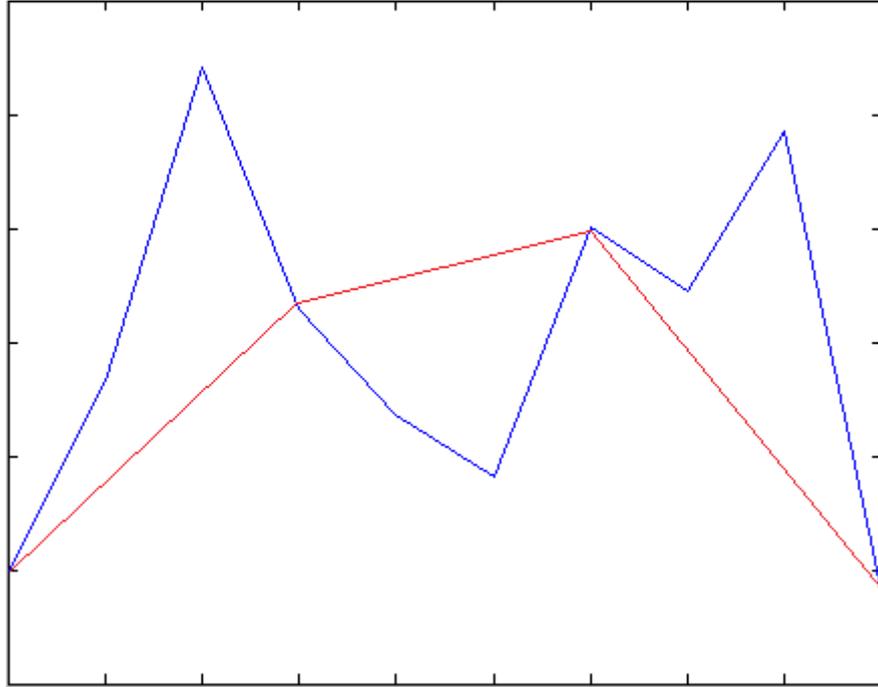


Abbildung 2.1: Wiener-Prozess  $W_l$  in blau und  $W_{l-1}$  in rot für  $M = 3$

Als erstes gilt es zu zeigen, dass der MSE der MLMC Methode die Ordnung  $\mathcal{O}(\epsilon^2)$  besitzt. Wie vorher gilt es einerseits die Varianz des Gesamtschätzers  $\tilde{Y}$  und andererseits, den quadratischen Bias  $E[(\tilde{Y} - Y)]^2$  zu untersuchen. Für die Varianz eines Korrekturterms gilt:

$$\begin{aligned} \text{Var}(\tilde{Y}_l) &= \text{Var}\left(\frac{1}{N_l} \sum_{i=1}^{N_l} (\tilde{P}_l^{(i)} - \tilde{P}_{l-1}^{(i)})\right) \\ &= \frac{1}{N_l^2} \text{Var}\left(\sum_{i=1}^{N_l} (\tilde{P}_l^{(i)} - \tilde{P}_{l-1}^{(i)})\right) \\ &= \frac{1}{N_l} \underbrace{\text{Var}(\tilde{P}_l^{(i)} - \tilde{P}_{l-1}^{(i)})}_{=: V_l} \end{aligned}$$

Das Ergebnis entspricht natürlich dem der einfachen Monte-Carlo Methode – Varianz einer Simulation geteilt durch die Anzahl der Simulationen – da diese ja gerade eingesetzt wird um die erwartete Differenz von  $\tilde{P}_l$  und  $\tilde{P}_{l-1}$  zu schätzen.

Für die Varianz des Gesamtschätzers  $\tilde{Y}$  folgt dann aufgrund der Unabhängigkeit der Einzelvarianzen

$$\text{Var}(\tilde{Y}) = \text{Var}\left(\sum_{l=0}^L \tilde{Y}_l\right) = \sum_{l=0}^L \text{Var}(\tilde{Y}_l) = \sum_{l=0}^L N_l^{-1} V_l. \quad (2.2)$$

Zu beachten ist hier, dass für die Schätzung von  $\tilde{Y}_l$  zwar die gleichen Brownschen Pfade verwendet werden für  $\tilde{P}_l^{(i)}$  und  $\tilde{P}_{l-1}^{(i)}$ , diese jedoch nicht in die Schätzung für  $\tilde{Y}_{l-1}$  oder  $\tilde{Y}_{l+1}$  eingehen. Für jeden Schätzer der einzelnen Summanden auf der rechten Seite der Gleichung (2.1) werden neue, unabhängige Pfade erzeugt.

Um eine Aussage über die Varianz treffen zu können, muss zuerst die optimale Anzahl an Simulationen auf jedem Level, sowie die Varianz einer Simulation bestimmt werden. Für die Varianz  $V_l$  gilt unter Beachtung der Rechenregel

$$\text{Var}(X + Y) = \text{Var}(X) + \text{Var}(Y) + 2\text{Cov}(X, Y)$$

die Abschätzung:

$$\begin{aligned} V_l &= \text{Var}(\tilde{P}_l - \tilde{P}_{l-1}) \\ &= \text{Var}(\tilde{P}_l - P + P - \tilde{P}_{l-1}) \\ &\leq \text{Var}(\tilde{P}_l - P) + \text{Var}(P - \tilde{P}_{l-1}) + 2|\text{Cov}(\tilde{P}_l - P, P - \tilde{P}_{l-1})| \end{aligned}$$

Für den Betrag der Kovarianz gilt im Allgemeinen

$$|\text{Cov}(X, Y)| \leq \text{Var}(X)^{1/2} \text{Var}(Y)^{1/2}$$

und damit

$$\begin{aligned} V_l &\leq \text{Var}(\tilde{P}_l - P) + \text{Var}(P - \tilde{P}_{l-1}) + 2 \left( \text{Var}(\tilde{P}_l - P)^{1/2} \text{Var}(P - \tilde{P}_{l-1})^{1/2} \right) \\ &= \left( \text{Var}(\tilde{P}_l - P)^{1/2} + \text{Var}(P - \tilde{P}_{l-1})^{1/2} \right)^2 \end{aligned}$$

Angenommen der Term  $\text{Var}(\tilde{P}_l - P)$  besitzt die Ordnung  $\mathcal{O}(h_l)$ , dann folgt mit der Gleichung  $h_l M = h_{l-1}$  sofort

$$V_l \leq \left( (c_1 h_l)^{1/2} + (c_1 h_{l-1})^{1/2} \right)^2$$

und somit  $V_l = \mathcal{O}(h_l)$ .

Es bleibt zu zeigen, dass tatsächlich  $\text{Var}(\tilde{P}_l - P)$  ein Term der Ordnung  $\mathcal{O}(h_l)$  ist. Aus dem Verschiebungssatz der Stochastik folgt die Ungleichung

$$\text{Var}(\tilde{P}_l - P) \leq E \left[ (\tilde{P}_l - P)^2 \right].$$

Verwendet man nun die Lipschitz-Eigenschaft der Payoff-Funktion  $f$  (und damit  $P$ ) folgt

$$\text{Var}(\tilde{P}_l - P) \leq E \left[ (\tilde{P}_l - P)^2 \right] \leq c_1^2 E \left[ \|\tilde{S}_{l,M^l} - S(T)\|^2 \right] \leq c_2 h_l.$$

Für die letzte Abschätzung verwendet man zudem die Ungleichung (1.16) aus Satz 1.15 (ii). Die Varianz einer Monte-Carlo Simulation ist also von der Ordnung  $\mathcal{O}(h_l)$ . Um die Varianz des Gesamtschätzers zu bestimmen, fehlt nun nur noch eine Aussage über die asymptotisch optimale Anzahl an Monte-Carlo Simulationen auf jedem Level. Optimal in dem Sinne, dass die Varianz  $\tilde{Y}$  minimiert werden soll, unter der Nebenbedingung eines festen Rechenaufwandes. Da auf jedem Level  $l \geq 1$  sowohl  $M^l$  Schritte für die feinere, als auch  $M^{l-1}$  Schritte für die gröbere Schrittweite berechnet werden müssen, ergibt sich der Gesamtrechenaufwand  $C$  zu

$$C = N_0 + \sum_{l=1}^L N_l (h_l^{-1} + h_{l-1}^{-1}). \quad (2.3)$$

Für das Minimierungsproblem wird ein Lagrange-Ansatz verwendet, wobei man den kleineren Summanden  $N_l h_{l-1}^{-1}$  vernachlässigt, womit der Rechenaufwand etwa proportional zu

$$C = \sum_{l=0}^L N_l h_l^{-1} \quad (2.4)$$

wird. Sei  $N = (N_0, N_1, \dots, N_L)^T$ ,  $h = (h_0, h_1, \dots, h_L)^T$  und  $g(N) = \sum_{l=0}^L N_l^{-1} V_l$  die zu minimierende Funktion unter der Nebenbedingungsgleichung (2.4). Es gilt

$$L(N) = g(N) + \lambda \left( \sum_{l=0}^L N_l h_l^{-1} - C \right)$$

und damit für die gleich null gesetzten partiellen ersten Ableitungen

$$0 \stackrel{!}{=} \frac{\partial L(N)}{\partial N_l} = -N_l^{-2} V_l + \lambda h_l^{-1}$$

für  $l = 0, 1, \dots, L$ . Auflösen nach  $N_l$  ergibt:

$$N_l = \sqrt{\lambda^{-1} V_l h_l}$$

Für die partielle Ableitung nach  $\lambda$  rechnet man

$$0 \stackrel{!}{=} \frac{\partial L(N)}{\partial \lambda} = \sum_{l=0}^L N_l h_l^{-1} - C = \sum_{l=0}^L \sqrt{\lambda^{-1} V_l h_l} h_l^{-1} - C$$

und somit

$$\lambda = \left( \frac{1}{C} \sum_{l=0}^L \sqrt{V_l h_l^{-1}} \right)^2.$$

Aus dem Lagrange-Ansatz lässt sich folgern, dass  $N_l$  proportional zu  $\sqrt{V_l h_l}$  gesetzt werden muss und damit  $\mathcal{O}(h_l)$  ist, da ja gerade  $V_l = \mathcal{O}(h_l)$  gilt. Setzt man die Anzahl der MC Simulationen als

$$N_l^* = c_1 \epsilon^{-2} L h_l \quad (2.5)$$

für ein  $c_1 \in \mathbb{R}$  folgt für die Varianz des Gesamtschätzers

$$\text{Var}(\tilde{Y}) = \sum_{l=0}^L N_l^{-1} V_l = \sum_{l=0}^L \frac{c_2 h_l}{c_1 \epsilon^{-2} L h_l} = (1 + L) \frac{c_2}{c_1 L} \epsilon^2$$

und damit die gewünschte Ordnung  $\epsilon^2$ , sofern die Wahl von  $L$  diese nicht erhöht. Konkret wählen wir hier für die Anzahl der MC Levels

$$L = -\frac{\ln(\epsilon)}{\ln(M)} + c_3, \quad (2.6)$$

welche die Ordnung für ein kleiner werdendes  $\epsilon$  nicht erhöht.

Für die Wahl (2.6) der MC Levels folgt dann für den einfachen Bias

$$E(\tilde{P}_L - P) \leq c_4 h_L = c_4 T M^{-L} = c_5 M^{\ln(\epsilon)/\ln(M)} M^{-c_3}.$$

Zusammenfassen, sowie ein Basiswechsel des Logarithmus zur Basis  $M$ , führt dann zum gewünschten Ergebnis

$$\begin{aligned} E(\tilde{P}_L - P) &\leq c_5 M^{(\log_M(\epsilon)/\log_M(e))/(\log_M(M)/\log_M(e))} M^{-c_3} \\ &= c_6 \epsilon = \mathcal{O}(\epsilon) \end{aligned}$$

Sowohl die Varianz als auch der quadratische Bias haben somit die Ordnung  $\mathcal{O}(\epsilon^2)$  und damit natürlich auch der gesamte MSE. Um den Rechenaufwand des MLMC Verfahrens zu untersuchen, betrachtet man wieder die Gleichung  $C = \sum_{l=0}^L N_l h_l^{-1}$  und setzt die obige Wahl (2.5) und (2.6) von  $N_l^*$  bzw.  $L$  ein.

$$\begin{aligned} C &= \sum_{l=0}^L c_1 \epsilon^{-2} L h_l h_l^{-1} \\ &= (1 + L) c_1 \epsilon^{-2} L \\ &\leq c_2 \epsilon^{-2} (\ln(\epsilon)^2 - \ln(\epsilon)) \end{aligned}$$

Da  $\ln(\epsilon)$  für  $\epsilon \rightarrow 0$  vernachlässigbar gegenüber  $\ln(\epsilon)^2$  ist, ist der Gesamtrechenaufwand der multi-level Monte-Carlo Methode  $\mathcal{O}(\epsilon^{-2} \ln(\epsilon)^2)$ . Bei gleicher Fehlerabschätzung wie bei der einfachen MC Methode erhält man also hier einen wesentlich geringeren Rechenaufwand.

## 2.2 Komplexitätstheorem

Ziel des folgenden Theorems ist es, das eben heuristisch hergeleitete Resultat allgemeiner, also insbesondere unabhängig von der verwendeten numerischen Approximationsmethode, zu formulieren. Der hier vorgestellte Beweis des sogenannten Komplexitätstheorems entspricht einer ausformulierten Version des in [9] zu findenden Vorgehens.

### Theorem 2.2 (Komplexitätstheorem)

Sei  $P$  ein Funktional der Lösung der stochastischen Differentialgleichung

$$dS(t) = a(t, S)dt + b(t, S)dW(t)$$

und  $\tilde{P}_l$  eine diskrete, numerische Approximation unter Verwendung der Schrittweite  $h_l$  mit

$$h_l = M^{-l}T.$$

Existiert ein unabhängiger Schätzer  $\tilde{Y}_l$  basierend auf  $N_l$  Monte-Carlo Simulationen, sowie positive Konstanten  $\alpha \geq \frac{1}{2}, \beta, c_1, c_2, c_3$ , so dass gilt

- (i)  $E[\tilde{P}_l - P] \leq c_1 h_l^\alpha$
- (ii)  $E[\tilde{Y}_l] = \begin{cases} E[\tilde{P}_0], & l = 0 \\ E[\tilde{P}_l - \tilde{P}_{l-1}], & l > 0 \end{cases}$
- (iii)  $Var(\tilde{Y}_l) \leq c_2 N_l^{-1} h_l^\beta$
- (iv)  $C_l \leq c_3 N_l h_l^{-1}$

wobei  $C_l$  eine obere Schranke für den Rechenaufwand des  $l$ -ten Levels darstellt. Dann existiert eine positive Konstante  $c_4$ , so dass es für alle  $\epsilon < e^{-1}$  ein  $L$  und ein  $N_l$  gibt, für die der multi-level Schätzer

$$\tilde{Y} = \sum_{l=0}^L \tilde{Y}_l$$

einen nach oben beschränkten  $MSE$  hat mit

$$MSE \equiv E \left[ (\tilde{Y} - E[P])^2 \right] < \epsilon^2.$$

Weiterhin gilt für den gesamten Rechenaufwand  $C$  die von  $\beta$  abhängige Schranke

$$C \leq \begin{cases} c_4 \epsilon^{-2}, & \beta > 1 \\ c_4 \epsilon^{-2} \ln(\epsilon)^2, & \beta = 1. \\ c_4 \epsilon^{-2-(1-\beta)/\alpha}, & \beta < 1 \end{cases}$$

*Beweis*

Das Vorgehen des Beweises ist ähnlich zur heuristischen Herleitung unter dem stochastischen Euler-Verfahren. Es gilt einerseits die obere Grenze des MSE, also des quadratischen Bias sowie der Varianz des Gesamtschätzers und andererseits den resultierenden Rechenaufwand zu zeigen. Um den quadratischen Bias abzuschätzen, reicht es ein spezielles  $L$  vorzugeben und die Bedingungen (i) und (ii) anzuwenden. Für die Varianz und die Aussage über den Rechenaufwand wird eine Fallunterscheidung nach  $\beta$  durchgeführt, die mit Hilfe einer speziellen Wahl der Simulationsanzahl  $N_l$  jeweils gelöst werden kann.

Sei  $[\cdot]: \mathbb{R} \rightarrow \mathbb{N}$  die Abbildung, die einer reellen Zahl  $x$  der natürlichen Zahl  $n$  zuordnet, welche der Ungleichung  $x \leq n \leq x + 1$  genügt. Sei weiterhin die Levelanzahl  $L$  der multi-level Monte-Carlo Methode gegeben durch die Gleichung

$$L = \left\lceil \frac{\ln(\sqrt{2}c_1 T^\alpha \epsilon^{-1})}{\alpha \ln(M)} \right\rceil.$$

Für den Term  $c_1 h_L^\alpha = c_1 (M^{-LT})^\alpha$  ergibt sich dann einerseits die obere Schranke:

$$\begin{aligned} c_1 h_L^\alpha &= c_1 (M^{-\left\lceil \frac{\ln(\sqrt{2}c_1 T^\alpha \epsilon^{-1})}{\alpha \ln(M)} \right\rceil} T)^\alpha \\ &\leq c_1 M^{-\frac{\alpha \ln(\sqrt{2}c_1 T^\alpha \epsilon^{-1})}{\alpha \ln(M)}} T^\alpha \\ &= c_1 M^{-\frac{\alpha \log_M(\sqrt{2}c_1 T^\alpha \epsilon^{-1}) / \log_M(e)}{\alpha \log_M M / \log_M(e)}} T^\alpha \\ &= c_1 (\sqrt{2}c_1 T^\alpha \epsilon^{-1})^{-1} T^\alpha \\ &= \frac{1}{\sqrt{2}} \epsilon \end{aligned}$$

Sowie die untere Schranke:

$$\begin{aligned} c_1 h_L^\alpha &= c_1 (M^{-\left\lceil \frac{\ln(\sqrt{2}c_1 T^\alpha \epsilon^{-1})}{\alpha \ln(M)} \right\rceil} T)^\alpha \\ &> c_1 M^{-\alpha \left( \frac{\ln(\sqrt{2}c_1 T^\alpha \epsilon^{-1})}{\alpha \ln(M)} + 1 \right)} T^\alpha \\ &= c_1 M^{-\frac{\alpha \log_M(\sqrt{2}c_1 T^\alpha \epsilon^{-1}) / \log_M(e)}{\alpha \log_M M / \log_M(e)}} M^{-\alpha} T^\alpha \\ &= c_1 (\sqrt{2}c_1 T^\alpha \epsilon^{-1})^{-1} M^{-\alpha} T^\alpha \\ &= \frac{1}{\sqrt{2}} M^{-\alpha} \epsilon \end{aligned}$$

Und damit

$$\frac{1}{\sqrt{2}} M^{-\alpha} \epsilon < c_1 h_L^\alpha \leq \frac{1}{\sqrt{2}} \epsilon. \quad (2.7)$$

Nach Bedingung (ii) des Theorems gilt wie am Anfang des Kapitels

$$E(\tilde{Y}) = E\left(\sum_{l=0}^L \tilde{Y}_l\right) = E(\tilde{P}_L).$$

Eingesetzt in Bedingung (i) und unter Verwendung der Abschätzung (2.7) folgt dann:

$$\left(E(\tilde{Y}) - E(P)\right) \leq \frac{1}{\sqrt{2}}\epsilon \Leftrightarrow (E(\tilde{Y}) - P)^2 \leq \frac{1}{2}\epsilon^2$$

Der quadratische Bias besitzt also die obere Grenze  $\epsilon^2/2$  und es bleibt zu zeigen, dass die Varianz von  $\tilde{Y}$  einer ähnlichen Obergrenze unterliegt. Dazu beweisen wir zuerst die Ungleichung

$$\sum_{l=0}^L h_l^{-1} < \frac{M^2}{M-1} (\sqrt{2}c_1)^{1/\alpha} \epsilon^{-2}. \quad (2.8)$$

Für die geometrische Reihe der Schrittweiten erhält man zunächst durch Umformen

$$\sum_{l=0}^L h_l^{-1} = \frac{1}{T} \sum_{l=0}^L M^l = \frac{1}{T} \sum_{l=0}^L M^{L-l} = \frac{M^L}{T} \sum_{l=0}^L M^{-l} < h_L^{-1} \frac{M}{M-1}. \quad (2.9)$$

wobei der letzte Schritt durch den Grenzwert der geometrischen Reihe zu rechtfertigen ist. Die erste Abschätzung in (2.7) und der daraus resultierenden Ungleichung für  $h_L^{-1}$  ergibt zudem

$$h_L^{-1} < M \left( \frac{\epsilon}{\sqrt{2}c_1} \right)^{-1/\alpha}. \quad (2.10)$$

Kombiniert man die obigen Ungleichungen (2.9) und (2.10) folgt

$$\sum_{l=0}^L h_l^{-1} < M \left( \frac{\epsilon}{\sqrt{2}c_1} \right)^{-1/\alpha} \frac{M}{M-1} = \frac{M^2}{M-1} (\sqrt{2}c_1)^{1/\alpha} \epsilon^{-1/\alpha}$$

und damit das gewünschte Ergebnis (2.8), da  $\epsilon^{-1/\alpha} \leq \epsilon^{-2}$  für  $\alpha \geq 1/2$  und  $\epsilon < e^{-1}$ . Diese Ungleichung wird in der folgenden Fallunterscheidung nach  $\beta$  benötigt.

#### Fall 1: $\beta = 1$

Sei die Anzahl der MC Simulationen  $N_l$  gegeben durch

$$N_l := \lceil 2\epsilon^{-2}(L+1)c_2h_l \rceil.$$

Für diese Wahl von  $N_l$  und der Eigenschaft (iii) aus Theorem 2.2 folgt für die Varianz

$$\text{Var}(\tilde{Y}) = \sum_{l=0}^L \text{Var}(\tilde{Y}_l) \leq \sum_{l=0}^L c_2 N_l^{-1} h_l \leq \sum_{l=0}^L c_2 (2\epsilon^{-2}(L+1)c_2 h_l)^{-1} h_l = \frac{1}{2}\epsilon^2$$

und damit die behauptete Größe  $\epsilon^2$  für den MSE.

Um die obere Schranke des Rechenaufwands zu beweisen, zeigt man zuerst

$$L + 1 \leq c_5 \ln(\epsilon^{-1}) \quad (2.11)$$

mit

$$c_5 = \frac{1}{\alpha \ln(M)} + \max\left(0, \frac{\ln(\sqrt{2}c_1 T^\alpha)}{\alpha \ln(M)}\right) + 2.$$

Da  $\epsilon < e^{-1}$  und damit  $\ln(\epsilon^{-1}) > 1$ , gilt für das eingangs angenommene  $L$ :

$$L \leq \frac{\ln(\epsilon^{-1})}{\alpha \ln(M)} + \frac{\ln(\sqrt{2}c_1 T^\alpha)}{\alpha \ln(M)} + 1$$

Und damit folgt Behauptung (2.11) mit Hilfe der Abschätzung

$$\begin{aligned} L + 1 &\leq \frac{\ln \epsilon^{-1}}{\alpha \ln(M)} + \frac{\ln(\sqrt{2}c_1 T^\alpha)}{\alpha \ln(M)} + 2 \\ &\leq \ln \epsilon^{-1} \left( \frac{1}{\alpha \ln(M)} + \max\left(0, \frac{\ln(\sqrt{2}c_1 T^\alpha)}{\alpha \ln(M)}\right) + 2 \right) \\ &= \ln(\epsilon^{-1}) c_5 \end{aligned}$$

Für die obige Wahl der  $N_l$ s, bzw. Annahme (iv) aus Theorem 2.2 ergibt sich durch Einsetzen und Umformen für den Rechenaufwand  $C$  die Abschätzung

$$C \leq c_3 \sum_{l=0}^L N_l h_l^{-1} \leq c_3 \sum_{l=0}^L (2\epsilon^{-2}(L+1)c_2 h_l + 1) h_l^{-1} = c_3 \left( 2\epsilon^{-2}(L+1)^2 c_2 + \sum_{l=0}^L h_l^{-1} \right)$$

und damit durch Verwenden der Formeln (2.8) und (2.11)

$$\begin{aligned} C &\leq c_3 2\epsilon^{-2} c_5^2 \ln(\epsilon^{-1})^2 c_2 + c_3 \frac{M^2}{M-1} (\sqrt{2}c_1)^{\frac{1}{\alpha}} \epsilon^{-2} \\ &\leq \left( 2c_3 c_5^2 c_2 + c_3 \frac{M^2}{M-1} (\sqrt{2}c_1)^{\frac{1}{\alpha}} \right) \ln(\epsilon)^2 \epsilon^{-2} \end{aligned}$$

Definiert man den vorderen konstanten Term als  $c_4$ , dann folgt die Behauptung.

Fall 2:  $\beta > 1$

Sei  $N_l$  definiert als:

$$N_l := \left\lceil 2\epsilon^{-2} c_2 T^{(\beta-1)/2} (1 - M^{-(\beta-1)/2})^{-1} h_l^{(\beta+1)/2} \right\rceil$$

Für die Varianz ergibt sich dann durch Einsetzen von  $N_l$  und der analogen Abschätzung wie im Fall  $\beta = 1$  aus Bedingung (iii) des Theorems

$$\text{Var}(\tilde{Y}) \leq \sum_{l=0}^L c_2 N_l^{-1} h_l \leq \frac{1}{2} \epsilon^2 T^{-(\beta-1)/2} (1 - M^{-(\beta-1)/2}) \sum_{l=0}^L h_l^{(\beta-1)/2}.$$

Für den Term  $\sum_{l=0}^L h_l^{(\beta-1)/2}$  zeigt man

$$\begin{aligned} \sum_{l=0}^L h_l^{(\beta-1)/2} &= T^{(\beta-1)/2} \sum_{l=0}^L \left( M_l^{-(\beta-1)/2} \right)^l \\ &< T^{(\beta-1)/2} \left( 1 - M_l^{-(\beta-1)/2} \right)^{-1}, \end{aligned} \quad (2.12)$$

wobei die letzte Zeile durch das Ergebnis der geometrischen Reihe gerechtfertigt ist. Eingesetzt in die obige Abschätzung der Varianz folgt sofort

$$\text{Var}(\tilde{Y}) \leq \frac{1}{2} \epsilon^2 T^{-(\beta-1)/2} (1 - M^{-(\beta-1)/2}) \sum_{l=0}^L h_l^{(\beta-1)/2} < \frac{1}{2} \epsilon^2$$

und damit auch für  $\beta > 1$  die im Theorem geforderte obere Schranke des MSE von  $\epsilon^2$ .

Die Simulationsanzahl  $N_l$  ist natürlich nach oben beschränkt durch

$$N_l < 2\epsilon^{-2} c_2 T^{(\beta-1)/2} (1 - M^{-(\beta-1)/2})^{-1} h_l^{(\beta+1)/2} + 1$$

womit nach Bedingung (iv) für den Rechenaufwand gilt:

$$\begin{aligned} C &< c_3 \sum_{l=0}^L \left( 2\epsilon^{-2} c_2 T^{(\beta-1)/2} (1 - M^{-(\beta-1)/2})^{-1} h_l^{(\beta+1)/2} + 1 \right) h_l^{-1} \\ &\leq c_3 \left( 2\epsilon^{-2} c_2 T^{(\beta-1)/2} (1 - M^{-(\beta-1)/2})^{-1} \sum_{l=0}^L h_l^{(\beta-1)/2} + \sum_{l=0}^L h_l^{-1} \right) \end{aligned}$$

Die beiden Summen am Ende lassen sich jeweils durch die Ungleichungen (2.12) und (2.8) nach oben abschätzen und somit zusammenfassen zu

$$\begin{aligned} C &< c_3 \left( 2\epsilon^{-2} c_2 T^{\beta-1} (1 - M_l^{-(\beta-1)/2})^{-2} + \frac{M^2}{M-1} (\sqrt{2}c_1)^{\frac{1}{\alpha}} \epsilon^{-2} \right) \\ &= \left( 2c_2 c_3 T^{\beta-1} (1 - M_l^{-(\beta-1)/2})^{-2} + c_3 \frac{M^2}{M-1} (\sqrt{2}c_1)^{\frac{1}{\alpha}} \right) \epsilon^{-2}. \end{aligned}$$

Der vordere konstante Term innerhalb der Klammer entspricht dann gerade  $c_4$  aus dem Theorem und die Aussage ist somit auch für  $\beta > 1$  bewiesen.

Fall 3:  $\beta < 1$ 

Analog zu den vorherigen beiden Fällen definiert man zuerst

$$N_l := \left[ 2\epsilon^{-2} c_2 h_L^{-(1-\beta)/2} (1 - M^{-(1-\beta)/2})^{-1} h_l^{(\beta+1)/2} \right],$$

womit für die Varianz dann gilt

$$\text{Var}(\tilde{Y}) \leq \sum_{l=0}^L c_2 N_l^{-1} h_l \leq \frac{1}{2} \epsilon^2 h_L^{(1-\beta)/2} (1 - M^{-(1-\beta)/2}) \sum_{l=0}^L h_l^{-(1-\beta)/2}.$$

Ähnlich zur Ungleichung (2.12) zeigt man unter Berücksichtigung des Konvergenzergebnisses der geometrischen Reihe für den hinteren Summenoperator

$$\begin{aligned} \sum_{l=0}^L h_l^{-(1-\beta)/2} &= h_L^{-(1-\beta)/2} \sum_{l=0}^L (M_l^{-(1-\beta)/2})^l \\ &< h_L^{-(1-\beta)/2} (1 - M_l^{-(1-\beta)/2})^{-1}. \end{aligned} \quad (2.13)$$

Einsetzen in die Abschätzung der Varianz liefert dann sofort wieder die gewünschte  $\epsilon^2$ -Schranke des MSE.

Da  $N_l$  nach oben beschränkt ist durch

$$N_l < 2\epsilon^{-2} c_2 h_L^{-(1-\beta)/2} (1 - M^{-(1-\beta)/2})^{-1} h_l^{(\beta+1)/2} + 1,$$

gilt für den Gesamtrechenaufwand

$$\begin{aligned} C &\leq c_3 \sum_{l=0}^L \left( 2\epsilon^{-2} c_2 h_L^{-(1-\beta)/2} (1 - M^{-(1-\beta)/2})^{-1} h_l^{(\beta+1)/2} + 1 \right) h_l^{-1} \\ &\leq c_3 \left( 2\epsilon^{-2} c_2 h_L^{-(1-\beta)/2} (1 - M^{-(1-\beta)/2})^{-1} \sum_{l=0}^L h_l^{-(1-\beta)/2} + \sum_{l=0}^L h_l^{-1} \right) \end{aligned}$$

Durch die Ungleichungen (2.13) und (2.8) lassen sich wieder die beiden Summenoperatoren abschätzen und es ergibt sich damit

$$\begin{aligned} C &\leq c_3 \left( 2\epsilon^{-2} c_2 h_L^{-(1-\beta)} (1 - M^{-(1-\beta)/2})^{-2} + \frac{M^2}{M-1} (\sqrt{2}c_1)^{\frac{1}{\alpha}} \epsilon^{-2} \right) \\ &= \epsilon^{-2} \left( 2c_2 c_3 h_L^{-(1-\beta)} (1 - M^{-(1-\beta)/2})^{-2} + c_3 \frac{M^2}{M-1} (\sqrt{2}c_1)^{\frac{1}{\alpha}} \right). \end{aligned}$$

Im Gegensatz zum Fall  $\beta > 1$  kann der geklammerte Ausdruck noch nicht zur Konstanten  $c_4$  zusammengefasst werden, da der Faktor  $h_L^{-(1-\beta)}$  nicht konstant, sondern abhängig von  $L$  ist. Unter Verwendung der am Anfang des Beweises gezeigten Ungleichung (2.7) gilt jedoch

$$h_L^{-(1-\beta)} < (\sqrt{2}c_1)^{(1-\beta)/\alpha} M^{1-\beta} \epsilon^{-(1-\beta)/2}$$

und damit zusammen mit der Tatsache, dass  $\epsilon^{-2} < \epsilon^{-2-(1-\alpha)/2}$  gilt für die im Theorem geforderten Größen der Parameter  $\alpha \geq 1/2$ ,  $\epsilon < e^{-1}$  und  $0 < \beta < 1$  ergibt sich die Behauptung

$$C \leq c_4 \epsilon^{-2-(1-\alpha)/2}$$

mit

$$c_4 := 2c_2c_3(\sqrt{2}c_1)^{\frac{(1-\beta)}{\alpha}} M^{1-\beta} (1 - M^{-(1-\beta)/2})^{-2} + c_3 \frac{M^2}{M-1} (\sqrt{2}c_1)^{\frac{1}{\alpha}}.$$

□

Sowohl aus dem Theorem selbst, als auch aus dem Beweis wird klar, dass der Parameter  $\beta$  eine besondere Rolle spielt. Tatsächlich gibt der Parameter  $\beta$  nicht nur die Ordnung des Rechenaufwands an, sondern gibt auch Auskunft über die Verteilung der Rechenkosten auf die verschiedenen Levels des multi-level Monte-Carlo Verfahrens. Am Anfang des Kapitels wurde gezeigt, dass die Anzahl der MC Simulationen  $N_l$  auf jedem Level proportional zu  $\sqrt{V_l h_l}$  zu wählen sind und da  $V_l$  die Ordnung  $h_l^\beta$  besitzt, gilt nach 2.2 Bedingung (iii) folglich  $\sqrt{V_l h_l} = \mathcal{O}(h_l^{(\beta+1)/2})$ . Für den Rechenaufwand des  $l$ -ten Levels ergibt sich damit  $C_l \approx N_l h_l^{-1} = \mathcal{O}(h_l^{(\beta-1)/2})$ .

Da  $h_l \leq 1$  ist und mit steigendem  $l$  exponentiell gegen 0 konvergiert, folgert man:

- Für  $\beta > 1$  liegt der meiste Rechenaufwand in den „groben“ Levels
- Für  $\beta = 1$  ist der Rechenaufwand etwa gleichmäßig verteilt auf alle Levels
- Für  $\beta < 1$  liegt der meiste Rechenaufwand in den „feinen“ Levels

Tatsächlich ist es vorteilhaft, wenn der Hauptteil der Rechenoperationen auf den größeren Levels benötigt wird, da man so ohne viel Mehraufwand eine größere Anzahl an Levels einsetzen kann um mehr Genauigkeit zu bekommen. Eine höhere Anzahl an Levels ist auch in der Implementierung von Vorteil um den verbleibenden Fehler besser abzuschätzen zu können, wie in Kürze noch gezeigt wird.

Die korrekte Ermittlung des Parameters  $\beta$  kann in der Anwendung allerdings problematisch werden. Während für  $\alpha$  aus Bedingung (i) – also der schwachen Approximationsordnung des numerischen Verfahrens – meist bekannt ist, welche Voraussetzungen an die SDGL gestellt werden

müssen um eine bestimmte schwache Approximationsordnung zu erzielen, ist dies bei der starken Approximationsordnung (und damit implizit  $\beta$ ) oft nicht so klar. Bei der Herleitung unter Verwendung des stochastischen Euler-Verfahrens am Anfang des Kapitels, konnte die Eigenschaft etwa nur mit Hilfe der Ungleichung (1.16) gezeigt werden, die aus dem Beweis über die starke Approximationsordnung stammt. Zudem war die Abschätzung nur möglich, weil wir angenommen hatten, dass die Auszahlungsfunktion  $f$ , bzw. der Payoff  $P$ , Lipschitz-stetig ist. Für Derivate deren Auszahlungsstrukturen Unstetigkeitsstellen aufweisen, wie zum Beispiel der digitalen Option, verringert sich  $\beta$  und erhöht damit den Rechenaufwand. Dies liegt daran, dass für ein gegebenes  $h_l$  ein Teil der Pfade auf dem  $l$ -ten Level so nahe an der Sprungstelle sind, dass der korrespondierende Pfad auf dem  $l - 1$ -ten Level sich auf der anderen Seite der Unstetigkeitsstelle befindet. Der Term  $\tilde{P}_l - \tilde{P}_{l-1}$  ist dadurch größeren Schwankungen unterworfen und im Falle der stochastischen Euler-Approximation ergibt sich für  $V_l$  die Ordnung  $\mathcal{O}(h_l^{1/2})$ , also  $\beta = 1/2$ . Da die schwache Konvergenzordnung und damit  $\alpha$  gleich bleiben, ergibt sich nach Theorem 2.2 ein Rechenaufwand von  $\mathcal{O}(\epsilon^{-2.5})$ , was immer noch eine merkliche Verbesserung gegenüber der einfachen Monte-Carlo Methode darstellt. Das Ergebnis ist allerdings auch schlechter, als die vorher erzielte Ordnung von  $\epsilon^{-2} \ln(\epsilon)^2$ .

## 2.3 Analyse und Erweiterungen

In der bisherigen Herleitung der MLMC Methode wurden einige Aspekte noch nicht betrachtet, wie etwa die Wahl von  $M$ , welche die Konvergenzgeschwindigkeit der Schrittweite  $h_l$  bestimmt. Auch fehlt noch ein geeignetes Abbruchkriterium für den Algorithmus in Abhängigkeit des verbleibenden Fehlers. Das und die Bestimmung der Simulationsanzahl  $N_l$  während der Laufzeit soll im Folgenden genauer diskutiert werden.

### 2.3.1 Optimales $M$

In der eingangs erwähnten Mehrgittermethode ist es üblich das nachfolgende Gitter jeweils zweimal so fein zu wählen wie das Vorherige. Tatsächlich wäre die Wahl von  $M = 2$  in einigen Fällen in der MLMC Methode eher nachteilhaft, wie am Beispiel des stochastischen Euler-Verfahrens und einer Lipschitz-stetigen Auszahlungsfunktion gezeigt werden soll.

Dazu betrachten wir noch einmal die Herleitung der Ordnung von  $V_l$  am Anfang des Kapitels:

$$\begin{aligned} V_l &= \text{Var}(\tilde{P}_l - \tilde{P}_{l-1}) \\ &= \text{Var}(\tilde{P}_l - P + P - \tilde{P}_{l-1}) \\ &= \text{Var}(\tilde{P}_l - P) + \text{Var}(P - \tilde{P}_{l-1}) + 2\text{Cov}(\tilde{P}_l - P, P - \tilde{P}_{l-1}) \end{aligned}$$

Anstatt nun den Betrag der Kovarianz abzuschätzen, verwendet man die Korrelation, die für zwei Zufallsvariablen definiert ist als:

$$\rho(X, Y) := \frac{\text{Cov}(X, Y)}{\sqrt{\text{Var}(X)\text{Var}(Y)}}$$

Der Korrelationskoeffizient  $\rho$  nimmt dabei Werte zwischen -1 und 1 an. Eingesetzt für die Kovarianz in der Gleichung von  $V_l$  ergibt sich

$$\begin{aligned} V_l &= \text{Var}(\tilde{P}_l - P) + \text{Var}(P - \tilde{P}_{l-1}) + 2\rho \text{Var}(\tilde{P}_l - P)^{1/2} \text{Var}(P - \tilde{P}_{l-1})^{1/2} \\ &\leq c_0 h_l + c_0 h_{l-1} + 2\rho (c_0 h_l)^{1/2} (c_0 h_{l-1})^{1/2} \\ &= c_0 h_l + c_0 h_l M + 2\rho (c_0 h_l)^{1/2} (c_0 h_l M)^{1/2} \\ &= c_0 h_l (1 + M + 2\rho \sqrt{M}) \end{aligned}$$

Für  $\rho = \pm 1$ , das heißt perfekt korreliert, bzw. antikorreliert, erhält man sowohl eine obere, als auch eine untere Grenze von  $\text{Var}(\tilde{P}_l - \tilde{P}_{l-1})$ . Es gilt

$$(\sqrt{M} - 1)^2 c_0 h_l \leq \text{Var}(\tilde{P}_l - \tilde{P}_{l-1}) \leq (\sqrt{M} + 1)^2 c_0 h_l.$$

Nimmt man nun an, dass  $V_l$  in etwa dem geometrischem Mittel der beiden Schranken entspricht, dann folgt

$$\text{Var}(\tilde{P}_l - \tilde{P}_{l-1}) \approx (M - 1)c_0 h_l.$$

Der Term  $(M - 1)c_0$  entspricht hier gerade dem  $c_2$  aus Bedingung (iii) des Komplexitätstheorems und da im Fall des stochastischen Euler-Verfahrens bekanntlich  $\beta = 1$  gilt, ergibt sich für das entsprechende  $N_l$  aus dem Beweis

$$N_l = \lceil 2\epsilon^{-2}(L + 1)c_2 h_l \rceil = \lceil 2\epsilon^{-2}(L + 1)(M - 1)c_0 h_l \rceil.$$

Aus Gleichung (2.3) wissen wir, dass der Rechenaufwand  $C_l$  des  $l$ -ten Levels proportional zu  $N_l(h_l^{-1} - h_{l-1}^{-1})$  ist und damit

$$\begin{aligned} N_l(h_l^{-1} - h_{l-1}^{-1}) &= N_l h_l^{-1} (1 + M^{-1}) \\ &= \lceil 2\epsilon^{-2}(L + 1)(M - 1)c_0 h_l \rceil h_l^{-1} (1 + M^{-1}) \\ &\approx 2\epsilon^{-2}(L + 1)(M - M^{-1})c_0. \end{aligned}$$

Mit der Beobachtung, dass  $L = \mathcal{O}(\ln(\epsilon^{-1})/\ln(M))$  gilt, ergibt sich, dass der über alle Levels aufsummierte Rechenaufwand asymptotisch für  $\epsilon \rightarrow 0$  in etwa proportional ist zu

$$C = 2\epsilon^{-2} \ln(\epsilon)^2 \frac{M - M^{-1}}{\ln(M)^2}.$$

Für den von  $M$  abhängigen, hinteren Teil definieren wir

$$f(M) := \frac{M - M^{-1}}{\ln(M)^2}.$$

Ordnungstechnisch ändert sich der Rechenaufwand durch die Wahl von  $M$  natürlich nicht, aber wie in Abbildung 2.2 zu erkennen ist, ist er im Minimum bei  $M = 7$  nur etwa halb so groß wie bei  $M = 2$ . In den Beispielen am Ende des Kapitels wird jeweils  $M = 4$  verwendet, da dies sowohl eine relativ recheneffiziente Wahl darstellt, als auch dazu führt, dass der Algorithmus nicht mit zu wenigen Levels abbricht. Diese können nämlich verwendet werden, um den restlichen Bias zu schätzen wie im Folgenden gezeigt wird.

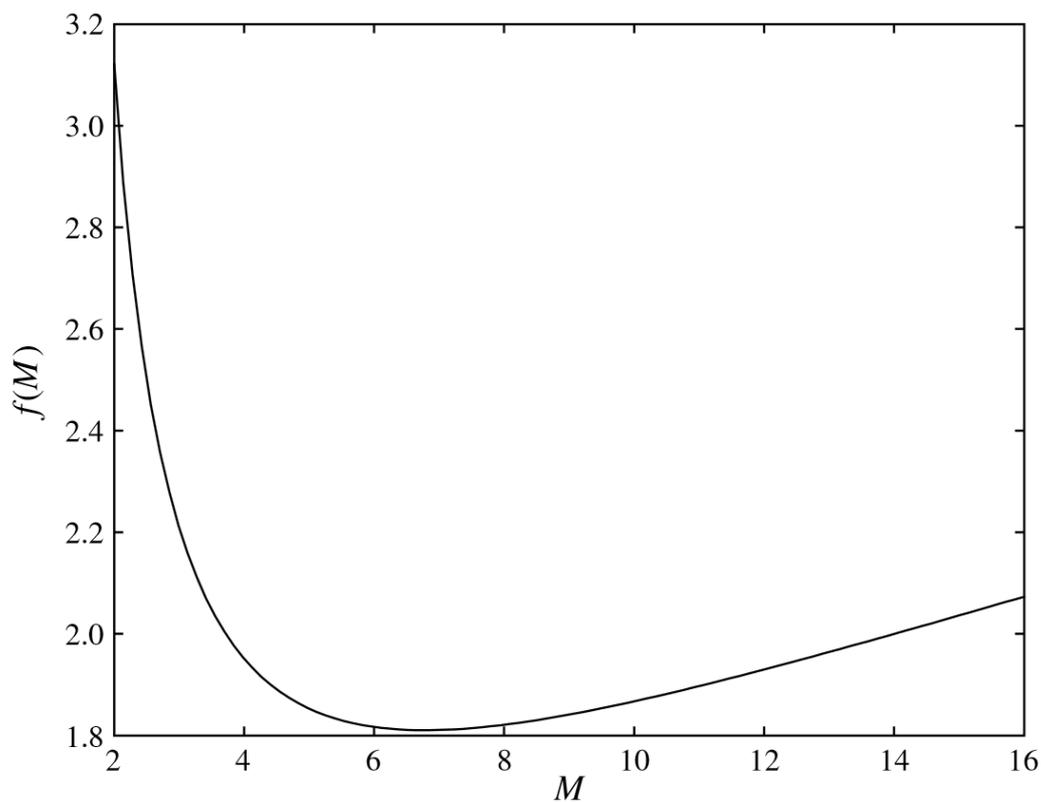


Abbildung 2.2: Graph der Funktion  $f(M)$  für die Bestimmung des optimalen  $M$  ([9])

### 2.3.2 Bias-Schätzung und die Richardson Extrapolation

Um den Algorithmus formulieren zu können, wird noch ein geeignetes Abbruchkriterium benötigt bei dem das Verfahren erfolgreich terminiert. Um Informationen über den verbleibenden Fehler zu bekommen, kann der jeweils zu schätzende Term  $E(\tilde{P}_l - \tilde{P}_{l-1})$  verwendet werden. Am Anfang des Kapitels wurde gezeigt, dass für das stochastische Euler-Verfahren asymptotisch für  $l \rightarrow \infty$  gilt

$$E(P - \tilde{P}_l) \approx c_1 h_l.$$

Dies stimmt natürlich auch für andere numerische Verfahren, deren schwache Konvergenzordnung, und damit  $\alpha$  aus Bedingung (i) des Komplexitätstheorems, gleich eins ist. Für den Erwartungswert der Differenz zweier Levels gilt dann in etwa:

$$\begin{aligned} E[\tilde{P}_l - \tilde{P}_{l-1}] &= E[\tilde{P}_l - P + P - \tilde{P}_{l-1}] \\ &\approx -c_1 h_l + c_1 h_{l-1} \\ &= (M-1)c_1 h_l \\ &\approx (M-1)E[P - \tilde{P}_l] \end{aligned} \quad (2.14)$$

Das heißt der Term, der auf jedem Level mittels der MC Methode geschätzt wird, entspricht ungefähr dem einfachen Bias, also dem verbleibenden Fehler, multipliziert mit  $M-1$ . Nach den vorherigen Überlegungen sollte der einfache Bias echt kleiner sein als  $\epsilon/\sqrt{2}$  um einen MSE von  $\epsilon^2$  zu erhalten, also

$$E[P - \tilde{P}_l] < \frac{1}{\sqrt{2}}\epsilon$$

und damit nach obigen

$$|\tilde{Y}_L| = |E[\tilde{P}_L - \tilde{P}_{L-1}]| \approx |(M-1)E[P - \tilde{P}_L]| < \frac{1}{\sqrt{2}}(M-1)\epsilon. \quad (2.15)$$

Der Algorithmus sollte also so lange feinere Levels erzeugen, bis die Gleichung (2.15) erfüllt ist. In der eigentlichen Implementierung werden wir noch etwas strenger sein und fordern, dass die letzten beiden Levels die Forderung an den einfachen Bias erfüllen. Analog zu (2.14) leitet man dazu die Beziehung

$$E[\tilde{P}_{l-1} - \tilde{P}_{l-2}] \approx M(M-1)E[P - \tilde{P}_l]$$

her, womit wie in (2.15) die Bedingung

$$|\tilde{Y}_{L-1}| = |E[\tilde{P}_{L-1} - \tilde{P}_{L-2}]| \approx |M(M-1)E[P - \tilde{P}_L]| < \frac{1}{\sqrt{2}}M(M-1)\epsilon. \quad (2.16)$$

folgt. Kombiniert man nun (2.15) und (2.16) und normiert letztere Ungleichung durch  $M^{-1}$  erhält man als Abbruchkriterium des Algorithmus

$$\max\{M^{-1}|\tilde{Y}_{L-1}|, |\tilde{Y}_L|\} < \frac{1}{\sqrt{2}}(M-1)\epsilon. \quad (2.17)$$

Ein alternatives Kriterium dazu lässt sich über die *Richardson Extrapolation* herleiten, bei der eine genauere Schätzung von  $\tilde{Y}_l$  mit Hilfe von zwei Approximationen unterschiedlicher Schrittweite berechnet wird.

Im Allgemeinen lässt sich die Richardson Extrapolation wie folgt formulieren. Sei  $\tilde{A}(h)$  eine Approximation des exakten Ergebnisses  $A$  für welches gilt:

$$A = \tilde{A}(h) + a_0 h^{k_0} + a_1 h^{k_1} + \dots$$

Verwendet man nun zwei verschiedene Schrittweiten  $h$  und  $\frac{h}{t}$  für ein  $t > 0$  erhält man die beiden Gleichungen

$$A = \tilde{A}(h) + a_0 h^{k_0} + \mathcal{O}(h^{k_1}) \quad (2.18)$$

und

$$A = \tilde{A}\left(\frac{h}{t}\right) + a_0 \left(\frac{h}{t}\right)^{k_0} + \mathcal{O}(h^{k_1}). \quad (2.19)$$

Multiplizieren von (2.19) mit  $t^{k_0}$  und anschließendes Subtrahieren von (2.18) liefert dann

$$t^{k_0} A - A = t^{k_0} \tilde{A}\left(\frac{h}{t}\right) - \tilde{A}(h) + \mathcal{O}(h^{k_1}),$$

Auflösen nach  $A$  ergibt dann

$$A = \frac{t^{k_0} \tilde{A}\left(\frac{h}{t}\right) - \tilde{A}(h)}{t^{k_0} - 1} + \mathcal{O}(h^{k_1}). \quad (2.20)$$

Durch dieses Vorgehen konnte der führende Fehlerterm der Ordnung  $\mathcal{O}(h^{k_0})$  eliminiert werden und man gewinnt eine Ordnung an Genauigkeit. Der generelle Nachteil der Methode, dass zwei Approximationen zu unterschiedlichen Schrittweiten durchgeführt werden müssen, relativiert sich natürlich in der gemeinsamen Betrachtung mit der multi-level Monte-Carlo Methode.

Angewendet auf die MLMC Methode setzt man nun  $t = M$  und  $\tilde{A} = \tilde{Y}_l$ . Da  $\tilde{Y}_l$  die Fehlerordnung  $\mathcal{O}(h_l^1)$  besitzt, ist  $k_0$  aus Gleichung (2.20) gleich eins, womit sich der neue Schätzer des Erwartungswertes  $E[\tilde{P}_L - \tilde{P}_{L-1}]$

$$\tilde{Y}_{l,RE} = \frac{M\tilde{Y}_l - \tilde{Y}_{l-1}}{M - 1} \quad (2.21)$$

ergibt. Der neue Gesamtschätzer lautet

$$\tilde{Y}_{RE} = \sum_{l=0}^L \tilde{Y}_{l,RE},$$

wobei für das dabei entstehende  $\tilde{Y}_{-1}$  der Wert null gilt. Um wie vorher ein Kriterium für die Terminierung des Algorithmus zu bekommen, bedarf es noch einiger Umformungen.

$$\begin{aligned}
\sum_{l=0}^L \tilde{Y}_{l,RE} &= \sum_{l=0}^L \frac{M\tilde{Y}_l - \tilde{Y}_{l-1}}{M-1} \\
&= (M-1)^{-1} \left( \sum_{l=0}^L M\tilde{Y}_l - \sum_{l=1}^L \tilde{Y}_{l-1} \right) \\
&= (M-1)^{-1} (M\tilde{Y}_0 - \tilde{Y}_0 + M\tilde{Y}_1 - \tilde{Y}_1 + \dots + M\tilde{Y}_L) \\
&= (M-1)^{-1} \left( (M-1)\tilde{Y}_0 + (M-1)\tilde{Y}_1 + \dots + (M-1)\tilde{Y}_L + \tilde{Y}_L \right) \\
&= \sum_{l=0}^L \tilde{Y}_l + (M-1)^{-1}\tilde{Y}_L
\end{aligned}$$

Dies entspricht fast der gewollten Form um den Bias abzuschätzen. Ausklammern des Faktors  $\frac{M}{M-1}$  und Umformen der Summe führt zu:

$$\begin{aligned}
\sum_{l=0}^L \tilde{Y}_l + (M-1)^{-1}\tilde{Y}_L &= \frac{M}{M-1} \left( \left( \frac{M-1}{M} \right) \sum_{l=0}^L \tilde{Y}_l + M^{-1}\tilde{Y}_L \right) \\
&= \frac{M}{M-1} \left( \sum_{l=0}^L \tilde{Y}_l - \frac{1}{M} \sum_{l=0}^L \tilde{Y}_l + \frac{1}{M} \tilde{Y}_L \right) \\
&= \frac{M}{M-1} \left( \tilde{Y}_0 + \sum_{l=1}^L \tilde{Y}_l - \frac{1}{M} \left( \sum_{l=1}^L \tilde{Y}_{l-1} + \tilde{Y}_L - \tilde{Y}_L \right) \right) \\
&= \frac{M}{M-1} \left( \tilde{Y}_0 + \sum_{l=1}^L \left( \tilde{Y}_l - \frac{1}{M} \tilde{Y}_{l-1} \right) \right)
\end{aligned}$$

Diese Form des neuen Gesamtschätzers  $\tilde{Y}_{RE}$  hat den Vorteil, dass nun wie vorher schon während der Laufzeit des Algorithmus auf jedem Level der Term  $\tilde{Y}_l - \frac{1}{M} \tilde{Y}_{l-1}$  zur Abschätzung des verbleibenden Fehlers verwendet werden kann. Da in diesem Fall der Schätzer (2.21) eine Ordnung von  $\mathcal{O}(h_l^2)$  besitzt, rechnet man analog zu (2.14) nun

$$\begin{aligned}
E[\tilde{P}_l - \tilde{P}_{l-1}] &= E[\tilde{P}_l - P + P - \tilde{P}_{l-1}] \\
&\approx -c_1 h_l^2 + c_1 h_{l-1}^2 \\
&= (M^2 - 1)c_1 h_l^2 \\
&\approx (M^2 - 1)E[P - \tilde{P}_l].
\end{aligned} \tag{2.22}$$

Kombiniert man all diese Ergebnisse wieder mit dem Kriterium für die Terminierung des Algorithmus – der einfache Bias soll echt kleiner als  $\epsilon/\sqrt{2}$  sein – erhält man wie in (2.15):

$$|\tilde{Y}_{L,RE}| = \left| \frac{M}{M-1} \left( \tilde{Y}_L - \frac{1}{M} \tilde{Y}_{L-1} \right) \right| = |E[\tilde{P}_L - \tilde{P}_{L-1}]| \approx |(M^2 - 1)E[P - \tilde{P}_L]| < \frac{1}{\sqrt{2}} (M^2 - 1)\epsilon$$

Und damit

$$\left| \frac{M}{M-1} \left( \tilde{Y}_L - \frac{1}{M} \tilde{Y}_{L-1} \right) \right| < \frac{1}{\sqrt{2}} (M^2 - 1) \epsilon,$$

wobei man in der eigentlichen Implementierung den Faktor  $\frac{M}{M-1}$  weglässt und nur das Abbruchkriterium

$$|\tilde{Y}_L - M^{-1} \tilde{Y}_{L-1}| < \frac{1}{\sqrt{2}} (M^2 - 1) \epsilon \quad (2.23)$$

verwendet. Tatsächlich lässt sich mit der Verwendung von (2.21) und der damit verbundenen verbesserten Bias-Schätzung Rechenzeit gegenüber (2.17) sparen. Ein Vergleich der multi-level Monte-Carlo Methode mit und ohne Richardson Extrapolation befindet sich am Ende des Kapitels.

### 2.3.3 Anzahl der Simulationen

Ziel der Bias-Schätzungen (2.14) und (2.23) ist den verbleibenden Fehler während der Laufzeit abzuschätzen und den Algorithmus erst zu beenden, wenn die Abweichung unter der gewünschten Schranke liegt. Der andere Fehlerteil des MSE, die Varianz des Gesamtschätzers, kann durch eine Erhöhung der Simulationsanzahl verringert werden. Im Beweis des Komplexitätstheorems wurde in Abhängigkeit von  $\beta$  eine bestimmte Wahl für  $N_l$  verwendet, die die Einhaltung der  $\epsilon^2/2$ -Schranke garantierte. In der eigentlichen Implementierung wird die Simulationsanzahl

$$N_l = \left\lceil 2\epsilon^{-2} \sqrt{V_l h_l} \left( \sum_{i=0}^L \sqrt{V_i h_i^{-1}} \right) \right\rceil \quad (2.24)$$

gesetzt. Für diese Wahl ist die Varianz des Schätzers unabhängig des  $\beta$ -Wertes kleiner als die gewünschte  $\epsilon^2/2$ -Obergrenze, da gilt:

$$\begin{aligned} \text{Var}(\tilde{Y}) &= \sum_{l=0}^L \frac{V_l}{N_l} \\ &\leq \sum_{l=0}^L \frac{V_l}{2\epsilon^{-2} \sqrt{V_l h_l} \left( \sum_{i=0}^L \sqrt{V_i h_i^{-1}} \right)} \\ &= \frac{1}{2} \epsilon^2 \left( \sum_{l=0}^L \sqrt{V_l h_l^{-1}} \right) \left( \sum_{i=0}^L \sqrt{V_i h_i^{-1}} \right)^{-1} \\ &= \frac{1}{2} \epsilon^2 \end{aligned}$$

## 2.4 Implementierung und Anwendung

Die bisher gezeigten Resultate erlauben es uns nun den Grundalgorithmus der multi-level Monte-Carlo Methode zu formulieren und diesen kritisch zu analysieren. Der Algorithmus wurde zudem im Rahmen der vorliegenden Masterarbeit in MATLAB implementiert und wird am Ende des Kapitels eingesetzt, um einerseits den fairen Preis verschiedener Finanzderivate zu ermitteln, und andererseits um Anhand von Größen wie der Varianz einen Vergleich mit der einfachen Monte-Carlo Methode zu ermöglichen.

### 2.4.1 Numerischer Algorithmus

Sei wie bisher  $L$  die Anzahl der Levels,  $V_L$  die Varianz des Schätzers und  $N_L$  die Anzahl der Simulationen, dann formuliert man für die MLMC Methode den Algorithmus:

#### Algorithmus 2.3

*Schritt 1:* Starte mit  $L = 0$

*Schritt 2:* Schätze  $\tilde{Y}_L$  und  $V_L$  mittels  $N_L = 10^4$  Monte-Carlo Simulationen

*Schritt 3:* Berechne neue  $N_l$  für  $l = 0, 1, \dots, L$  mittels der Gleichung (2.24):

$$N_l = \left\lceil 2\epsilon^{-2} \sqrt{V_l h_l} \left( \sum_{l=0}^L \sqrt{V_l h_l^{-1}} \right) \right\rceil$$

*Schritt 4:* Berechne für alle Levels die zusätzlich benötigten Simulationen entsprechend der  $N_l$  und aktualisiere alle  $\tilde{Y}_l$ s und  $V_l$ s

*Schritt 5:* Falls  $L \geq 2$  teste auf Konvergenz mittels der Gleichung (2.17)

$$\max\{M^{-1}|\tilde{Y}_{L-1}|, |\tilde{Y}_L|\} < \frac{1}{\sqrt{2}}(M-1)\epsilon,$$

oder der Richardson Extrapolation (2.23):

$$|\tilde{Y}_L - M^{-1}\tilde{Y}_{L-1}| < \frac{1}{\sqrt{2}}(M^2 - 1)\epsilon.$$

*Schritt 6:* Falls  $L < 2$  oder keine Konvergenz vorliegt setze  $L = L + 1$  und gehe zu *Schritt 2*

□

Ausgehend vom Level 0, das heißt die gesamte Laufzeit der zugrundeliegenden SDGL wird in nur einem einzigen Schritt approximiert, wird das Ergebnis und die Varianz über die Monte-Carlo Methode ermittelt. Iterativ wird dann die Levelanzahl erhöht bis die vorher festgesetzte minimale

Anzahl an Levels erreicht ist und zudem der Konvergenztest vermuten lässt, dass der verbleibende Fehler kleiner als  $\epsilon^2$  ist. Hier liegt auch die Schwäche des Algorithmus, da sowohl die Einzelvarianzen  $V_l$ , als auch der Bias geschätzt werden, ist der Algorithmus heuristischer Natur und kann einen MSE kleiner als  $\epsilon^2$  nicht garantieren. Im Komplexitätstheorem wurde die Schranke zwar bewiesen, es wurde dabei aber implizit angenommen, dass die Größe der Konstanten  $c_1$  und  $c_2$ , die die Konvergenzgeschwindigkeit der schwachen Approximationsordnung und der Varianz bestimmen, bekannt sind. Im Allgemeinen ist dies natürlich nicht der Fall, weshalb sie im Algorithmus geschätzt werden, wie etwa in der Bias-Schätzung und der zur Herleitung verwendeten Gleichung (2.14).

Tatsächlich ist die Bias-Schätzung der Schwachpunkt des Algorithmus, da für den Varianzteil des MSE eine entsprechende  $\epsilon^2/2$  Schranke gezeigt werden kann. Dazu muss die initiale Anzahl an Pfaden  $N_L$  (im obigen Algorithmus auf  $10^4$  gesetzt) proportional zu  $\epsilon^{-p}$  mit  $0 < p < 2 - 1/\alpha$  gesetzt werden. Für  $\epsilon \rightarrow 0$  lässt sich dann zeigen, dass die Wahrscheinlichkeit einer korrekten Approximation gegen eins geht, ohne die Ordnung des Rechenaufwands zu erhöhen. Für die Bias-Schätzung gibt es leider kein solches Vorgehen. Das resultierende Problem eines verfrühten Abbruchs des Algorithmus lässt sich wie folgt darstellen.

Angenommen  $p(S)$  sei die wahre Wahrscheinlichkeitsverteilung der SDGL zum Laufzeitende  $T$  und  $p_l(S)$  mit  $l = 0, 1, \dots, L$  die geschätzten Verteilungen aus den Approximationen der  $L + 1$  Levels. Im Allgemeinen werden die Verteilungen  $p(S)$  und  $p_l(S)$  natürlich linear unabhängig sein und es gilt

$$p(S) = g(S) + \sum_{l=0}^L p_l(S),$$

wobei  $g(S)$  ein zu den  $p_l(S)$  orthogonaler Rest ist. Angenommen die Auszahlungsfunktion  $f$  besitzt nun eine Komponente, die lediglich von dem durch  $g(S)$  beschriebenen Teil abhängt, dann ist klar, dass der berechnete Erwartungswert einen womöglich großen Fehler beinhaltet. Denn egal wie der von  $g(S)$  abhängige Teil der Auszahlungsfunktion erhöht oder verringert wird, der Algorithmus terminiert nach  $L$  Levels mit dem gleichen Erwartungswert.

Dieses Problem betrifft nicht nur die multi-level, sondern auch die einfache Monte-Carlo Methode. Eine „Lösung“ für dieses Problem ist die Erhöhung der Minimalanzahl der Levels, womit der unbekannte Rest  $g(S)$  kleiner wird und demzufolge auch der Fehler. Im Hinblick auf den Rechenaufwand sind die Kosten einer solchen Erhöhung im Falle der Euler-Approximation und einer Lipschitz-stetigen Payoff-Funktion proportional zu  $L^2$ . Dies ist eine wesentliche Verbesserung gegenüber der einfachen Monte-Carlo Methode, bei der die Kosten proportional zu  $M^L$  sind. Am Ende des Beweises des Komplexitätstheorems wurde bemerkt, dass für den Fall  $\beta > 1$  die

Rechenkosten vor allem auf den groben Levels liegen, womit eine Erhöhung von  $L$  besonders unproblematisch ist. Ob es tatsächlich ein  $L$  gibt, für das bewiesen werden kann, dass der Algorithmus mit einem MSE kleiner als  $\epsilon^2$  terminiert, ist Gegenstand der derzeitigen Forschung.

### 2.4.2 Beispiele

Für die folgenden Optionsbewertungen wird wieder das stochastische Euler-Verfahren als numerische Approximationsmethode eingesetzt und im Fall der MLMC Methode wird der Parameter  $M = 4$  verwendet. Sofern nicht anders erwähnt, wird zur Bias-Schätzung die Gleichung (2.17) verwendet. Die zur Kursmodellierung eingesetzte stochastische Differentialgleichung ist wie im vorherigen Beispiel die geometrische Brownsche Bewegung (1.10) mit den Parametern  $\mu = 0.05$ ,  $\sigma = 0.2$  und  $S_0 = 1$ .

Als erstes soll wieder die europäische Call-Option (1.1) mit  $T = K = 1$  betrachtet werden.

<b>Ergebnis</b> <b>Methoden</b>	<b>Lauf 1</b>	<b>Lauf 2</b>	<b>Lauf 3</b>	<b>Lauf 4</b>	<b>#Funktionsauswertungen</b>
<b>MLMC</b> $\epsilon = 0.001$	0.1038	0.1044	0.1047	0.1043	$L = 2$ ~97.000
<b>MLMC</b> $\epsilon = 0.0005$	0.1048	0.1043	0.1041	0.1047	$L = 2$ ~390.000
<b>MLMC</b> $\epsilon = 0.0002$	0.1043	0.1043	0.1047	0.1044	$L = 3$ ~3.600.000
<b>MLMC</b> $\epsilon = 0.0001$	0.1046	0.1044	0.1044	0.1045	$L = 3$ ~14.300.000
<b>MLMC</b> $\epsilon = 0.00005$	0.1044	0.1045	0.1045	0.1044	$L = 3$ ~57.000.000

Tabelle 2.1: Ergebnisse einer europäischen Call Bewertung mittels der MLMC Methode

In der Spalte „#Funktionsauswertungen“ der Tabelle 2.1 ist neben der ungefähren Anzahl an berechneten diskreten Punkten innerhalb der SDGL-Simulationen auch die Anzahl der Monte-Carlo Levels  $L$  abgetragen. Ersteres lässt sich nur ungefähr angeben, da die zur Berechnung der Simulationsanzahl verwendete Formel (2.24) die Varianzen  $V_l$  beinhaltet, welche über das Monte-Carlo Verfahren geschätzt werden. Die über die Formel (2.3) berechnete Anzahl an Simulationen variiert daher geringfügig von Durchlauf zu Durchlauf.

Die Ergebnisse der Tabelle 2.1 bestätigen die theoretische Analyse, dass im Vergleich zur einfachen Monte-Carlo Methode (Tabelle 1.1) eine schnellere Konvergenzrate durch einen multi-level Ansatz erzielt werden kann. Besonders deutlich wird der Vorteil der MLMC Methode beim Vergleich der Varianzen der Schätzer  $Y_l$ .

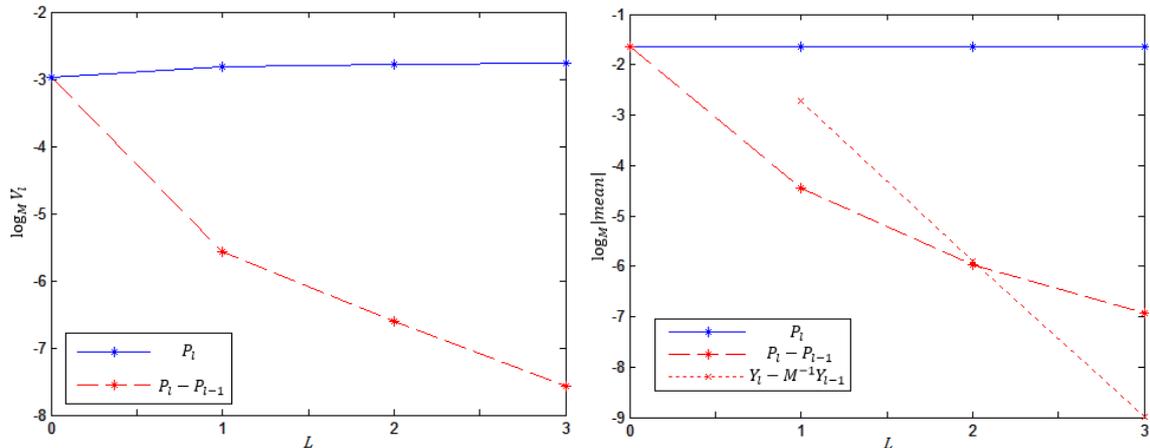


Abbildung 2.3: Vergleich der Entwicklung der MC/MLMC Varianz- und Erwartungswertschätzer bei der europäischen Call-Option

Der linke Graph in Abbildung 2.3 zeigt die Entwicklung der Varianzen der Schätzer sowohl von der einfachen Monte-Carlo Methode (blau), also auch von der multi-level Monte-Carlo Methode (rot). Die Varianzen wurden geplottet bezüglich der  $\log_M$ -Varianz auf der Y-Achse und  $L$  auf der X-Achse. Entsprechend der Wahl der Achsen erwarten wir aus der theoretischen Analyse heraus eine Steigung von  $-1$  für den Graphen der  $V_l$ s, da  $V_l$  die Ordnung  $\mathcal{O}(h_l)$  besitzt und  $h_l$  proportional zu  $M^{-l}$  ist. Tatsächlich lässt sich eine entsprechende Steigung zwischen den Varianzen der multi-level Schätzer  $V_1, V_2$  und  $V_3$  beobachten und die numerische Simulation bestätigt somit die theoretische Analyse. Die Varianzentwicklung des einfachen Monte-Carlo Schätzers ist wie erwartet unabhängig von der Wahl der Schrittweite und bleibt konstant. Bemerkenswert ist der besonders deutliche Unterschied der Varianz des einfachen Schätzers  $P_0$  und dem ersten multi-level Schätzer  $P_1 - P_0$ .

Der rechte Graph der Abbildung 2.3 zeigt die  $\log_M$ -Entwicklung der Schätzer selbst. Der blaue Graph der einfachen MC Methode bleibt natürlich konstant, während sich für die Terme  $E[\tilde{P}_l - \tilde{P}_{l-1}]$  wieder die vermutete Steigung von  $-1$ , also die Ordnung  $\mathcal{O}(h_l)$ , beobachten lässt. Der zusätzlich eingetragene, rot-gepunktete Graph des Schätzers der Richardson Extrapolation zeigt eine Steigung von  $-2$ , was einer  $\mathcal{O}(h_l^2)$  Konvergenz entspricht und sich auch entsprechend mit der Herleitung in Kapitel 2.3.2 deckt, da ja gerade der führende Fehler der Ordnung  $\mathcal{O}(h_l)$  eliminiert wurde.

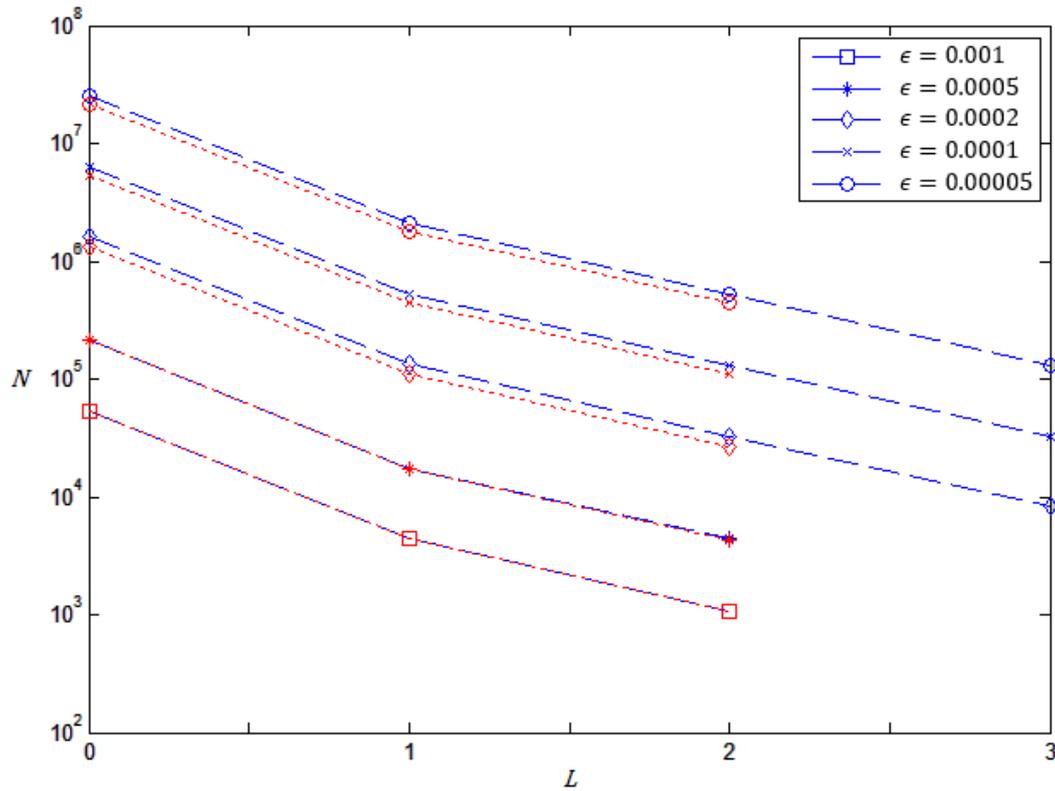


Abbildung 2.4:  $N_l$  für verschiedene  $\epsilon$  beim europ. Call jeweils mit und ohne Rich. Ext.

Die obige Abbildung 2.4 zeigt die Anzahl der berechneten Simulationen pro Level der MLMC Methode für verschiedene  $\epsilon$ , wobei für die blauen Graphen der normale Schätzer bzw. Bias-Schätzer verwendet wurde und für die roten Graphen die Richardson Extrapolation zum Einsatz kam. Wie erwartet nimmt  $N$  auf allen Levels größere Werte an je kleiner  $\epsilon$  gewählt wird. Unabhängig vom Schätzer lässt sich in beiden Fällen eine deutliche Abnahme der Simulationsanzahl auf höheren Levels feststellen. Interessant ist, dass mit der Richardson Extrapolation nicht nur weniger Simulationen benötigt werden um die gewünschte Genauigkeit zu erhalten, sondern für die Fälle  $\epsilon$  gleich 0.0002, 0.0001 und 0.00005 sogar ein Level weniger simuliert wird.

Für die Analyse des Rechenaufwands multiplizieren wir die über Formel (2.3) ermittelten Rechenkosten  $C$  mit dem MSE  $\epsilon^2$ . Für die MLMC Methode erwarten wir dann nur einen geringen Anstieg der Größe  $\epsilon^2 C$  für ein kleiner werdendes  $\epsilon$ , da nach Theorem 2.2 der Rechenaufwand proportional zu  $\epsilon^{-2} \ln(\epsilon)^2$  ist. Um den Rechenaufwand von MC und MLMC miteinander vergleichen zu können, schätzen wir die Rechenkosten der einfachen MC Methode mit einer vergleichbaren Genauigkeit auf

$$C_{MC} = \sum_{l=0}^L N_l^* M^l, \quad (2.24)$$

wobei  $N_l^* = 2\epsilon^{-2} V_l$ .

Diese Wahl der  $N_l^*$  führt wie im MLMC Fall zu einer  $\frac{1}{2}\epsilon^2$  Varianz des Schätzers und die Summation über mehrere Levels erlaubt eine äquivalente Bias-Schätzung.

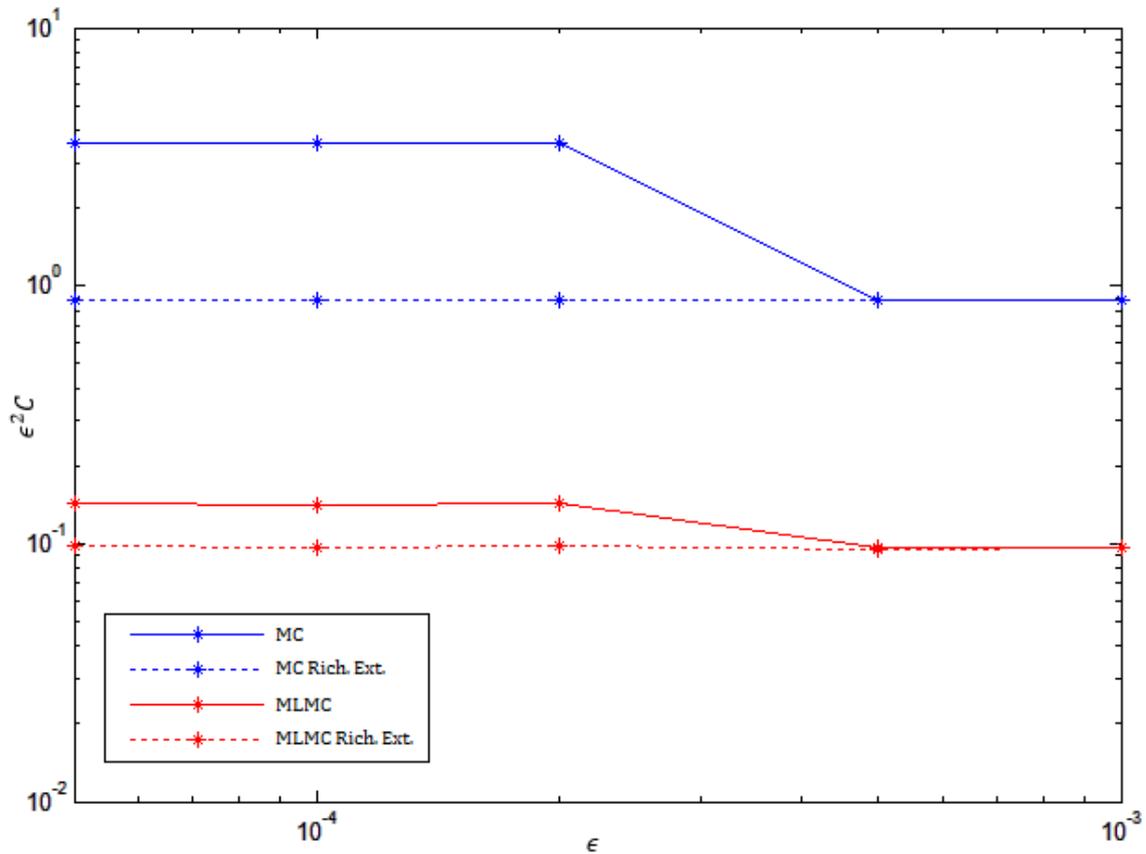


Abbildung 2.5: Entwicklung der Rechenkosten bei MC und MLMC mit und ohne Rich. Ext.

Die obige Abbildung 2.5 zeigt die Entwicklung der Rechenkosten multipliziert mit  $\epsilon^2$  in Abhängigkeit von  $\epsilon$ , jeweils mit und ohne Verwendung der Richardson Extrapolation, wobei zu bemerken gilt, dass beide Achsen logarithmisch skaliert wurden. Ohne Richardson Extrapolation lässt sich für die MLMC Methode für ein kleiner werdendes  $\epsilon$  die vermutete leichte Steigung beobachten, die der asymptotischen  $\ln(\epsilon)^2$  Entwicklung entspricht. Diese Steigung lässt sich aufgrund der für den Rechenaufwand verwendeten Formel (2.3) nur beobachten, wenn sich die Levelanzahl erhöht. Für die einfache Monte-Carlo Methode erwartet man aus der Theorie einen linearen Anstieg für  $\epsilon^2 C$  bzgl.  $\epsilon^{-1}$ , da der Aufwand proportional zu der Anzahl der Berechnungen auf der kleinsten Schrittweite ist. Dieser lineare Anstieg lässt sich an dem plötzlichen Sprung beobachten, der mit der Erhöhung der Levelanzahl  $L$  von zwei auf drei einhergeht. Unter Verwendung der Richardson Extrapolation erwartet man für die MLMC Methode einen asymptotischen Anstieg proportional zu  $\epsilon^{-1/2}$ , da für die im Beispiel gewählten RMSE Werte von 0.001 bis 0.00005 jedoch immer nur zwei Levels benötigt wurden, bleibt der  $\epsilon^2 C$  hier konstant. Analoges gilt für die einfache MC Methode.

Insgesamt lässt sich beobachten, dass für eine Genauigkeit von  $\epsilon$  gleich 0.00005 und unter Verwendung der Richardson Extrapolation der MLMC Algorithmus fast zehn Mal so schnell wie der einfachen MC Methode ist. Ohne Richardson Extrapolation ist die MLMC Methode für die Bewertung von europäischen Optionen sogar um das 25-fache effizienter.

Als nächstes wollen wir das multi-level Monte-Carlo Verfahren an asiatischen Optionen testen. Das in Kapitel 1 definierte Auszahlungsprofil lautete

$$V(T, S(T)) = \max\left(\frac{1}{T} \int_0^T S(\tau) d\tau - K, 0\right),$$

wobei für die Implementierung das Integral in der Auszahlungsfunktion des  $l$ -ten Levels approximiert wird durch die Summe

$$\bar{S}_l = \frac{1}{2} h_l \sum_{n=0}^{n_T} (\tilde{S}_{n+1} + \tilde{S}_n).$$

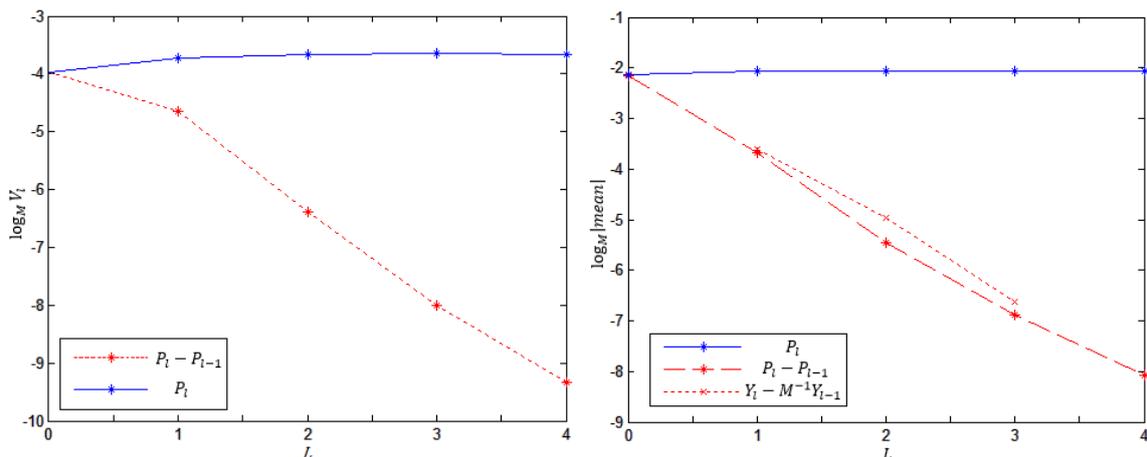


Abbildung 2.6: Vergleich der Entwicklung der MC/MLMC Varianz- und Erwartungswertschätzer bei der asiatischen Call-Option

Die in Abbildung 2.6 präsentierten Ergebnisse für den asiatischen Call sind analog zu denen der europäischen Option zu interpretieren. Interessant ist der etwas steiler als  $-1$  fallende Graph der Varianzen  $V_l$  im linken Plot, was auf eine leicht bessere Konvergenz der  $V_l$  hinweist, als es die aus theoretischen Analyse ermittelte Ordnung  $\mathcal{O}(h_l)$  zuerst vermuten lässt. Bemerkenswert ist außerdem die Tatsache, dass die Richardson Extrapolation keine sichtliche Verbesserung bringt. Da diese durch die Eliminierung des führenden Fehlers der Ordnung  $\mathcal{O}(h_l)$  erzielt wird, lässt sich im Kontext mit den Ergebnissen zur MLMC Methode ohne Extrapolation vermuten, dass der führende Fehler relativ gering sein muss und wahrscheinlich deshalb die Anwendung keine merklichen Vorteile bringt.

Allerdings sollte die korrekte Terminierung und Bias-Schätzung im Fall der Richardson Extrapolation trotzdem kritisch hinterfragt werden. Dies gilt insbesondere dadurch, dass, wie in Abbildung 2.7 gezeigt ist, der Algorithmus mit Richardson Extrapolation für einige  $\epsilon$  mit weniger Levels terminiert als die MLMC Methode ohne Extrapolation.

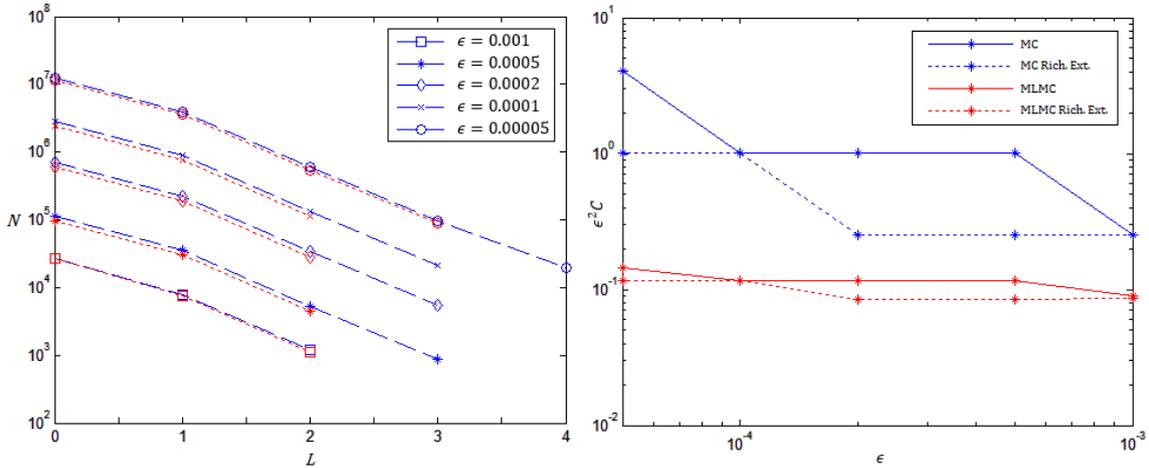


Abbildung 2.7: Entwicklung der Simulationsanzahl und Rechenkosten der MC und MLMC mit und ohne Rich. Ext. für eine asiatische Call-Option

Dass der Algorithmus heuristisch ist und aufgrund der Bias-Schätzung nicht immer sicher mit der gleichen Anzahl an Levels terminiert, zeigte sich beim Test etwa für den Wert  $\epsilon = 0.0001$ , für den mit Richardson Extrapolation entweder zwei oder drei Levels benötigt wurden.

Als letztes wollen wir den Algorithmus für die Bewertung einer digitalen Option einsetzen. Das Besondere an dieser Option ist, dass – im Gegensatz zur europäischen oder asiatischen Option – die Auszahlungsfunktion nicht stetig ist.

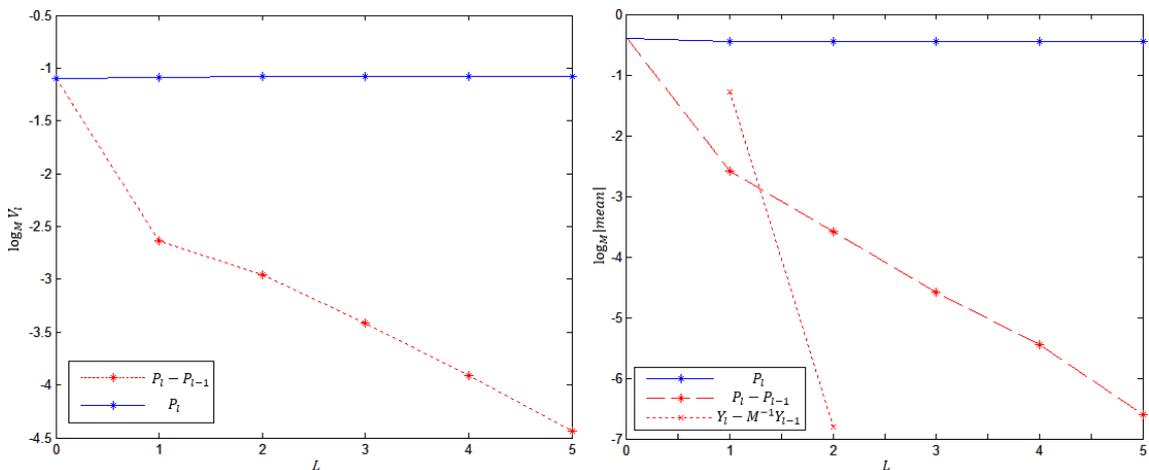


Abbildung 2.8: Vergleich der Entwicklung der MC/MLMC Varianz- und Erwartungswertschätzer bei der digitalen Call-Option

Wie erwartet lässt sich im linken Plot der Abbildung 2.8 ein langsames Sinken der Varianzen beobachten. Die Steigung ist in etwa  $-1/2$ , was einer Konvergenzordnung von  $\mathcal{O}(h_l^{1/2})$  entspricht. Dies hat natürlich eine erhöhte Anzahl an Simulation und Levels zur Folge, um die gewünschte Genauigkeit zu erhalten. Besonders auffällig ist die enorme Einsparung die sich durch die Verwendung der Richardson Extrapolation ergibt. Für die höchste Genauigkeit  $\epsilon = 0.0002$  verringert sich nicht nur die Anzahl der Levels von fünf auf zwei, sondern auch die  $N_l$  sinken deutlich wie in Abbildung 2.9 zu sehen ist.

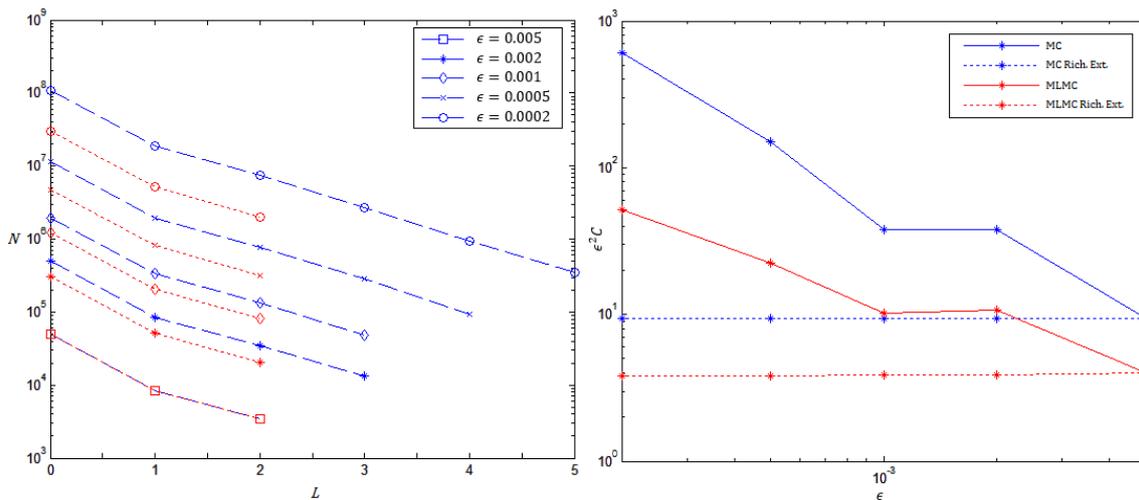


Abbildung 2.9: Entwicklung der Simulationsanzahl und Rechenkosten der MC und MLMC mit und ohne Rich. Ext. bei der digitalen Call-Option

Der rechte Plot der obigen Abbildung zeigt, dass auch hier ein multi-level Ansatz beachtliche Recheneinsparungen ermöglicht. Die Konvergenzordnung im Fall der Richardson Extrapolation lässt sich zwar nicht klar erkennen, aber die MLMC Methode ist auch hier klar besser als das einfache Monte-Carlo Verfahren.

### 3. Die MLMC Methode und das Milstein-Verfahren

Im letzten Kapitel wurde gezeigt, dass der Rechenaufwand durch den Einsatz eines multi-level Ansatzes deutlich reduziert werden konnte. Für die Simulation des Basiswerts wurde dabei stets das stochastische Euler-Verfahren eingesetzt, mit dem sich der Rechenaufwand von  $\mathcal{O}(\epsilon^{-3})$  auf  $\mathcal{O}(\epsilon^{-2} \ln(\epsilon)^2)$  reduzierte. Das Komplexitätstheorem 2.2, das unabhängig von der Wahl des numerischen Approximationsverfahrens formuliert wurde, besagt, dass der Rechenaufwand sogar auf  $\mathcal{O}(\epsilon^{-2})$  verringert werden kann. Dazu muss allerdings die Varianz des Schätzers  $\tilde{Y}_l$  die Ordnung  $\mathcal{O}(h_l^\beta)$  für ein  $\beta > 1$  besitzen.

Basierend auf [11] und [12] soll in diesem Kapitel zunächst für skalare SDGL gezeigt werden, dass durch die Verwendung des Milstein-Verfahrens weitere Konvergenzverbesserungen für europäische, asiatische und digitale Optionen theoretisch möglich sind. Die theoretischen Ergebnisse werden wie vorher anhand numerischer Simulationen überprüft und diskutiert. Im zweiten Teil des Kapitels wird kurz die sich ergebende Problematik bei mehrdimensionalen stochastischen Differentialgleichungen erläutert sowie der in [13] vorgestellte Lösungsansatz skizziert.

#### 3.1 Eindimensionale SDGL

Für skalare SDGL lässt sich das Milstein-Verfahren problemlos in die multi-level Monte-Carlo Methode integrieren, wobei für die theoretische Analyse der Konvergenzgeschwindigkeiten jedoch einige Ergebnisse aus der Numerik und Statistik benötigt werden. Insbesondere für exotische Optionen wie digitale-, lookback- oder Barriere-Optionen sind mehrere Resultate über Extremwerte von Verteilungen sowie die Wahrscheinlichkeit von „extremen“ Pfaden und ihre Auswirkung auf den Erwartungswert von Nöten. Die Notwendigkeit dieser Resultate kommt dabei aus der Konstruktion der Schätzer  $\tilde{Y}_l$ . In Kapitel 2 sahen wir, dass die MLMC Methode auf der Schätzung der rechten Seite von Gleichung (2.1)

$$E[\tilde{P}_L] = E[\tilde{P}_0] + \sum_{l=1}^L E[\tilde{P}_l^f - \tilde{P}_{l-1}^g]$$

beruht. Auf dem  $l$ -ten Level wird also ein genauerer Erwartungswert  $E[\tilde{P}_l^f]$  unter Verwendung der feinen Schrittweite  $h_l = M^{-l}T$  und ein ungenauerer Erwartungswert  $E[\tilde{P}_{l-1}^g]$  mittels der groben Schrittweite  $h_{l-1} = M^{-l-1}T$  bestimmt.

Natürlich gilt Gleichung (2.1) nur, wenn für die Erwartungswerte gleicher Schrittweite von benachbarten Levels gilt:

$$E[\tilde{P}_l^f] = E[\tilde{P}_l^g] \quad (3.1)$$

Zusammen mit dem stochastischen Euler-Verfahren wurde dies bisher ganz einfach dadurch gelöst, dass für alle Levels  $l = 0, 1, \dots, L$  die jeweiligen Schätzer immer gleich als

$$\tilde{Y}_l = \frac{1}{N_l} \sum_{i=1}^{N_l} (\tilde{P}_l^{(i)} - \tilde{P}_{l-1}^{(i)})$$

definiert wurden, wobei  $\tilde{P}_{-1}^{(i)} = 0$  gilt. Um die höhere starke Konvergenzordnung des Milstein-Verfahrens ausnutzen zu können und ein  $\beta > 1$  zu erreichen, reicht es für Lipschitz-stetige Auszahlungsfunktionen aus, den obigen Schätzer weiterzuverwenden. Im Fall von nicht Lipschitz-stetigen Auszahlungsfunktionen müssen allerdings neue Schätzer für den „feinen“ und „groben“ Erwartungswert  $E[\tilde{P}_l^f]$  und  $E[\tilde{P}_{l-1}^g]$  definiert werden. Diese müssen einerseits das Ziel der Varianzreduktion – und somit einen höheren  $\beta$ -Wert – haben und andererseits Gleichung (3.1) genügen. Um eine niedrigere Varianz zu erreichen, verwendet man für die Kalkulation von  $E[\tilde{P}_{l-1}^g]$  Informationen aus der Berechnung von  $E[\tilde{P}_l^f]$ . Allerdings stehen diese Informationen nicht für die Berechnung von  $E[\tilde{P}_{l-1}^f]$  im vorherigen Level zur Verfügung, was die Konstruktion erschwert, da weiterhin Gleichung (3.1) gelten muss. Um dennoch entsprechende Schätzer konstruieren zu können, verwendet man zusätzliche Approximationen der Brownschen Bewegung zwischen den diskreten Zeitpunkten  $t_n, t_{n+1}$ . Besonders relevant sind diese zusätzlichen Approximationen für die Schätzer von Barriere-Optionen, aber auch bei den hier betrachteten europäischen, asiatischen und digitalen Optionen werden einige zusätzliche Ergebnisse benötigt, die im Folgenden wie in [11] und [12] eingeführt werden sollen.

Grundsätzlich nehmen wir an, dass die zur Simulation des Basiswertes verwendete stochastische Differentialgleichung

$$dS(t) = a(t, S(t))dt + b(t, S(t))dW_t$$

skalar ist und  $a \in C^{1,2}(\mathbb{R}^+ \times \mathbb{R})$ , sowie  $b \in C^{1,3}(\mathbb{R}^+ \times \mathbb{R})$  gilt. Wie in Kapitel 1 erwähnt, ergibt das Milstein-Verfahren für obige SDGL mit Startwert  $S(0) = \tilde{S}_0$  dann die Iterationsvorschrift

$$\tilde{S}_{n+1} = \tilde{S}_n + a_n h + b_n \Delta W_n + \frac{1}{2} b_n' b_n (\Delta W_n^2 - h), \quad (3.2)$$

wobei  $b_n'$  die Ableitung  $\partial b / \partial S$  sei. In dem Werk „Numerical Solution of Stochastic Differential Equations“ ([4]) definieren Kloeden und Platen zudem eine kontinuierliche Interpolante für den Zeitraum  $t$  zwischen zwei diskreten Zeitpunkten, also  $t_n \leq t \leq t_{n+1}$ .

Dieser ist gegeben durch

$$\tilde{S}_{KP}(t) = \tilde{S}_n + a_n(t - t_n) + b_n(W(t) - W_n) + \frac{1}{2}b'_n b_n((W(t) - W_n)^2 - (t - t_n)).$$

Für die Kloeden-Platen Interpolante und der eigentlichen Lösung  $S(t)$  lässt sich nun folgender Satz zeigen, wobei die verwendeten Differentialoperatoren  $L_0$  und  $L_1$  definiert seien als

$$L_0 \equiv \frac{\partial}{\partial t} + a \frac{\partial}{\partial S} \quad \text{und} \quad L_1 \equiv b \frac{\partial}{\partial S}.$$

### Satz 3.1

Angenommen für die Funktionen  $a(t, S(t))$  und  $b(t, S(t))$  der stochastischen Differentialgleichung gelten die folgenden Bedingungen:

- B1 – Lipschitz-Bedingung – es existiert eine Konstante  $K_1$ , so dass

$$|a(t, x) - a(t, y)| + |b(t, x) - b(t, y)| + |L_1 b(t, x) - L_1 b(t, y)| \leq K_1 |x - y|$$

- B2 – Lineare-Wachstum-Bedingung – es existiert ein  $K_2$ , so dass

$$\begin{aligned} |a(t, x)| + L_0 |a(t, x)| + L_1 |a(t, x)| + |b(t, x)| + L_0 |b(t, x)| \\ + L_1 |b(t, x)| + L_0 L_1 |b(t, x)| + L_1 L_1 |b(t, x)| \leq K_2 (1 + |x|) \end{aligned}$$

- B3 – Zusätzliche-Lipschitz-Bedingung – es existiert ein  $K_3$ , so dass

$$|b(t, x) - b(s, x)| \leq K_3 (1 + |x|) \sqrt{|t - s|}$$

Dann gilt für alle  $m \in \mathbb{N}^+$

$$E \left[ \sup_{0 < t < T} |S(t)|^m \right] < \infty,$$

zudem existiert zu jedem  $m \in \mathbb{N}^+$  eine Konstante  $C_m$ , so dass für die Kloeden-Platen Interpolante die Abschätzungen

$$E \left[ \sup_{0 < t < T} |S(t) - \tilde{S}_{KP}(t)|^m \right] < C_m h^m,$$

$$E \left[ \sup_{0 < t < T} |\tilde{S}_{KP}(t)|^m \right] < C_m$$

gelten.

*Beweis*

Siehe [4], Kapitel 10.

□

Die Aussage des Satzes 3.1 in Verbindung mit dem Verschiebungssatz wird uns später helfen, die Ordnungen der Einzelvarianzen  $V_l$  abzuschätzen. Für asiatische Optionen ist der Zeitraum  $[t_n, t_{n+1}]$  zwischen zwei diskreten Zeitpunkten natürlich besonders interessant, da der Wert ja über das Integral des Kurswertes bestimmt wird. Um mehr über die Verteilung der SDGL zwischen zwei Zeitpunkten zu erfahren, betrachtet man sogenannte Brownsche Brücken. Eine Brownsche Brücke ist eine Brownsche Bewegung, bei der nicht nur der Startwert, sondern auch der Endwert vorgegeben ist, zum Beispiel  $S(t_n)$  und  $S(t_{n+1})$ . Für die Konstruktion der Brownschen Brücken verwendet man hier eine stochastische Differentialgleichung deren Funktionen  $a$  und  $b$  konstant sind, womit sich die Lösung der SDGL zu

$$S(t) = S_0 + at + bW(t)$$

ergibt. Für ein  $t^*$  im Zeitintervall  $[t_n, t_{n+1}]$  der Länge  $h$  gilt dann

$$S(t^*) = S_n + \lambda(S_{n+1} - S_n) + b(W(t^*) - W_n - \lambda(W_{n+1} - W_n)) \quad (3.3)$$

mit  $\lambda = (t^* - t_n)/h$ . Das heißt, die Abweichung der Lösung  $S(t)$  von den linear interpolierten Werten  $S_n$  und  $S_{n+1}$  ist proportional zu der Abweichung der Brownschen Bewegung  $W(t)$  zu dessen linearer Interpolation. In [7] wird dazu folgendes Lemma über die Verteilung des Integrals von  $S(t)$  gezeigt.

### Lemma 3.2

Abhängig von  $S_n$  und  $S_{n+1}$  ist die Verteilung des Integrals von  $S(t)$  über das Intervall  $[t_n, t_{n+1}]$  gegeben durch

$$\int_{t_n}^{t_{n+1}} S(t) dt = \frac{1}{2}h(S_n + S_{n+1}) + bI_n,$$

wobei

$$I_n \equiv \int_{t_n}^{t_{n+1}} W(t) - W_n - \lambda(W_{n+1} - W_n) dt$$

eine  $N\left(0, \frac{1}{12}h^3\right)$ -verteilte Zufallsvariable ist, die unabhängig vom Inkrement  $W_{n+1} - W_n$  ist.

Neben der Verteilung der SDGL zwischen den diskreten Zeitpunkten sind wir zudem an der Abweichung von dem durch (3.3) ermittelten Wert und der Kloeden-Platen Interpolante interessiert. Um den entsprechenden Satz beweisen zu können, benötigt man allerdings noch zwei Ergebnisse aus der Theorie über Extremwerte von Verteilungen, welche im Folgenden ohne Beweis wie in [12] angegeben werden. Die Beweise finden sich in [12] und [14] wieder.

**Lemma 3.3**

Wenn  $Z_n$  mit  $n = 1, \dots, N$  unabhängige Zufallsvariablen einer Standardnormalverteilung sind, dann gilt für alle  $m \in \mathbb{N}^+$

$$E \left[ \max_n |Z_n|^m \right] = \mathcal{O}((\ln N)^{m/2})$$

für  $N \rightarrow \infty$ .

**Korollar 3.4**

Seien  $W_n(t)$ ,  $n = 1, \dots, N$  unabhängige Brownsche Pfade auf dem Intervall  $[0,1]$  mit den identischen Start- und Endwerten  $W_n(0) = W_n(1) = 0$ , dann gilt für alle  $m \in \mathbb{N}^+$

$$E \left[ \max_n \sup_{[0,1]} |W_n(t)|^m \right] = \mathcal{O}((\ln N)^{m/2})$$

für  $N \rightarrow \infty$ .

Die Aussagen des Lemmas 3.3 und Korollars 3.4 erlauben nun die Formulierung und Beweisführung des folgenden Satzes.

**Satz 3.5**

Es gelten B1 – B3 aus Satz 3.1, weiterhin sei  $\tilde{S}(t)$  gegeben durch

$$\tilde{S}(t) = \tilde{S}_n + \lambda(\tilde{S}_{n+1} - \tilde{S}_n) + b(W(t) - W_n - \lambda(W_{n+1} - W_n)) \quad (3.4)$$

mit  $\lambda = (t - t_n)/h$  und  $\tilde{S}_{KP}(t)$  die Kloeden-Platen Interpolante, dann gilt für alle  $m \in \mathbb{N}^+$

(i)

$$E \left[ \sup_{[0,T]} |\tilde{S}(t) - \tilde{S}_{KP}(t)|^m \right] = \mathcal{O}((h \ln(h))^m)$$

(ii)

$$\sup_{[0,T]} E \left[ |\tilde{S}(t) - \tilde{S}_{KP}(t)|^m \right] = \mathcal{O}(h^m)$$

(iii)

$$E \left[ \left( \int_0^T \tilde{S}(t) - \tilde{S}_{KP}(t) dt \right)^2 \right] = \mathcal{O}(h^3)$$

*Beweis*

Für alle drei Aussagen werden wir ausnutzen, dass der Erwartungswert  $E[\max |b'_n b_n|^m]$  existiert aufgrund von Bedingung B2 und Satz 3.1.

Ersetzt man den Wert  $\tilde{S}_{n+1}$  in  $\tilde{S}(t)$  durch die in Formel (3.2) gegebene Gleichung, lässt sich für die Differenz von  $\tilde{S}(t)$  und  $\tilde{S}_{KP}(t)$  leicht nachrechnen, dass gilt

$$\tilde{S}(t) - \tilde{S}_{KP}(t) = \frac{1}{2} b'_n b_n Y(t),$$

wobei

$$Y(t) = \lambda(W_{n+1} - W_n)^2 - (W(t) - W_n)^2.$$

Alternativ lässt sich  $Y(t)$  auch wie folgt darstellen:

$$\begin{aligned} Y(t) &= \lambda(1 - \lambda)(W_{n+1} - W_n)^2 - (W(t) - W_n - \lambda(W_{n+1} - W_n))^2 - 2\lambda(W_{n+1} - W_n)W(t) \\ &\quad - W_n - \lambda(W_{n+1} - W_n) \end{aligned}$$

In dieser Darstellung lässt sich erkennen, dass der Erwartungswert von  $Y$  aus den Zufallsvariablen  $\Delta W_n$  und  $I_n$  aus Lemma 3.2 besteht.

(i) Für die erste Aussage verwendet man die Höldersche Ungleichung:

$$\begin{aligned} E \left[ \sup_{[0, T]} |\tilde{S}(t) - \tilde{S}_{KP}(t)|^m \right] &= E \left[ \sup_{[0, T]} \left| \frac{1}{2} b'_n b_n Y(t) \right|^m \right] \\ &\leq 2^{-m} \sqrt{E \left[ \max_n |b'_n b_n|^{2m} \right] E \left[ \sup_{[0, T]} |Y(t)|^{2m} \right]} \end{aligned}$$

Der erste Erwartungswert unter der Wurzel ist, wie eingangs erwähnt, aufgrund von Satz 3.1 und Bedingung B2 endlich. Für den zweiten Erwartungswert von  $Y(t)$  verwendet man zur Abschätzung Lemma 3.3 und Korollar 3.4, womit die Behauptung folgt.

(ii) Setzt man  $W(t) - W_n = \sqrt{\lambda h} Z_1$  und  $W_{n+1} - W(t) = \sqrt{1 - \lambda h} Z_2$ , wobei  $Z_1, Z_2$  unabhängig identisch verteilte Zufallsvariablen einer Standardnormalverteilung sind, dann gilt

$$|Y| \leq h \max(Z_1^2, Z_2^2) \Rightarrow |Y|^m \leq h^m \max(Z_1^{2m}, Z_2^{2m}) \leq h^m (Z_1^{2m} + Z_2^{2m}).$$

Anwendung der Hölder-Ungleichung liefert

$$E \left[ |\tilde{S}(t) - \tilde{S}_{KP}(t)|^m \right] = 2^{-m} E[|b'_n b_n|^m] E[|Y|^m]$$

und damit mit Satz 3.1 und der obigen Abschätzung die Behauptung, da die Momente von standardnormalverteilten Zufallsvariablen die Ordnung  $\mathcal{O}(1)$  besitzen.

(iii) Für die dritte Behauptung setzt man  $X_n := \int_{t_n}^{t_{n+1}} Y(t) dt$ , womit mit der Beobachtung, dass es sich um ein Integral über die diskreten Zeitpunkte  $t_0, \dots, t_{N-1}$  handelt, folgt:

$$E \left[ \left( \int_0^T \tilde{S}(t) - \tilde{S}_{KP}(t) dt \right)^2 \right] = \frac{1}{4} E \left[ \left( \sum_{n=0}^{N-1} b'_n b_n X_n \right)^2 \right].$$

Für  $n > m$  gilt  $E[b'_m b_m X_m b'_n b_n X_n] = 0$ , da  $X_n$  unabhängig von  $b'_m b_m X_m b'_n b_n$  ist und  $E[X_n] = 0$  ist. Damit fallen alle gemischten Terme des Summenoperators weg und es gilt

$$E \left[ \left( \int_0^T \tilde{S}(t) - \tilde{S}_{KP}(t) dt \right)^2 \right] = \frac{1}{4} E[X_0^2] \sum_{n=0}^{N-1} (b'_n b_n)^2.$$

Da  $E[X_0^2] = \mathcal{O}(h^4)$  gilt für Momente von Inkrementen Brownscher Bewegungen, folgt die Behauptung. □

Mit Satz 3.1 und 3.5 ist es nun möglich die Konvergenzordnung der Varianz von europäischen und asiatischen Optionen zu bestimmen. Dazu betrachtet man in beiden Fällen die folgende Umformung der Differenz  $\tilde{S}^f(t) - \tilde{S}^g(t)$ :

$$\begin{aligned} \tilde{S}^f(t) - \tilde{S}^g(t) &= \left( \tilde{S}^f(t) - \tilde{S}_{KP}^f(t) \right) - \left( \tilde{S}^g(t) - \tilde{S}_{KP}^g(t) \right) \\ &\quad + \left( \tilde{S}_{KP}^f(t) - S(t) \right) - \left( \tilde{S}_{KP}^g(t) - S(t) \right) \end{aligned} \quad (3.5)$$

Der Fehler in den ersten beiden Termen lässt sich dabei durch Satz 3.5 abschätzen und für die hinteren zwei Differenzen kann mit Satz 3.1 eine obere Schranke angegeben werden. Bevor wir zu den Konstruktionen der Schätzer für die hier betrachteten europäischen, asiatischen und digitalen Optionen kommen, soll analog zu Kapitel 2 die optimale Wahl von  $M$  betrachtet werden.

### 3.1.1 Optimales M

Analog zur Analyse in Kapitel 2.3.1 im Zusammenhang mit dem stochastischen Euler-Verfahren betrachtete man zunächst die Gleichung:

$$\begin{aligned} V_l &= \text{Var}(\tilde{P}_l - P) + \text{Var}(P - \tilde{P}_{l-1}) + 2\text{Cov}(\tilde{P}_l - P, P - \tilde{P}_{l-1}) \\ &= \text{Var}(\tilde{P}_l - P) + \text{Var}(P - \tilde{P}_{l-1}) + 2\rho \text{Var}(\tilde{P}_l - P)^{1/2} \text{Var}(P - \tilde{P}_{l-1})^{1/2} \end{aligned}$$

Mit der Beobachtung, dass  $\tilde{S}_{KP}(t)$  für  $t = t_n$  bzw.  $t = t_{n+1}$  gerade der durch das Milstein-Verfahren approximierte Wert  $\tilde{S}_n$  bzw.  $\tilde{S}_{n+1}$  ist, lässt sich mit Theorem 3.1 und der Lipschitz-Eigenschaft folgern, dass die Abschätzung

$$\text{Var}(\tilde{P}_l - P) \leq E \left[ (\tilde{P}_l - P)^2 \right] \leq L^2 E \left[ \|\tilde{S}_{l,M^l} - S(T)\|^2 \right] \leq c_0 h_l^2$$

gilt, und damit:

$$\begin{aligned} V_l &\leq c_0 h_l^2 + c_0 h_{l-1}^2 + 2\varrho (c_0 h_l^2)^{1/2} (c_0 h_{l-1}^2)^{1/2} \\ &= c_0 h_l^2 + c_0 h_l^2 M^2 + 2\varrho (c_0 h_l^2)^{1/2} (c_0 h_l^2 M^2)^{1/2} \\ &= c_0 h_l^2 (1 + M^2 + 2\varrho M) \end{aligned}$$

Für  $\varrho = \pm 1$  folgen wieder die Grenzen

$$c_0 h_l^2 (1 + M^2 - 2M) \leq V_l \leq c_0 h_l^2 (1 + M^2 + 2M),$$

und damit ergibt sich das geometrische Mittel der beiden Schranken zu

$$V_l \approx (M^2 - 1) c_0 h_l^2.$$

Der Term  $(M^2 - 1)c_0$  entspricht wieder dem  $c_2$  aus Bedingung (iii) des Komplexitätstheorems, wobei im Fall des Milstein-Verfahrens nun  $\beta = 2$  gilt, und damit  $N_l$  gewählt wird als

$$\begin{aligned} N_l &= \left\lceil 2\epsilon^{-2} c_2 T^{(\beta-1)/2} (1 - M^{-(\beta-1)/2})^{-1} h_l^{(\beta+1)/2} \right\rceil \\ &= \left\lceil 2\epsilon^{-2} (M^2 - 1) c_0 h_l^2 T^{1/2} (1 - M^{-1/2})^{-1} h_l^{3/2} \right\rceil. \end{aligned}$$

Analog zum Fall mit dem stochastischen Euler-Verfahren lässt sich mit Hilfe der Definition des Rechenaufwands  $C$  und Wahl von  $L$  die folgende Gleichung  $f(M)$  extrahieren, die den Rechenaufwand in Verhältnis zu  $M$  setzt:

$$f(M) = (M^2 - 1) \frac{1 + M^{-1}}{(1 - M^{-1/2}) \ln M}$$

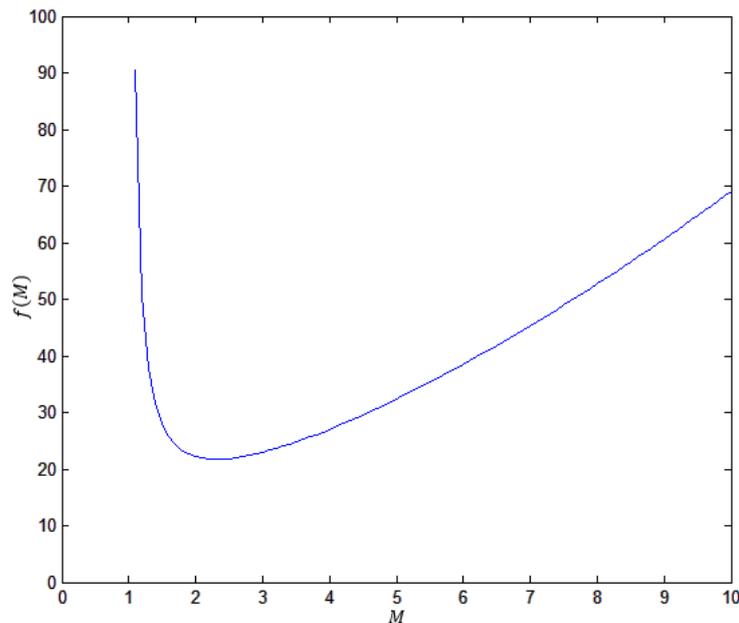


Abbildung 3.1: Graph der Funktion  $f(M)$  im Milstein-Verfahren

Der in Abbildung 3.1 gezeigte Graph der Funktion  $f(M)$  zeigt, dass unter Verwendung des Milstein-Verfahrens die Wahl  $M = 2$  optimal ist. Alle Berechnungen in den folgenden Beispielen wurden mit  $M = 2$  durchgeführt.

### 3.1.2 Europäische Optionen

Die hier betrachtete europäische Call-Option fällt unter die allgemeinere Kategorie der Finanzderivate mit Lipschitz-stetiger Auszahlungsfunktion  $f$ . Der Payoff  $P$  ist also abhängig vom Basiswert zu endlich vielen Zeitpunkten  $t_n$ ,  $n = 0, \dots, N$ :

$$P = f(S(T_0), \dots, S(T_N))$$

Für die Funktion  $f$  gilt dabei die Lipschitz-Abschätzung

$$\left| f(S_0^{(2)}, \dots, S_N^{(2)}) - f(S_0^{(1)}, \dots, S_N^{(1)}) \right| \leq L \sum_{n=0}^N |S_n^{(2)} - S_n^{(1)}|.$$

Verwendet man nun den in Kapitel 2 eingeführten multi-level Monte-Carlo Schätzer  $\tilde{Y}_l$  lässt sich folgender Satz zeigen.

#### Satz 3.6

Unter Verwendung des multi-level Schätzers aus Kapitel 2

$$\tilde{Y}_l = \frac{1}{N_l} \sum_{i=1}^{N_l} (\tilde{P}_l^{(i)} - \tilde{P}_{l-1}^{(i)})$$

besitzt die Einzelvarianz  $V_l$  im MLMC-Verfahren mit einer Lipschitz-stetigen Auszahlungsfunktion die Konvergenzordnung  $\mathcal{O}(h_l^2)$ .

#### Beweis

Unter Verwendung des Verschiebungssatzes, der Lipschitz-Eigenschaft und der Jensen-Ungleichung gilt für  $V_l$  die Abschätzung

$$\text{Var}(\tilde{P}_l^f - \tilde{P}_{l-1}^g) \leq E \left[ (\tilde{P}_l^f - \tilde{P}_{l-1}^g)^2 \right] \leq L^2 \sum_{n=0}^N E \left[ (\tilde{S}^f(t_n) - \tilde{S}^g(t_n))^2 \right].$$

Die Erwartungswerte lassen sich nun wie in Formel (3.5) umformen, wobei die durch die Quadratfunktion entstehenden gemischten Terme nach oben abgeschätzt werden, so dass gilt

$$\begin{aligned} E \left[ (\tilde{S}^f(t_n) - \tilde{S}^g(t_n))^2 \right] &\leq 4 \left( E \left[ (\tilde{S}^f(t_n) - \tilde{S}_{KP}^f(t_n))^2 \right] + E \left[ (\tilde{S}^g(t_n) - \tilde{S}_{KP}^g(t_n))^2 \right] \right. \\ &\quad \left. + E \left[ (\tilde{S}_{KP}^f(t_n) - S(t_n))^2 \right] + E \left[ (\tilde{S}_{KP}^g(t_n) - S(t_n))^2 \right] \right). \end{aligned}$$

Mit Hilfe der Sätze 3.1 und 3.5 lassen sich die einzelnen Terme abschätzen, wobei die Variable  $m$  aus den Sätzen gleich zwei ist, womit die behauptete Ordnung  $\mathcal{O}(h_l^2)$  folgt.

□

Für die folgenden numerischen Simulationen wurden die gleichen Beispieldaten wie in Kapitel 2 verwendet und der Parameter  $M$  entsprechend der vorherigen Analyse gleich zwei gesetzt.

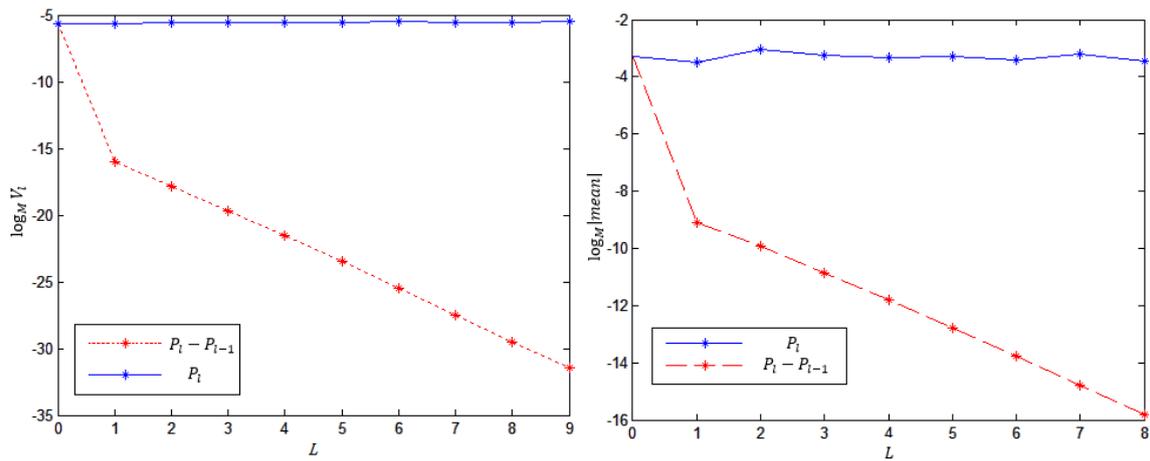


Abbildung 3.2: Varianz und Erwartungswertentwicklung der europäischen Call-Option unter Verwendung des Milstein-Verfahrens

Die im linken Plot gezeigte Entwicklung der Varianz besitzt eine Steigung nahe an -2 und bestätigt damit die theoretische Analyse, dass  $V_l$  die Ordnung  $\mathcal{O}(h_l^2)$  besitzt und der  $\beta$ -Wert aus dem Komplexitätstheorem gleich zwei ist. Das im rechten Plot abgebildete Fallen des Erwartungswerts von  $\tilde{P}_l - \tilde{P}_{l-1}$  besitzt wieder die Steigung -1, was zu der entsprechenden schwachen Konvergenzordnung von eins des Milstein-Verfahrens passt.

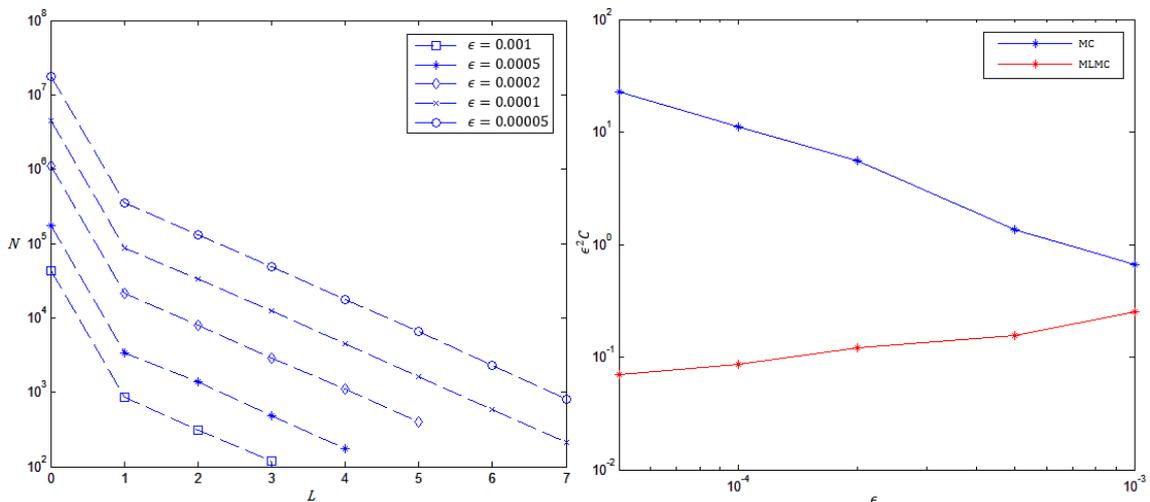


Abbildung 3.3: Anzahl an Simulationen pro Level und Rechenkosten in Abhängigkeit der Genauigkeit beim Milstein-Verfahren

Der linke Plot in Abbildung 3.3 zeigt die Anzahl der Monte-Carlo Simulationen pro Level für verschiedene Genauigkeiten  $\epsilon$ . Gut zu erkennen ist, dass im Vergleich zur MLMC Methode mit stochastischen Euler-Verfahren, die Levelanzahl schneller nach oben skaliert, da  $M = 2$  und nicht wie vorher  $M = 4$  verwendet wurde. Der rechte Plot zeigt den Rechenaufwand multipliziert mit  $\epsilon^2$  der MLMC und der einfachen MC Methode. Wie erwartet steigt der relative Rechenaufwand der normalen Monte-Carlo Methode linear bzgl.  $\epsilon^2 C$  an, während der der multi-level Methode sogar leicht sinkt. Dieses Absinken bestätigt den in Kapitel 2 vermuteten Vorteil eines numerischen Approximationsverfahrens, mit dem  $\beta > 1$  erreicht werden kann. Da der meiste Rechenaufwand in den größeren Levels liegt, kann relativ kostengünstig die Anzahl der Levels erhöht und somit eine höhere Genauigkeit erreicht werden. Natürlich erhöht sich mit kleiner werdenden  $\epsilon$  der absolute Rechenaufwand, relativ zu  $\epsilon^2 C$  gesehen, verringert er sich allerdings. Vergleicht man den Rechenaufwand der MLMC Methode unter Verwendung des stochastischen Euler-Verfahrens (Abbildung 2.4) und des Milstein-Verfahrens, zeigt sich dementsprechend, dass gerade für sehr kleine  $\epsilon$  der Einsatz eines Approximationsverfahrens mit höherer Konvergenzordnung deutliche Vorteile bringt.

### 3.1.3 Asiatische Optionen

Für die Kalkulation der asiatischen Optionen betrachtet man das Integral über den Pfad des stochastischen Prozesses, wobei nach Lemma 3.2 für den feinen Schätzer die Approximation

$$\bar{S} = T^{-1} \int_0^T S(\tau) d\tau \approx T^{-1} \sum_{n=0}^{n_T} \left( \frac{1}{2} h_l (\tilde{S}_{n+1}^f + \tilde{S}_n^f) + b_n I_n^f \right) = \bar{S}^f \quad (3.6)$$

gilt, mit  $I_n$  wie im Lemma als  $N(0, h^3/12)$ -verteilte Zufallsvariable. Für den groben Schätzer verwendet man die gleiche Definition, wobei man für  $I_n^g$  und  $M = 2$  durch Aufspalten des Integrals rechnet:

$$\begin{aligned} I_n^g &= \int_{t_n}^{t_n+2h} W(t) - W_n - \frac{t - t_n}{2h_l} (W_{n+2} - W_n) dt \\ &= I_n^f + I_{n+1}^f - \frac{1}{2} h_l (W_{n+2} + 2W_{n+1} + W_n) \end{aligned}$$

Für den feinen Pfad werden also zusätzliche Zufallsvariablen erzeugt und Resultate über Brownsche Brücken ausgenutzt, wobei anschließend der grobe Pfad aus den Informationen des feinen Pfades gewonnen wird. In [12] wird zusätzlich noch gezeigt, dass auch ohne die Terme  $I_n^f$  und  $I_n^g$  der Schätzer aus Kapitel 2 die Varianzordnung  $\mathcal{O}(h_l^2)$  hat, aber auch der obige besitzt diese Ordnung wie der folgende Satz zeigt.

**Satz 3.7**

Die Varianz des Schätzer in Formel (3.6) besitzt die Ordnung  $\mathcal{O}(h_l^2)$ .

*Beweis*

Integrieren des Erwartungswertes von (3.5) und Abschätzen der gemischten Terme liefert

$$E \left[ (\bar{S}^f - \bar{S}^g)^2 \right] \leq 4 \left( E \left[ (\bar{S}^f - \bar{S}_{KP}^f)^2 \right] + E \left[ (\bar{S}^g - \bar{S}_{KP}^g)^2 \right] + E \left[ (\bar{S}_{KP}^f - \bar{S})^2 \right] \right. \\ \left. + E \left[ (\bar{S}_{KP}^g - \bar{S})^2 \right] \right)$$

und mit den Aussagen von Satz 3.1 und 3.5 folgt die Behauptung. □

Für die folgenden numerischen Simulationen wird sowohl der in Formel (3.6) konstruierte Schätzer verwendet, als auch der aus Kapitel 2 bekannte, der auch schon für die europäische Option zum Einsatz kam. Die numerischen Ergebnisse für den eben bewiesenen Schätzer (3.6) werden mit den Ergebnissen des „einfachen“ Schätzers verglichen, um zu diskutieren, ob eine aufwendigere Konstruktion der Schätzer weitere Vorteile bringen kann.

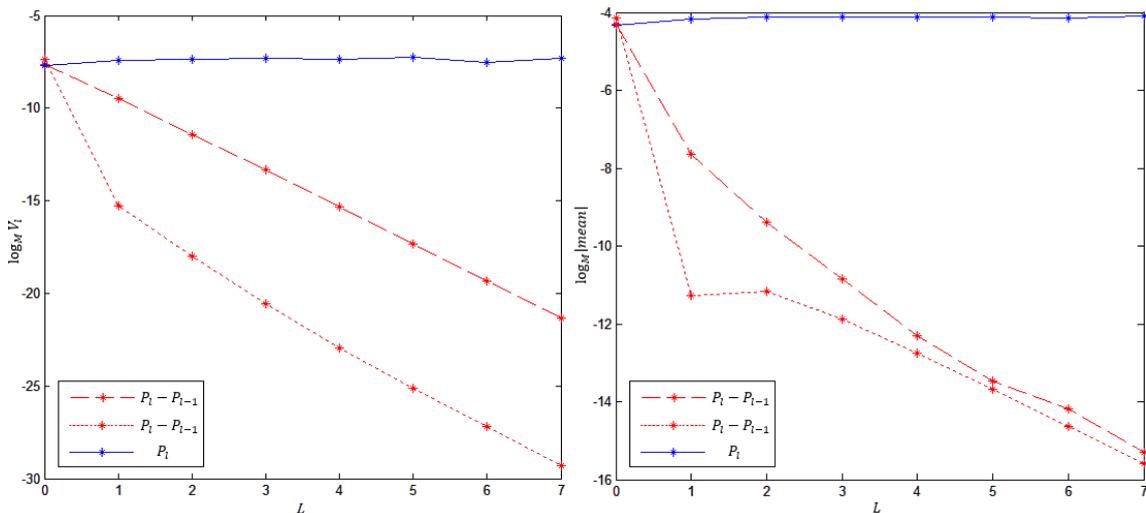


Abbildung 3.4: Varianz und Erwartungswertentwicklung der asiatischen Call-Option unter Verwendung des Milstein-Verfahrens

Die in Abbildung 3.4 gezeigten Ergebnisse der Simulation bestätigen, dass der Algorithmus für beide Schätzer die Konvergenzordnung  $\mathcal{O}(h_l^2)$  für die Varianz erbringt. Rot gestrichelt dargestellt ist der einfache und rot gepunktet der in Formel (3.6) eingeführte Schätzer. Der wesentlich stärkere initiale Abfall der Varianz des aufwendiger konstruierten Schätzers legt nahe, dass tatsächlich weitere Vorteile durch neue Schätzer erreicht werden können. Die  $\log_M$ -Korrektur ist in beiden Fällen wie beim stochastischen Euler-Verfahren anfangs etwas besser als erwartet, nimmt

aber mit steigendem Level die zur schwachen Konvergenzordnung eine korrespondierende Steigung von  $-1$  an.

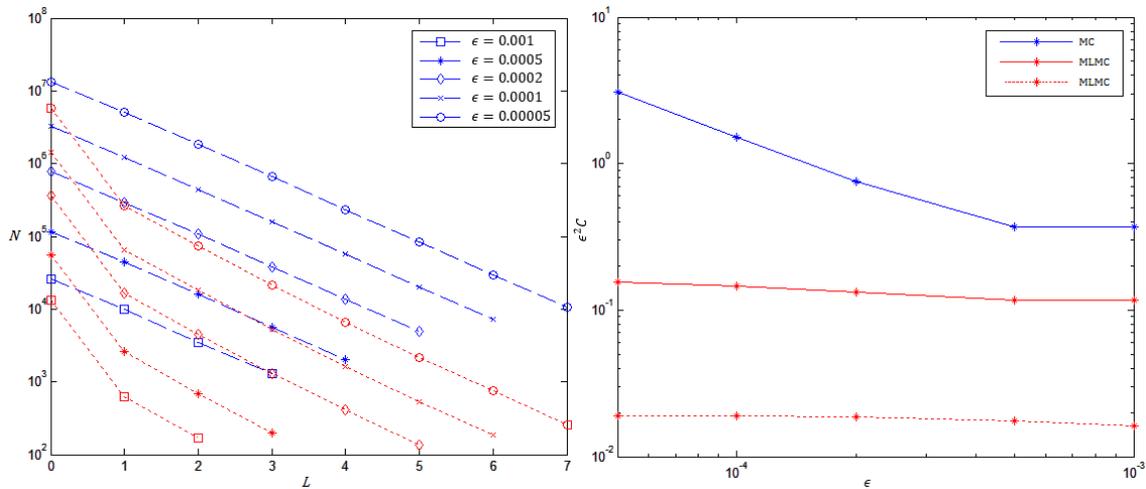


Abbildung 3.5: Anzahl an Simulationen pro Level und Rechenkosten für die asiatische Option beim Milstein-Verfahren

Die letzte Abbildung 3.5 zur asiatischen Option zeigt die Anzahl an Simulationen pro Level die für verschiedene Genauigkeiten benötigt werden und die  $\epsilon^2 C$ -Entwicklung bzgl.  $\epsilon$ . Der linke Plot der  $N_L$ s zeigt, dass der komplexere Schätzer (3.6) (rot gepunktet) für jedes getestete  $\epsilon$  nicht nur wesentlich weniger Simulationen, sondern auch zum Teil weniger Levels zum terminieren benötigt. Die im rechten Plot dargestellte Entwicklung der Rechenkosten weist dementsprechend einen deutlich geringeren Rechenaufwand für den Schätzer (3.6) aus. Für die höchste Genauigkeit von  $\epsilon = 0.00005$  benötigt der neue Schätzer nur etwa ein Zehntel des Rechenaufwands des einfachen aus Kapitel 2. Insgesamt wird in beiden Fällen klar, dass durch die Verwendung der multi-level Methode wesentlich effizienter die geforderten Fehlertoleranzen erreicht werden können als durch das einfache Monte-Carlo Verfahren.

### 3.1.4 Digitale Optionen

Wie in Kapitel 2 gezeigt wurde, verringert sich aufgrund der unstetigen Auszahlungsfunktion – und der damit verbundenen Möglichkeit, dass die Pfade der feinen und groben Approximation auf unterschiedlichen Seiten der Unstetigkeitsstelle enden – die Ordnung bei digitalen Optionen. Auch beim Milstein-Verfahren wird diese Problematik zu einer geringeren Konvergenzordnung der Varianz als bei Lipschitz-stetigen und asiatischen Optionen führen. Trotzdem kann eine beachtliche Verbesserung gegenüber der  $\mathcal{O}(h_l^{1/2})$  erreicht werden. Um den entsprechenden Satz beweisen zu können, wird noch folgendes Lemma benötigt, wobei wir schreiben  $u < h^\alpha$ , wenn  $u > 0$  und eine Konstante  $c > 0$  existiert, so dass  $u < ch^\alpha$  für alle  $h$ .

**Lemma 3.8**

Für alle  $\gamma > 0$  ist die Wahrscheinlichkeit, dass ein Brownscher Pfad  $W(t)$ , das Inkrement  $\Delta W_n$  und die dazugehörige Lösung der stochastischen Differentialgleichung  $S(t)$ , sowie die feine und grobe Approximation der Lösung  $\tilde{S}_n^f$  bzw.  $\tilde{S}_n^g$  eine der folgenden *Extrembedingungen* erfüllt gegeben durch  $\mathcal{O}(h^p)$  für alle  $p > 0$ .

$$\begin{aligned} \max_n(\max(|S(nh)|, |\tilde{S}_n^f|, |\tilde{S}_n^g|)) &> h^{-\gamma} \\ \max_n(\max(|S(nh) - \tilde{S}_n^g|, |S(nh) - \tilde{S}_n^f|, |\tilde{S}_n^f - \tilde{S}_n^g|)) &> h^{1-\gamma} \\ \max_n |\Delta W_n| &> h^{1/2-\gamma} \\ \sup_{[0,T]} |\tilde{S}^f(t) - S(t)| &> h^{1-\gamma} \\ \sup_{[0,T]} |W(t) - \bar{W}(t)| &> h^{1/2-\gamma} \end{aligned}$$

Die Variable  $\bar{W}(t)$  bezeichnet dabei die stückweise lineare Interpolation, des durch die diskreten Zeitpunkte  $W_n$  gegebene Brownschen Pfades. Zudem gelten die folgenden Gleichungen, falls keine der obigen Extrembedingungen erfüllt ist und  $\gamma > 1/2$  gilt.

$$\max_n |\tilde{S}_n^f - \tilde{S}_{n-1}^f| < h^{1/2-2\gamma} \quad (3.7)$$

$$\max_n |b_n^f - b_{n-1}^f| < h^{1/2-2\gamma} \quad (3.8)$$

$$\max_n \max(|b_n^f|, |b_n^g|) < h^{-\gamma} \quad (3.9)$$

$$\max_n |b_n^f - b_n^c| < h^{1/2-2\gamma} \quad (3.10)$$

Falls  $n$  ungerade ist, soll  $b_n^g = b_{n-1}^g$  gelten.

*Beweis*

Siehe [12], Lemma 3.6.

□

Grundsätzlich kann die Konvergenzordnung der Varianz bei digitalen Optionen von  $\mathcal{O}(h_l^{1/2})$  durch das Ersetzen des stochastischen Euler-Verfahrens mit dem Milstein-Verfahren auf  $\mathcal{O}(h_l)$  verbessert werden. Durch die Konstruktion neuer Schätzer, für die feine und grobe Approximation lässt sich allerdings die Ordnung sogar auf  $\mathcal{O}(h_l^{3/2-\delta})$  für alle  $\delta > 0$  erhöhen. Dazu simuliert man mit Hilfe des Milstein-Verfahrens alle diskreten Werte  $\tilde{S}_0, \dots, \tilde{S}_{N-1}$  der SDGL bis auf den letzten Wert  $\tilde{S}_N$ . Stattdessen setzt man am vorletzten Wert der feinen Approximation an und verwendet ab da an eine Brownsche Bewegung mit konstanter Drift- und Volatilitätsfunktion. Die

Wahrscheinlichkeit, dass für den so simulierten letzten Wert  $\tilde{S}_N^f$  dann  $\tilde{S}_N^f > K$  gilt, ist gegeben durch

$$\tilde{P}_l^f = \Phi\left(\frac{\tilde{S}_{N-1}^f + a_{N-1}^f h_l - K}{|b_{N-1}^f| \sqrt{h_l}}\right), \quad (3.11)$$

wobei  $\Phi$  die Funktion der kumulierten Normalverteilung ist. Analog definiert man den Schätzer für die grobe Approximation als

$$\tilde{P}_{l-1}^g = \Phi\left(\frac{\tilde{S}_{N-2}^g + 2a_{N-2}^g h_l + b_{N-2}^g \Delta W_{N-2} - K}{|b_{N-2}^g| \sqrt{h_l}}\right), \quad (3.12)$$

wobei  $\Delta W_{N-2}$  der Zufallsvariable entspricht, die für den vorletzten feinen Schritt verwendet wurde. Für die obigen Schätzer lässt sich nun folgender Satz formulieren.

**Satz 3.9**

Sei  $b(K, T) \neq 0$  und  $S(t)$  besitzt eine beschränkte Dichte in der Umgebung von  $K$ , dann gilt für die multi-level Schätzer (3.11) und (3.12) der digitalen Option, dass die Ordnung der Varianz  $V_l$  gegeben ist durch  $V_l = \mathcal{O}(h_l^{3/2-\delta})$  für alle  $\delta > 0$ .

*Beweisskizze*

Wir skizzieren den in [12] geführten Beweis. Zunächst unterscheidet man drei verschiedene Arten von auftretenden Pfaden und untersucht die drei Untergruppen auf ihre Auswirkung auf den Erwartungswert  $E[(\tilde{P}_l^f - \tilde{P}_{l-1}^g)^2]$ . Man betrachtet dazu:

- (i) Extreme Pfade
- (ii) Nicht-extreme Pfade für die  $|S_N - K| > h_l^{1/2-3\gamma}$  gilt
- (iii) Die restlichen Pfade die nicht in (i) oder (ii) fallen

Zu (i): Extreme Pfade sind Pfade, die mindestens eine der Extrembedingungen aus Lemma 3.8 erfüllen für  $0 < \gamma < 1/4$ . Für digitale Optionen ist sowohl  $E[(P^f)^4]$ , als auch  $E[(P^g)^4]$  endlich und damit besitzt die Auswirkung der extremen Pfade nach Lemma 3.8 die Ordnung  $\mathcal{O}(h_l^p)$  für alle  $p > 0$ . Der Beitrag der extremen Pfade zum Erwartungswert ist demnach vernachlässigbar.

Zu (ii): Sei  $\tilde{S}_N^f$  der durch die normale Milstein-Approximation erzeugte Endwert, dann gilt

$$\frac{\tilde{S}_{N-1}^f + a_{N-1}^f h_l - K}{|b_{N-1}^f| \sqrt{h_l}} = \frac{\tilde{S}_N^f - K}{|b_{N-1}^f| \sqrt{h_l}} - \frac{b_{N-1}^f}{|b_{N-1}^f| \sqrt{h_l}} \left( \Delta W_{N-1} + \frac{1}{2} (b_{N-1}^f)' ((\Delta W_{N-1})^2 - h_l) \right).$$

Ziel ist es nun, die einzelnen Terme der rechten Seite abzuschätzen. Da die hier betrachteten Pfade nicht extrem sind, können wir annehmen, dass  $\Delta W_n \leq h^{1/2-\gamma}$ ,  $|\tilde{S}^f(t) - S(t)| \leq h^{1-\gamma}$  und nach

Lemma 3.8 zudem  $|b_{N-1}^f| < h_l^{-\gamma}$  gilt. Da für die hier betrachteten Pfade  $|S_N - K| > h_l^{1/2-3\gamma}$  gilt, folgt für ein ausreichend kleines  $h$  und eine geeignete Konstante  $C > 0$ :

$$\frac{\tilde{S}_{N-1}^f + a_{N-1}^f h_l - K}{|b_{N-1}^f| \sqrt{h_l}} > C h_l^{-2\gamma}$$

Ein analoges Resultat lässt sich für die grobe Approximation  $\tilde{P}_{l-1}^g$  herleiten, womit für die Differenz  $\tilde{P}_l^f - \tilde{P}_{l-1}^g = \mathcal{O}(h_l^p)$  folgt für alle  $p > 0$ , womit wiederum auch  $E[(\tilde{P}_l^f - \tilde{P}_{l-1}^g)^2] = \mathcal{O}(h_l^p)$  für alle Pfade der Gruppe (ii) gilt und der Beitrag zum Erwartungswert somit wieder vernachlässigbar ist.

Zu (iii): Die verbleibenden Pfade sind nicht-extreme Pfade, für die  $|S_N - K| \leq h_l^{1/2-3\gamma}$  gilt. Um die im Satz behauptete Ordnung zu zeigen, nutzt man die Lipschitz-Eigenschaft der Verteilungsfunktion  $\Phi(\cdot)$  aus und schätzt die Differenz  $|\tilde{P}_l^f - \tilde{P}_{l-1}^g|$  nach oben ab. Damit lässt sich zunächst  $\tilde{P}_l^f - \tilde{P}_{l-1}^g = \mathcal{O}(h_l^{1/2-5\gamma})$  zeigen und zudem gilt  $E[\mathbb{I}_{(iii)}] = \mathcal{O}(h_l^{1/2-3\gamma})$ , aufgrund der im Satz geforderten beschränkten Dichte von  $S(t)$  in der Umgebung von  $K$ .  $E[\mathbb{I}_{(iii)}]$  entspricht dabei dem Erwartungswert der Indikatorfunktion, bzgl. der Unterscheidung, ob ein Pfad in die Gruppe (iii) fällt oder nicht. Für den Erwartungswert  $E[(\tilde{P}_l^f - \tilde{P}_{l-1}^g)^2 \mathbb{I}_{(iii)}]$  folgt insgesamt damit die Ordnung  $\mathcal{O}(h_l^{3/2-13\gamma})$  und damit mit der Wahl  $\gamma < \min(1/4, \delta/13)$  die Behauptung.

□

Für die folgenden Beispiele wurden die in Formel (3.11) und (3.12) konstruierten Schätzer und die bisherigen Beispieldaten verwendet.

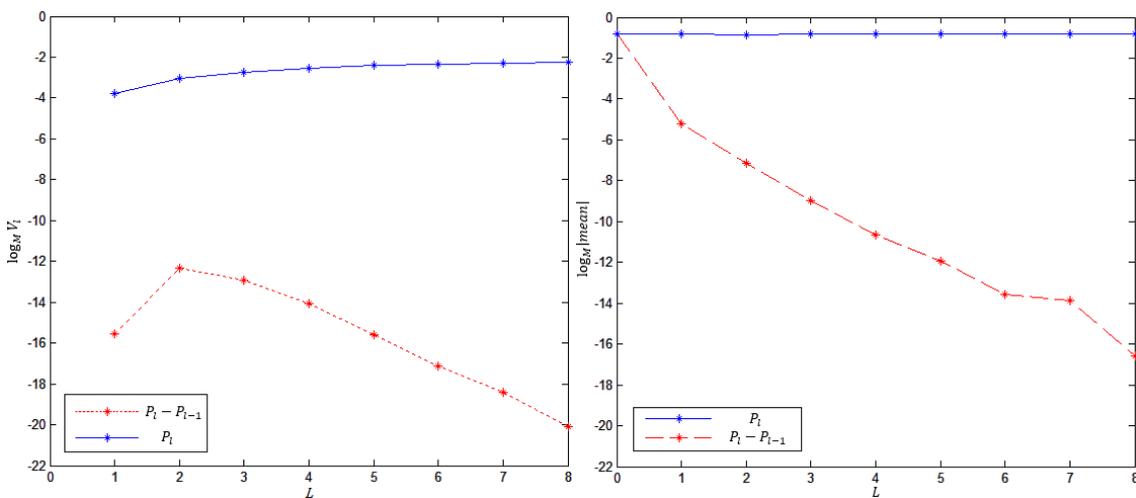


Abbildung 3.6: Entwicklung der Varianz und des Schätzers für eine digitale Option unter Verwendung des Milstein-Verfahrens und den Schätzern (3.11), (3.12)

Die im linken Plot in Abbildung 3.6 gezeigte Entwicklung der Varianz bestätigt die obige theoretische Analyse einer Konvergenzordnung von etwa  $\mathcal{O}(h_l^{3/2})$  für die späteren Levels. Das eher ungewöhnliche Verhalten der  $\log_M$ -Varianz am Anfang, lässt sich auf die Konstruktion der Schätzer zurückführen. Für  $l = 0$  wird nur ein einziger Schritt bis zum Laufzeitende  $T$  für die Simulation verwendet und damit entsprechend der obigen Herleitung nur die Formel (3.11). Ähnliches gilt für die grobe Approximation für  $l = 1$ . Tatsächlich muss die Varianz in der Implementierung für  $l = 0$  gleich null gesetzt werden, damit der Algorithmus zum richtigen Ergebnis gelangt. Das Abfallen der  $\log_M$ -Korrekturterme im rechten Plot ist dagegen wie gewohnt bei etwa  $-1$ , entsprechend der schwachen Konvergenzordnung. Zu bemerken gilt noch, dass für die beiden Plots die Mindestlevelanzahl  $L_{min}$  auf acht – statt wie bisher auf zwei – gesetzt wurde, um besser die entsprechende Entwicklung auf höheren Levels zu demonstrieren. In den vorherigen Beispielen, wurde jeweils das kleinste  $\epsilon$  für die  $\log_M V_l$ -Plots verwendet, welches auch für die Plots zur Darstellung der  $N_l$ s und der  $\epsilon^2 C$ -Kosten benutzt wurde.

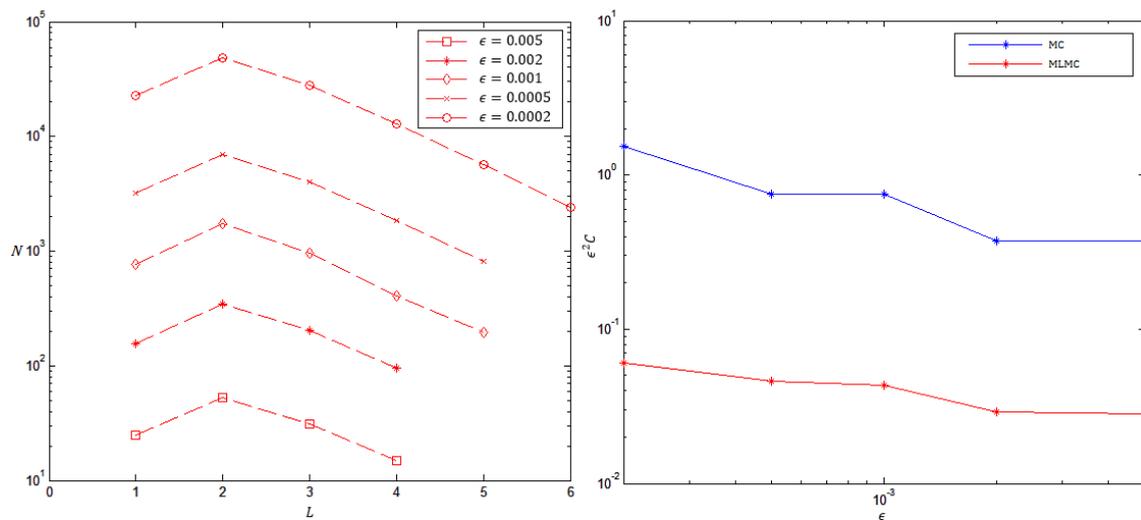


Abbildung 3.7: Anzahl an Simulationen pro Level und Veranschaulichung der Rechenkosten für die digitale Option beim Milstein-Verfahren und Schätzer (3.11),(3.12)

In der letzten Abbildung zur digitalen Option ist wieder die Entwicklung der  $N_l$ s für verschiedene  $\epsilon$ -Werte, sowie der  $\epsilon^2 C$  Kosten dargestellt. Der Grund warum für alle  $\epsilon$  der  $N_0$ -Wert nicht eingetragen ist, ist dass der Algorithmus die „optimale“ Anzahl zu null berechnet, was mit der Varianz von null in Zusammenhang steht. Natürlich wird trotzdem ein erster Schätzwert ermittelt, basierend auf der im Algorithmus eingestellten Anzahl an Startsimulationen, welche in Kapitel 2 mit  $10^4$  angesetzt wurden. Diese 10000 Erstsimulationen scheinen im Fall der digitalen Option und einem relativ großen  $\epsilon$  allerdings als zu großzügig, da zum Teil weniger als 100 Simulationen tatsächlich benötigt werden. Der Algorithmus lieferte für alle  $\epsilon$  auch mit 100 eingestellten Erstsimulationen zuverlässig das richtige Ergebnis innerhalb der Fehlertoleranz.

### 3.2 Mehrdimensionale SDGL

In Kapitel 1 wurde in der Definition der Itô-stochastischen Differentialgleichung die eindimensionale Version eingeführt und in allen darauffolgenden Betrachtungen verwendet. Natürlich gibt es auch mehrdimensionale SDGLen und darauf basierende Modelle in der Finanzmathematik. Es ergeben sich allerdings einige Schwierigkeiten, wenn man die höhere starke Konvergenzordnung des Milstein-Verfahrens ausnutzen will. Im Folgenden soll kurz die Problematik und ein möglicher Lösungsweg umrissen werden.

In ihrem Paper „Antithetic Multilevel Monte Carlo Estimation For Multi-Dimensional SDES Without Lévy Area Simulation“ ([13]) betrachten M. Giles und L. Szpruch zunächst die allgemeine Klasse mehrdimensionaler stochastischer Differentialgleichungen der Form

$$dX(t) = f(X(t))dt + g(X(t))dW(t),$$

mit  $X(t) \in \mathbb{R}^d$  für alle  $t \geq 0$  und  $f \in C^2(\mathbb{R}^d, \mathbb{R}^d)$ ,  $g \in (\mathbb{R}^d, \mathbb{R}^{d \times D})$  und der  $D$ -dimensionalen Brownschen Bewegung  $W(t)$ . Die  $i$ -te Komponente der Milstein-Approximation ist dann gegeben durch

$$\tilde{X}_{i,n+1} = \tilde{X}_{i,n} + f_i(\tilde{X}_n)h + \sum_{j=1}^D g_{ij}(\tilde{X}_n)\Delta W_{j,n} + \sum_{j,k=1}^D v_{ijk}(\tilde{X}_n)(\Delta W_{j,n}\Delta W_{k,n} - \Omega_{j,k}h - A_{jk,n}),$$

wobei  $\Omega$  die Korrelationsmatrix der Brownschen Bewegungen ist und die Terme  $v_{ijk}$  und  $A_{jk,n}$  definiert sind als

$$v_{ijk}(x) := \frac{1}{2} \sum_{l=1}^d g_{lk}(x) \frac{\partial g_{ij}}{\partial x_l}(x)$$

und

$$A_{jk,n} := \int_{t_n}^{t_{n+1}} (W_j(t) - W_j(t_n))dW_k(t) - \int_{t_n}^{t_{n+1}} (W_k(t) - W_k(t_n))dW_j(t).$$

Die Terme  $A_{jk,n}$  bezeichnet man auch als Lévy Areas und die numerische Approximation ist nur unter sehr hohem Aufwand möglich. Problematisch daran ist, dass Clark und Cameron in [15] sowie Müller-Gronbach in [16] zeigen, dass ohne die Simulation der Lévy Areas das Milstein-Verfahren keine bessere Konvergenzordnung erzielen kann, als das stochastische Euler-Verfahren.

In [13] wird eine Möglichkeit gezeigt, dieses Problem zu umgehen und auch ohne Lévy Area-Simulation die Konvergenzordnung  $\mathcal{O}(h_t^2)$  für europäische und asiatische Optionen zu erhalten.

Dazu wird die in Kapitel 1 vorgestellte Methode der antithetischen Zufallsvariablen verwendet, um die feine Approximation  $\tilde{P}_l^f$  zu verbessern und der Schätzer des  $l$ -ten Levels mit  $l > 0$  besteht dann aus den Teilen

$$\frac{1}{2}(E[\tilde{P}_l^f] + E[\tilde{P}_l^a]) - E[\tilde{P}_{l-1}^g],$$

wobei  $\tilde{P}_l^a$  der Payoff unter Verwendung der zu  $\tilde{P}_l^f$  antithetischen Zufallsvariablen ist.

In [13] wird gezeigt, dass sowohl für die von Clark und Cameron untersuchte SDGL

$$\begin{aligned} dX_1(t) &= dW_1(t) \\ dX_2(t) &= X_1(t)dW_2(t), \end{aligned}$$

als auch für das in der Finanzmathematik weit verbreitete Heston-Modell

$$\begin{aligned} dS(t) &= rS(t)dt + \sqrt{v(t)}s(t)dW_1(t) \\ dv(t) &= \kappa(\theta - v(t))dt + \sigma\sqrt{v(t)}dW_2(t), \end{aligned}$$

es so möglich ist, Schätzer für europäische und asiatische Optionen zu konstruieren, die dieselbe Konvergenzordnung wie im skalaren Fall erreichen. Für eine genauere Betrachtung der theoretischen Herleitung und der numerischen Ergebnisse sei der geneigte Leser an dieser Stelle nur auf das Paper [13] verwiesen.

## 4. Multi-level Quasi-Monte-Carlo Methoden

Eine Möglichkeit zur weiteren Verbesserung der multi-level Methode ist durch die Verwendung von Quasi-Monte-Carlo (QMC) Methoden gegeben, deren Anwendung und Grundgedanke wir in diesem abschließenden Kapitel noch kurz diskutieren wollen. Das erste publizierte Paper in dem QMC Methoden zur Bewertung von Finanzprodukten verwendet wurden, wurde 1995 veröffentlicht (siehe [17]) und überraschte mit denen der einfachen Monte-Carlo Methode deutlich überlegenen Resultaten. Seit dem erfreuen sich die verschiedenen QMC Verfahren immer größerer Beliebtheit bei der Bewertung von Finanzderivaten und die Weiterentwicklung und Analyse der Modelle ist auch heute noch Gegenstand der Forschung.

In diesem Kapitel soll zuerst das Prinzip der einfachen Quasi-Monte-Carlo Methode und einige der möglichen Umsetzungen erläutert werden. Danach wird ein Algorithmus formuliert, der den multi-level Ansatz mit dem QMC Verfahren verbindet. Der MLQMC Algorithmus wird anschließend verwendet, um den Wert einer europäischen Option zu bestimmen und die Ergebnisse werden im Vergleich zur einfachen MC und MLMC Methode diskutiert. Das hier vorgestellte Vorgehen und die Ergebnisse basieren im Wesentlichen auf [20] und [21], wobei der Großteil der zugrundeliegenden Theorie von QMC Methoden sich in [7], [18] und [19] vertieft nachvollziehen lässt.

### 4.1 Prinzip der Quasi-Monte-Carlo Methode

Ziel von Quasi-Monte-Carlo Methoden ist es, das Integral einer Funktion  $f(x)$  über den  $d$ -dimensionalen Hyperwürfel zu bestimmen. Ähnlich wie bei der normalen Monte-Carlo Methode approximiert man das Integral mittels

$$\int_{[0,1]^d} f(x) dx \approx \frac{1}{N} \sum_{i=0}^{N-1} f(x_i). \quad (4.1)$$

Der Unterschied zur gewöhnlichen Monte-Carlo Methode liegt nun darin, dass die  $x_i$  einem vorgegebenen Muster folgen, anstatt gleichverteilt aus dem Hyperwürfel zufällig gezogen zu werden. Die Hoffnung ist, dass durch eine gezielt gleichmäßigere Abdeckung des Raums durch die so generierten Zufallszahlen letztendlich eine bessere Approximation erreicht wird, als durch die Pseudozufallszahlen des Computers. Je nach Anwendung und Konstruktion der Folge der  $x_i$  kann damit erreicht werden, dass der Fehler in etwa proportional zu  $1/N$  ist, was eine wesentliche Verbesserung zu der  $1/\sqrt{N}$  Fehlerkonvergenz des normalen MC Verfahrens darstellt.

Allerdings verliert man so zum Teil auch vorteilhafte Eigenschaften, wie die Erwartungstreue des Schätzers und die Möglichkeit Konfidenzintervalle zu konstruieren. Um diese Eigenschaften zurückzuerlangen, verwendet man häufig sogenannte „*randomized Quasi-Monte-Carlo Methoden*“. Die Idee dahinter ist, in die Konstruktion der  $x_i$  ein zufälliges Element einfließen zu lassen, dass innerhalb der Schätzung – basierend auf  $N$  Simulationen – konstant ist. Durch das Berechnen mehrerer solcher Schätzer, mit jeweils unabhängigen Zufallsverschiebungen erhält man durch die Mittelwertbildung wieder einen erwartungstreuen Schätzer und die Möglichkeit Konfidenzintervalle zu konstruieren. Der Nachteil dieses Vorgehens ist natürlich einerseits ein erhöhter Rechenaufwand und andererseits eine Verringerung der durch die QMC Methode dazugewonnenen Genauigkeit.

Die verwendeten  $x_i$ -Folgen nennt man auch „*Low Discrepancy Sequences*“ (LDS), also Folgen mit geringer Diskrepanz. Die Diskrepanz ist dabei ein Maß für die Gleichverteilung der Punkte im Raum, wobei im Zusammenhang mit QMC Methoden häufig die folgende Definition der sogenannten *Sterndiskrepanz* (engl. *star discrepancy*) verwendet wird.

**Definition 4.1**

Die *Sterndiskrepanz*  $D_N^*(x_1, \dots, x_N)$  einer Menge von  $N$  Punkten ist gegeben durch

$$D_N^*(x_1, \dots, x_N) = \sup_{B \in J} \left| \frac{A(B)}{N} - \lambda(B) \right|,$$

wobei  $J$  die Menge aller Hyperrechtecken der Form

$$\prod [u_i^-, u_i^+], \quad u_i^\pm \in [0,1]$$

sei. Die Funktion  $A(B)$  gibt dabei die Anzahl aller Punkte  $x_i$  die in  $B$  liegen an und  $\lambda(B)$  das Volumen von  $B$ .

Bei Folgen mit geringer Diskrepanz lässt sich für die Sterndiskrepanz eine obere Schranke angeben, die allerdings mit steigender Dimension  $d$  des Problems wächst. Für die theoretische Analyse des Fehlers und der entsprechenden asymptotischen Abschätzung über die Koksma-Hlawka Ungleichung, sei hier nur auf die eingangs erwähnte Literatur verwiesen. Zwei sehr bekannte Vertreter von solchen LDS sind zum Beispiel die „*rank-1 lattice rule*“ und die *Sobol-Sequenz*.

Die  $x_i$  der rank-1 lattice rule besitzen die eher simple Konstruktion

$$x_i = \left\{ \frac{i}{N} z \right\}, \tag{4.2}$$

wobei mit dem Ausdruck  $\{\cdot\}$  die Funktion beschrieben ist, die den ganzzahligen Teil eines Wertes abzieht und nur den Ausdruck hinter dem Komma übrig lässt, also  $\{x\} = x - [x]$ . Die Variable  $z$

ist ein spezieller  $d$ -dimensionaler *generierender Vektor* mit ganzzahligen Einträgen, wobei diese gerade so gewählt werden, dass jeder Eintrag relativ prim, bzw. teilerfremd zu  $N$  ist.

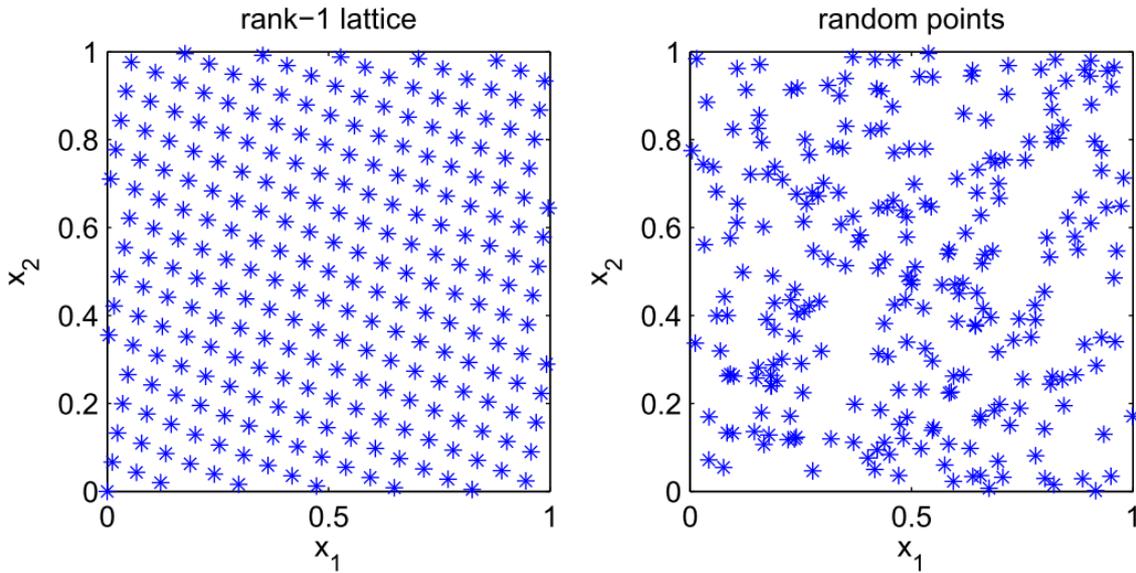


Abbildung 4.1: Darstellung der ersten beiden Dimensionen einer rank-1 lattice rule im Vergleich zu pseudozufälligen Punkten wie in [21]

In Abbildung 4.1 sind die ersten beiden Dimensionen des Vektors  $x_i$  dargestellt, einmal erzeugt über die rank-1 lattice rule für die Quasi-Monte-Carlo Methode und die pseudozufällig gezogenen Punkte des Computers für das normale MC Verfahren. Im rechten Plot ist gut zu erkennen, dass sich durch die gleichverteilten Zufallswerte des Computers Cluster bilden, während bei der rank-1 lattice rule eine gleichmäßige Abdeckung der beiden Dimensionen entsteht.

Die randomized-QMC Variante dieses Verfahrens hätte dann die Gestalt

$$x_i = \left\{ \frac{i}{N} z + \Delta \right\}, \quad (4.3)$$

wobei  $\Delta$  ein  $d$ -dimensionaler Vektor aus dem  $[0,1)^d$  ist. Durch die Verwendung von  $q$  unabhängigen Simulationen mit jeweils eigenem zufälligen Verschiebungsvektor  $\Delta$ , entsteht dann durch das arithmetische Mittel der Schätzungen ein erwartungstreuer Schätzer. Genauere Details zur Implementierung und der Wahl des Vektors  $z$  lassen sich in [7] Kapitel 5.3 wiederfinden, wo vor allem auf die sogenannte Korobov-Regel eingegangen wird, bei der der Vektor  $z$  die Form

$$z = (1, a, a^2, \dots, a^{d-1})^T$$

für ein  $a \in \mathbb{N}$  annimmt.

Eine der weitverbreitetsten LDS ist die Sobol-Seqenz, die wesentlich aufwendiger in der Konstruktion ist als die rank-1 lattice rule. Die genaue Herleitung von Sobol-Sequenzen lässt sich u.a. in [7] Kapitel 5.2.3 nachvollziehen. Für die meisten Programmiersprachen gibt es Bibliotheken, in denen entsprechende Generatoren für Sobol-Sequenzen enthalten sind. In MATLAB lässt sich eine  $(m \times n)$ -Matrix  $Q$  von  $n$   $m$ -dimensionalen Sobol-Sequenzen über die Befehlsreihe

```
Q = grandstream('sobol',n);
```

```
grand(Q,m)
```

generieren. Eine randomisierte Sobol-Seqenz lässt sich über das sogenannte „digital scrambling“ erzeugen, bei dem eine Bitweise angewandte „exclusive-or“ Operation verwendet wird. Dadurch ist es möglich, die Folgen zu randomisieren, aber gleichzeitig die essenziellen Eigenschaften der Sobol-Punkte zu erhalten. Das genaue vorgehen lässt sich in [7] und [20] nachlesen, in MATLAB lässt sich mittels

```
S = sobolset(n);
```

```
S = scramble(S,'MatousekAffineOwen');
```

```
Q = grandstream(Q);
```

```
grand(Q,m)
```

eine so randomisierte Sobol-Seqenz generieren.

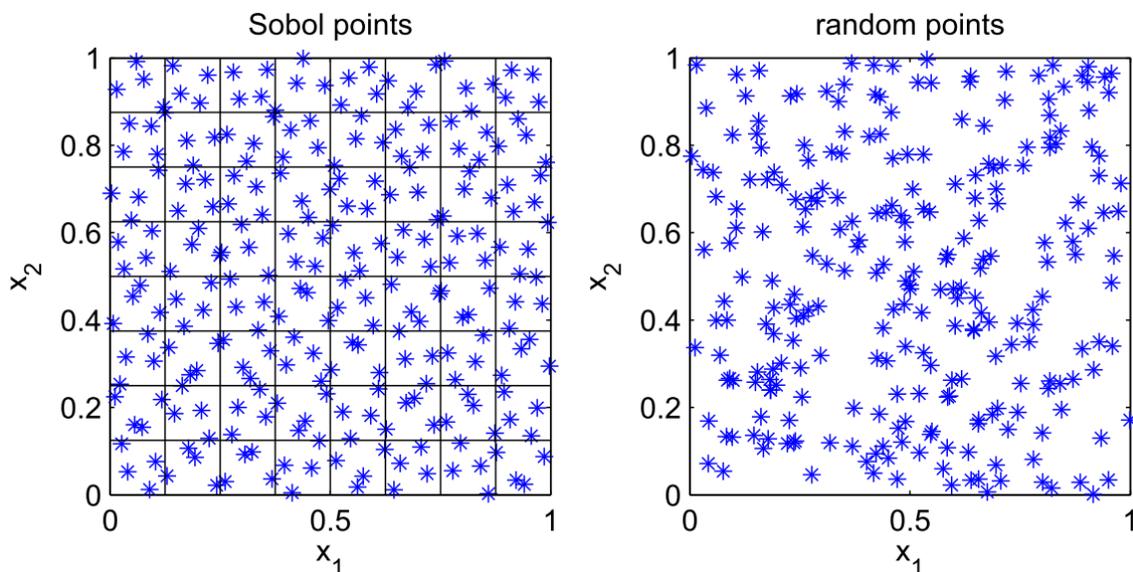


Abbildung 4.2: Darstellung der ersten beiden Dimensionen einer Sobol-Seqenz im Vergleich zu pseudozufälligen Punkten wie in [21]

Links in Abbildung 4.2 sind die ersten beiden Dimensionen von 256 generierten Sobol-Punkten dargestellt und rechts die zufällig vom Computer gezogenen. Man beachte, dass im linken Plot der Sobol-Punkte in jedem kleinen Quadrat der Größe  $0.1 \times 0.1$  jeweils genau vier Punkte liegen

und damit sichergestellt ist, dass trotz eingeschränkt möglicher Clusterbildung es keine größeren Leerflächen gibt.

Nach der Konstruktion der LDS stellt sich natürlich noch die Frage, wie das Problem der Hyperwürfel-Integration auf die Bewertung einer Option projiziert werden kann. Für eine skalare SDGL, die mit Hilfe eines Diskretisierungsverfahrens durch  $n_T$  Punkte approximiert wird, ist die Dimension  $d$  des Problems gleich  $n_T$  und der Erwartungswert entspricht dem Integral

$$\int_{\mathbb{R}^d} f(x) \frac{\exp\left(-\frac{1}{2}x^T \Sigma^{-1}x\right)}{(2\pi)^{d/2} \sqrt{\det \Sigma}} dx.$$

Die Funktion  $f(x)$  ist dabei die Payoff-Funktion des Derivats in Abhängigkeit des Vektors  $x$ , dessen Elemente die in die Diskretisierung eingehenden Werte des Wiener-Pfades sind. Die  $d$ -dimensionale Matrix  $\Sigma$  entspricht der Kovarianz-Matrix der Elemente von  $x$  und besitzt die Einträge  $\Sigma_{i,j} = \min(t_i, t_j)$ . Um dieses Integral auf die Integration des Hyperwürfels umzuformulieren, substituiert man  $x = Ay$  und  $y = \Phi^{-1}(z)$ , wobei für die Matrix  $A$  gilt  $AA^T = \Sigma$  und  $\Phi^{-1}$  die auf jedes Element des Vektors  $z$  angewendete inverse der kumulierten Normalverteilungsfunktion ist. Mit Hilfe dieser Umformungen folgt dann

$$\int_{\mathbb{R}^d} f(Ay) \frac{\exp\left(-\frac{1}{2}y^T y\right)}{(2\pi)^{d/2}} dy = \int_{[0,1]^d} f(A\Phi^{-1}(z)) dz.$$

Für die Matrix  $A$  kann jede Matrix verwendet werden, die der Bedingung  $AA^T = \Sigma$  genügt. Der einfachste Ansatz wäre den Cholesky-Faktor von  $\Sigma$  zu verwenden, womit die in den  $n$ -ten Schritt eingehende Zufallsvariable  $\Delta W_n$  definiert ist durch die  $n$ -te Komponente von  $x_i$ :

$$\Delta W_n = \sqrt{h} \Phi^{-1}(x_{i,n})$$

Dieser Ansatz, der auch als Standardmethode bezeichnet wird, wird auch in dem Beispiel in Kapitel 4.2.2 verwendet.

Eine Alternative dazu wäre zum Beispiel die Verwendung einer Brownschen Brücke, bei der das erste Element von  $x$  verwendet wird um  $W(T)$  zu definieren, das zweite für  $W(T/2)$  und so weiter. Diese Methode wurde zum Beispiel für die in [20] angegebenen numerischen Ergebnisse verwendet.

Eine weitere bekannte Möglichkeit ist die sogenannte „*Principal Components Analysis*“ (PCA) Methode, bei der die Spalten der Matrix  $A$  durch  $\sqrt{\lambda_n} v_n$  gegeben sind, wobei die  $\lambda_n$ , die nach der Größe geordneten Eigenwerte der Matrix  $\Sigma$  und  $v_n$  die korrespondierenden Eigenvektoren sind.

## 4.2 Multi-level Ansatz für QMC Methoden

Um den multi-level Ansatz mit der Quasi-Monte-Carlo Methode zu verbinden, wird ein etwas abgeänderter Algorithmus als der in Kapitel 2 vorgestellte Algorithmus 2.3 benötigt. Im Folgenden wollen wir diesen neuen MLQMC Algorithmus zuerst formulieren und dann die dazugehörigen Simulationsergebnisse der Bewertung einer europäischen Option diskutieren.

### 4.2.1 MLQMC Algorithmus

Wie bereits erwähnt, ist es lohnenswert eine randomisierte Quasi-Monte-Carlo Methode zu verwenden, um bestimmte Eigenschaften wieder zurückzugewinnen. Dafür führen wir die neue Variable  $N_{sets}$  ein, die die Anzahl der QMC-Sets angibt, die auf jedem Level  $l$  berechnet werden. Das arithmetische Mittel der  $N_{sets}$  eines Levels ergibt dann wieder den Gesamtschätzer des  $l$ -ten Levels. Um den MSE unter die gewünschte  $\epsilon^2$ -Schranke zu bringen, erhöht man einerseits die an geeigneter Stelle die Anzahl der Simulationen so lange, bis die Summe der Einzelvarianzen kleiner als  $\epsilon^2/2$  ist, und andererseits schätzt man den verbleibenden Bias mittels

$$\max\left(\frac{1}{2}|\tilde{Y}_{L-1}|, |\tilde{Y}_L|\right). \quad (4.4)$$

Der Bias wird wie vorher durch die Erhöhung der Levelanzahl  $L$  verringert. Der neue MLQMC Algorithmus lässt sich dann in die folgenden Schritte zusammenfassen.

#### Algorithmus 4.2

*Schritt 1:* Starte mit  $L = 0$

*Schritt 2:* Berechne  $V_L$  und  $\tilde{Y}_L$  basierend auf  $N_{sets}$  randomisierten QMC-Sets und mittels  $N_l = 1$  Simulationen

*Schritt 3:* Solange  $\sum_{l=0}^L V_l > \epsilon^2/2$  gilt, verdoppele die Simulationsanzahl  $N_l$  auf dem Level mit dem höchsten  $V_l/(2^l N_l)$ -Wert und berechne  $V_l$  und  $\tilde{Y}_l$  neu

*Schritt 4:* Falls  $L < L_{min}$ , oder der Bias-Schätzer (4.4) größer als  $\epsilon/\sqrt{2}$  ist, setze  $L = L + 1$  und gehe zu *Schritt 2*

Das Verdoppeln der Simulationsanzahl  $N_l$  in Schritt 3 zielt darauf ab, möglichst viel Varianzreduktion pro zusätzlichen Rechenaufwand zu erhalten. Man beachte, dass, im Gegensatz zur normalen multi-level Monte-Carlo Methode, in Schritt 3 nicht zusätzliche Simulationen berechnet werden, die dann den alten hinzugefügt werden können, sondern, dass das komplette Level neu berechnet werden muss. Der Grund dafür liegt darin, dass die generierten Punkte einer LDS aufeinander aufbauen und neu erzeugte Punkte somit nur den gleichen Bereich abdecken. Man

könnte zwar einen Puffer aufbauen, indem von vornherein mehr QMC Punkte als benötigt erzeugt und gespeichert werden, was allerdings zu einem enormen Speicherbedarf führt, da für alle Sets eines jeden Levels die gepufferten Sequenzen gespeichert werden müssten.

### Bemerkung 4.3

Die in [20] verwendete Formel zur Berechnung der Rechenkosten lautet

$$C = N_{sets} \sum_l 2^l N_l, \quad (4.5)$$

wobei  $2^l = M^l$  gilt, da wie im vorherigen Kapitel die Schrittweite für jedes Level verdoppelt wird. Es ist auffällig, dass die anfallenden Kosten durch die Neuberechnung beim Verdoppeln der Simulationsanzahl  $N_l$  nicht berücksichtigt wurden. Die präsentierten Ergebnisse bzgl. des Rechenaufwands sind also nur unter Vorbehalt, bzw. unter Beachtung eines extrem hohen Speicherbedarfs, zu betrachten. Da  $N_l$  aufgrund der Konstruktionsweise des Algorithmus einer Zweierpotenz entspricht und bekanntlich  $2^k = 2^{k-1} + \dots + 2^0 + 1$  für alle  $k \in \mathbb{N}$  gilt, wäre

$$C = N_{sets} \sum_l 2^{l+1} N_l$$

eine realistischere Näherung, sofern alle QMC Reihen neu berechnet werden müssen.

### 4.2.2 Beispiel

Im Rahmen der Masterarbeit wurde der MLQMC Algorithmus 4.2 in MATLAB implementiert, wobei für die QMC Punkte erzeugende LDS eine scrambled Sobol-Seqenz verwendet wurde. Die folgenden Ergebnisse wurden alle mit 32 randomisierten QMC Sets durchgeführt und dem Milstein-Verfahren als numerische Approximationsmethode. Zum Vergleich der Varianz dienen die Ergebnisse zur europäischen Option aus Kapitel 3. Die Beispieldaten sind wie bisher durch  $r = 0.05$ ,  $\sigma = 0.2$  und  $T = K = S_0 = 1$  gegeben.

Im linken Plot der Abbildung 4.3 ist neben der  $\log_M V_l$ -Entwicklung der einfachen Monte-Carlo Methode und der MLMC Methode unter Verwendung des Milstein-Verfahrens, zusätzlich die des MLQMC Verfahrens eingezeichnet. Die MLQMC Varianzwerte wurden wie in [20] mit der Anzahl der Sets multipliziert, da die berechneten  $V_l$  als Durchschnitt der 32 Sets berechnet wurden. Es lässt sich erkennen, dass die MLQMC Methode eine durchgehend geringere Varianz als das normale multi-level Verfahren aufweist und zudem ein Level weniger für das Erreichen der Genauigkeit von  $\epsilon = 0.00005$  benötigt. Der rechte Plot zeigt die Entwicklung der  $E[\tilde{P}_{l-1} - \tilde{P}_l]$  Terme, die korrespondierend zur schwachen Konvergenzordnung von  $\alpha = 1$  mit  $-1$  abfallen. Dies bestätigt – neben der richtigen Bepreisung der Option – die korrekte Arbeitsweise des Algorithmus.

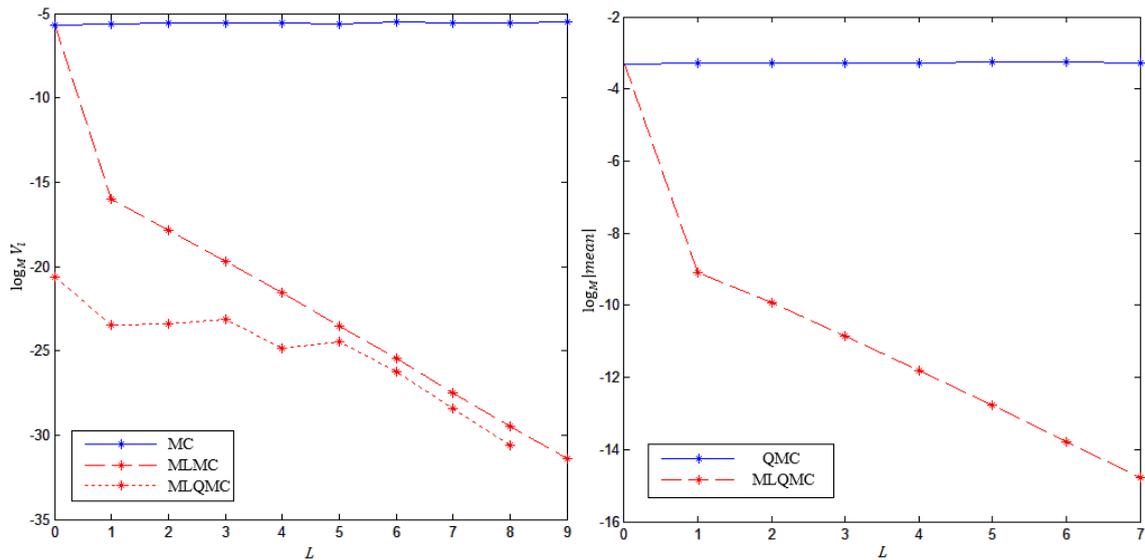


Abbildung 4.3: Links die Entwicklung der Varianz von  $\tilde{P}_l$  der normalen MC Methode und  $\tilde{P}_{l-1} - \tilde{P}_l$  der MLMC und MLQMC Methode, rechts die Entwicklung von  $E[\tilde{P}_l]$  und  $E[\tilde{P}_{l-1} - \tilde{P}_l]$

In dem linken Plot der unteren Abbildung 4.4 ist die Anzahl der Simulationen pro Level für verschiedene  $\epsilon$  abgetragen. Interessant ist, da alle  $N_l$  einer Zweierpotenz entsprechen, ist der Rückgang der Simulationsanzahl nicht mehr streng monoton wie in den vorherigen Beispielen. In einigen seltenen Fällen (nicht abgebildet) konnte sogar beobachtet werden, dass die Anzahl an benötigten Simulationen für zwei aufeinanderfolgende Levels ansteigt. Im Normalfall ergab sich jedoch ein ähnliches Bild wie bei den vorherigen Methoden.

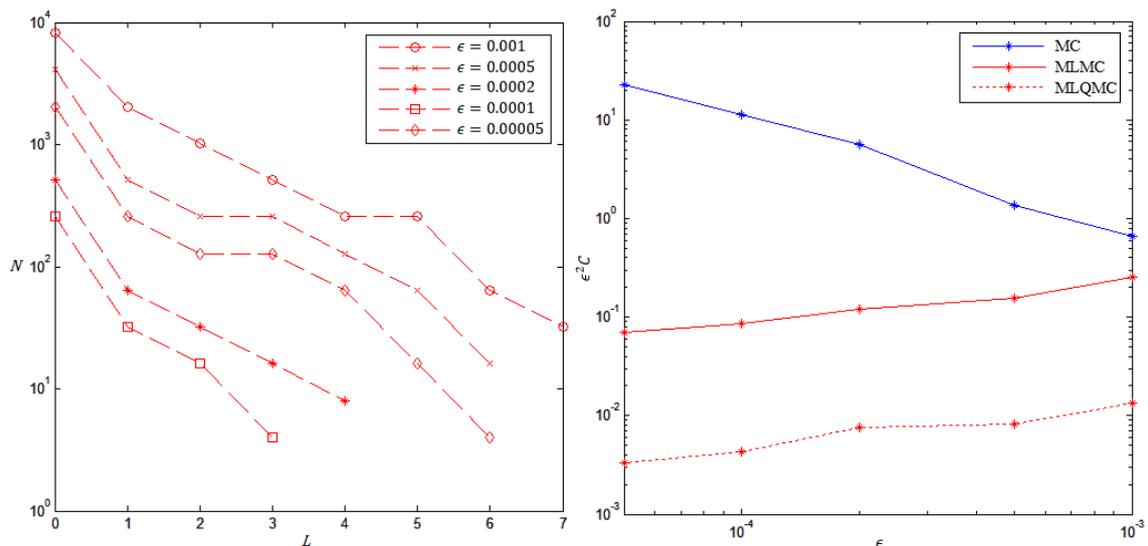


Abbildung 4.4: Links die Anzahl der Simulationen pro Level für verschiedene Genauigkeiten  $\epsilon$ , rechts Darstellung des Rechenaufwands im Vergleich zur MLMC und MC Methode

Der rechte Plot zeigt, dass die MLQMC Methode ein ähnliches Verhalten bzgl. des  $\epsilon^2 C$ -Rechenaufwands aufweist wie die einfache multi-level Monte-Carlo Methode. Der Rechenaufwand  $C$  wurde dabei über die Formel (4.5) ermittelt. Insgesamt lässt sich für alle vorgegebenen MSEs ein deutlich geringerer Rechenaufwand feststellen und die Erweiterung der MLMC Methode auf einen multi-level Ansatz basierend auf einer Quasi-Monte-Carlo Methode als lohnenswerte Verbesserung.

Die analogen numerischen Ergebnisse zur asiatischen und digitalen Option sind in [20] abgebildet, wobei sich in beiden Fällen ähnlich positive Effekte wie bei der europäischen Option beobachten lassen.

## 5. Fazit und Ausblick

Wir haben in dieser Arbeit sowohl theoretisch, als auch numerisch gezeigt, dass durch die Verwendung eines multi-level Ansatzes der Rechenaufwand für die Bewertung verschiedener Optionskontrakte wesentlich verringert werden kann. Der bewiesene Hauptsatz dieser Arbeit – das Komplexitätstheorem – zeigt, dass unter Verwendung des stochastischen Euler-Verfahrens der Rechenaufwand so von  $\mathcal{O}(\epsilon^{-3})$  auf  $\mathcal{O}(\epsilon^{-2} \ln(\epsilon)^2)$  gesenkt werden kann. Eine weitere Verbesserung kann durch den Einsatz des Milstein-Verfahrens erreicht werden, welches eine höhere starke Konvergenzordnung besitzt und wodurch sich der Rechenaufwand auf  $\mathcal{O}(\epsilon^{-2})$  reduzieren lässt. In allen untersuchten Optionskontrakten konnte auch numerisch eine deutliche Verbesserung der Konvergenzgeschwindigkeit gegenüber der einfachen Monte-Carlo Methode nachgewiesen werden. Insgesamt lässt sich also der große Nachteil der langsamen Konvergenzgeschwindigkeit der einfachen Monte-Carlo Methode verringern und gleichzeitig der Vorteil des flexiblen Einsatzes erhalten. Zudem ist die multi-level Monte-Carlo Methode ebenso erweiterbar wie das einfache Monte-Carlo Verfahren und kann zum Beispiel durch den Einsatz von Quasi-Monte-Carlo Methoden, oder antithetischen Zufallsvariablen weiter verbessert werden.

Die Möglichkeiten für weitere Forschung und Weiterentwicklungen der multi-level Monte-Carlo Methode sind reichlich. Einerseits bemerkten wir in Kapitel 2, dass der Algorithmus aufgrund der Bias-Schätzung heuristischer Natur ist und es noch nicht klar ist, ob eine Mindestlevelanzahl  $L_{min}$  existiert, so dass der vorgegebene MSE bewiesenermaßen eingehalten wird. In Kapitel 3 sahen wir zudem, dass durch den Einsatz komplexerer Schätzer zwar keine bessere Konvergenzordnung der Einzelvarianzen erreicht werden kann, wohl aber eine insgesamt niedrigere Varianz und somit weitere Einsparungen bei den Gesamtrechenkosten. Die Konstruktion dieser Schätzer ist sicherlich ein interessantes Themengebiet. Das größte Gebiet für weitere Forschung liegt in der Erweiterung auf mehrdimensionale stochastische Differentialgleichungen bei der Kursmodellierung des Basiswerts. M. Giles und L. Szpruch zeigen in [13], dass durch den Einsatz von antithetischen Zufallsvariablen die Möglichkeit besteht, den Vorteil des Milstein-Verfahrens auch in mehrdimensionalen Modellen zu behalten, allerdings fehlt die entsprechende Theorie für einige komplexere Finanztitel wie digitale oder Barriere-Optionen noch.

## Literaturverzeichnis

- [1] L. Grüne, *Numerische Methoden der Finanzmathematik*, Skript zur Vorlesung, Universität Bayreuth, 2015
- [2] J. Hull, *Options, Futures and Other Derivatives*, Pearson Prentice Hall, 6th Edition, 2005
- [3] Wikipedia.de: <https://de.wikipedia.org/wiki/Wiener-Prozess#/media/File:Wienerprozess.png>, Zugriff am 24.08.2016 um 16:00 Uhr.
- [4] P. E. Kloeden and E. Platen, *Numerical Solution of Stochastic Differential Equations*, Springer-Verlag, 1992
- [5] Wikipedia.de: [https://de.wikipedia.org/wiki/Geometrische\\_brownsche\\_Bewegung#/media/File:Geometrische\\_Brownsche\\_Bewegung.png](https://de.wikipedia.org/wiki/Geometrische_brownsche_Bewegung#/media/File:Geometrische_Brownsche_Bewegung.png), Zugriff am 24.08.2016 um 16:00 Uhr
- [6] R. Korn and E. Korn and Gerald Kroisandt, *Monte-Carlo Methods and Models in Finance and Insurance*, CRC Press, 2010
- [7] P. Glasserman, *Monte Carlo Methods in Financial Engineering*, Springer Science & Business Media, 2004
- [8] D. Duffie and P. Glynn, *Efficient Monte Carlo Simulation of Security Prices*, The Annals of Applied Probability, 1995
- [9] M. B. Giles, *Multilevel Monte Carlo path simulation*, Operations Research, Vol. 56, No. 3, pp. 607-617, 2008.
- [10] M. Ohlberger, *Numerische Analysis*, Skript zur Vorlesung, Westfälische Wilhelms-Universität Münster, 2010
- [11] M. B. Giles, *Improved multilevel Monte Carlo convergence using the Milstein scheme*, In: Keller, Alexander; Heinrich, Stefan; Niederreiter, Harald: Monte Carlo and Quasi-Monte Carlo Methods 2006, Springer, 2008
- [12] M. B. Giles and K. Debrabant and A. Rößler, *Numerical Analysis of multilevel Monte Carlo path simulation using the Milstein discretisation*, ArXiv preprint, 2013
- [13] M. B. Giles and L. Szpruch. *Antithetic multilevel Monte Carlo estimation for multi-dimensional SDEs without Lévy area simulation*, Annals of Applied Probability, 24(4):1585-1620, 2014
- [14] P. Embrechts and C. Klüppelberg and T. Mikosch, *Modelling Extremal Events: for Insurance and Finance*, Springer, 2008

- 
- [15] J. M. C. Clark and R. J. Cameron, *The maximum rate of convergence of discrete approximations for stochastic differential equations*, In: Stochastic Differential Systems (Proc. IFIP–WG 7/1 Working Conf., Vilnius, 1978) Lecture Notes in Control and Information Sci. 25 162–171. Springer, 1980
- [16] T. Muller-Gronbach, *Strong approximation of systems of stochastic differential equations*, Habilitation thesis, TU, Darmstadt., 2002
- [17] S. H. Paskov und J. F. Traub, *Faster valuation of financial derivatives*, In: J. Portfolio Management, 22(1) pp. 113-120, 1995
- [18] H. Niederreiter, *Random Number Generation and Quasi-Monte Carlo Methods*, SIAM, 1992
- [19] M. Rometsch, *Quasi-Monte Carlo Methods in Finance With Application to Optimal Asset Allocation*, Diplomica Verlag, 2008
- [20] M.B. Giles and B.J. Waterhouse, *Multilevel quasi-Monte Carlo path simulation*, pp.165-181 in Advanced Financial Modelling, in Radon Series on Computational and Applied Mathematics, de Gruyter, 2009
- [21] M. B. Giles, <http://people.maths.ox.ac.uk/~gilesm/mc2013/> - *Lectures on Monte-Carlo Methods*, 2013

## Inhalt der DVD-ROM

Für die numerischen Simulationen dieser Masterarbeit wurden mehrere Programme in MATLAB programmiert. Die im Folgenden angegebenen Dateien befinden sich auf dem beigelegten Datenträger.

Masterarbeit_Sascha_Pezoldt.pdf	Die Masterarbeit in .pdf Format
MLMC.m	Hauptroutine der MLMC Methode
MC_antithetisch.m	Antithetische MC Methode zur Berechnung einer europäischen Call-Option
MC_MLMC_Asia.m	Level 1 Schätzer für einen asiatischen Call basierend auf der Euler-Maruyama-Approximation
MC_MLMC_Asia_Milstein.m	Level 1 Schätzer für einen asiatischen Call basierend auf der Milstein-Approximation und unter Verwendung des Schätzers (3.6)
MC_MLMC_Asia_Milstein_TEST.m	Level 1 Schätzer für einen asiatischen Call basierend auf der Milstein-Approximation und unter Verwendung des einfachen Schätzers
MC_MLMC_Dig.m	Level 1 Schätzer für einen digitalen Call basierend auf der Euler-Maruyama-Approximation
MC_MLMC_Dig_Milstein.m	Level 1 Schätzer für einen digitalen Call basierend auf der Milstein-Approximation und unter Verwendung des Schätzers (3.11)
MLQMC.m	Hauptroutine des MLQMC Algorithmus (Kapitel 4)
mlqmc_1.m	Level 1 Quasi-Monte-Carlo Schätzer für einen europäischen Call basierend auf der Milstein-Approximation
qmc.m	QMC Methode zur Bewertung einer europäischen Call-Option
ReadMe MATLAB Eingabe.txt	Beispieleingaben für MATLAB

# Erklärung

Die vorliegende Arbeit wurde von mir selbstständig verfasst und es wurden keine anderen Hilfsmittel als die angegebenen Quellen benutzt. Die Arbeit wurde nicht in gleicher, oder ähnlicher Form bereits zur Erlangung eines akademischen Grades eingereicht.

Bayreuth, den 29. August 2016

---

Sascha Pezoldt