

WESTFÄLISCHE  
WILHELMS-UNIVERSITÄT  
MÜNSTER

# DAS MULTILEVEL MONTE CARLO VERFAHREN UND SEINE ANWENDUNG IM LIBOR MARKTMODELL

MASTERARBEIT

Vorgelegt von:  
*Patrick Wolff*

Westfälische Wilhelms-Universität Münster  
Fachbereich Mathematik und Informatik  
Institut für Mathematische Statistik

Betreuung:  
*PD Dr. Volkert Paulsen*

Münster, 28. Oktober 2013

## **Eidesstattliche Erklärung**

Hiermit versichere ich, Patrick Wolff, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe. Sinngemäß oder wörtlich Übernommenes habe ich durch Angabe von Herkunft oder Anmerkung belegt bzw. kenntlich gemacht. Darüber hinausgehend versichere ich, dass ich die verwendeten Abbildungen sowie die zur numerischen Bewertung notwendigen „Matlab“-Codes selbst erstellt habe.

Münster, den 28. Oktober 2013

# Inhaltsverzeichnis

<b>1 Einleitung</b>	<b>1</b>
<b>2 Eine Einführung in Monte Carlo Simulationen</b>	<b>3</b>
<b>3 Das Multilevel Monte Carlo Verfahren</b>	<b>6</b>
3.1 Die Idee des Verfahrens . . . . .	6
3.2 Fehlerschranken und Rechenaufwand . . . . .	8
3.3 Zentraler Satz . . . . .	12
3.4 Anwendung in der Praxis . . . . .	18
3.4.1 Biasschätzung . . . . .	18
3.4.2 Die optimale Anzahl an Pfadsimulationen . . . . .	19
3.5 Der Multilevel Monte Carlo Algorithmus . . . . .	20
<b>4 Das Libor Marktmodell</b>	<b>21</b>
4.1 Einleitung . . . . .	21
4.2 Mathematisches Modell . . . . .	21
4.3 Die Euler-Maruyama Diskretisierung im Libor Marktmodell . . . . .	26
<b>5 Das Milstein Verfahren</b>	<b>27</b>
5.1 Das Milstein Verfahren im Allgemeinen . . . . .	27
5.2 Das Milstein Verfahren ohne „Lévy area“-Simulation . . . . .	30
5.2.1 Antithetische Variablen . . . . .	31
5.2.2 Ein verallgemeinerter Multilevel Monte Carlo Schätzer . . . . .	32
5.2.3 Der „Antithetische“ Multilevel Monte Carlo Schätzer . . . . .	33
5.3 Das Milstein Verfahren im Libor Marktmodell . . . . .	36
<b>6 Zinsderivate</b>	<b>39</b>
6.1 Grundlegende Zinsderivate . . . . .	39
6.1.1 Swaps und Swaptions . . . . .	39
6.1.2 Caps und Floors . . . . .	42
6.2 CMS Caps und Floors . . . . .	43
6.3 TARNs . . . . .	45
<b>7 Bewertung von Zinsderivaten im Libor Marktmodell</b>	<b>48</b>
7.1 Anwendung des zentralen Satzes . . . . .	48
7.2 Vorstellung der produktunabhängigen Algorithmen . . . . .	50
7.2.1 Allgemeines Setting und Parameterübergabe . . . . .	50
7.2.2 Simulation der Libor-Raten-Pfade (Euler-Maruyama Diskretisierung) . . . . .	51
7.2.3 Der Multilevel Monte Carlo Algorithmus zur Preisbestimmung von Derivaten im Libor Marktmodell . . . . .	54
7.3 Vorstellung der produktabhängigen Algorithmen . . . . .	57
7.3.1 Bewertung von CMS Caps . . . . .	57
7.3.2 Bewertung von TARNs . . . . .	60
7.4 Simulation der Libor-Raten-Pfade (Milstein Diskretisierung) . . . . .	64

<b>8</b>	<b>Numerische Ergebnisse</b>	<b>65</b>
8.1	Marktkalibrierung . . . . .	65
8.2	Bewertung von Caplets . . . . .	68
8.3	Bewertung von CMS Caps und TARNs . . . . .	70
8.4	Vergleich der numerischen Verfahren . . . . .	72
<b>9</b>	<b>Fazit und Ausblick</b>	<b>79</b>
<b>A</b>	<b>Anhang</b>	<b>81</b>
A.1	Marktdaten . . . . .	81
<b>Literaturverzeichnis</b>		<b>83</b>

## 1 Einleitung

Auf den weltweiten Finanzmärkten wird eine Vielzahl an Finanzderivaten gehandelt, deren Wert sich nicht durch eine explizite Bewertungsformel bestimmen lässt. Aus diesem Grund haben sich numerische Bewertungsmethoden, insbesondere die Monte Carlo Simulationsverfahren, mit denen ein fairer Preis von Finanzderivaten bestimmt werden kann, zu einem wichtigen Bewertungsinstrument entwickelt. Außerdem bieten diese Verfahren ein hohes Maß an Flexibilität, welches aufgrund der Entwicklung immer neuer und individueller Finanzderivate erforderlich ist. Andererseits entstehen mit der Verwendung von numerischen Bewertungsmethoden zur Preisbestimmung von Finanzderivaten natürlich auch Nachteile. Das größte Problem liegt darin, dass ein hoher Grad an numerischer Genauigkeit der Preise meist auch einen hohen numerischen Rechenaufwand mit sich bringt. Dies erfordert also eine gründliche Abwägung zwischen hoher Präzision auf der einen Seite und einem niedrigen Rechenaufwand auf der anderen Seite. Damit stellt sich nun die Frage wie sich der Rechenaufwand von Monte Carlo Verfahren verringern lässt, während die Genauigkeit erhalten bleibt. Eine Antwort bietet das sogenannte Multilevel Monte Carlo Verfahren, bei dem durch die Variation der Diskretisierungsschrittweiten bei der Simulation ein deutlicher Varianzreduktionseffekt und damit eine Verringerung der Anzahl der notwendigen Simulationen erzielt werden kann.

Aus dieser Motivation heraus werden wir im nachfolgenden zweiten Kapitel dieser Arbeit zunächst eine Einführung in das „klassische“ Monte Carlo Simulationsverfahren geben, um im dritten Kapitel das sogenannte Multilevel Monte Carlo Verfahren einzuführen. Insbesondere werden wir zeigen, inwiefern der Rechenaufwand bei gleichbleibender Genauigkeit verringert werden kann. Dabei wird zunächst exemplarisch die Idee und Vorgehensweise des Verfahrens dargelegt. Anschließend beweisen wir den zentralen Satz, welcher theoretisch zusichert, dass das Verfahren den gewünschten Effekt erzielen kann. Zum Abschluss des Kapitels werden wir einige Hinweise zur allgemeinen Anwendbarkeit des Satzes in der Praxis liefern. Im anschließenden vierten Kapitel werden wir das Libor Marktmodell einführen, um einen Modellrahmen für die Anwendung des Multilevel Monte Carlo Verfahrens zu schaffen. In einem solchen Zinsstrukturmodell kann dann eine Vielzahl an Zinsderivaten bewertet werden, was unter anderem besonders reizvoll ist, da verschiedenste Zinsderivate vor allem für Banken eine sehr große Bedeutung haben. Dies liegt wiederum daran, dass viele (exotische) Derivate den Banken und ihren Kunden als innovative, individuelle und kosteneffektive Werkzeuge zur Anpassung bzw. zur Realisierung der eigenen Risikovorstellungen dienen. Deshalb werden wir nach einem Exkurs über das Milstein Diskretisierungsverfahren zur Verbesserung der Konvergenzgeschwindigkeit des Multilevel Verfahrens, welcher in Kapitel 5 durchgeführt wird, im sechsten Kapitel der Arbeit eine Einführung in die wichtigsten grundlegenden Zinsderivate geben. Darauf aufbauend werden wir in diesem Kapitel die zentralen Zinsprodukte einführen, deren Bewertung im Anwendungsteil dieser Arbeit angestrebt wird. Für die ausgewählten Derivate, die sogenannten CMS Caps und TARNs, liegt keine geschlossene Bewer-

---

tungsformel vor, was den Einsatz des Multilevel Monte Carlo Verfahrens besonders interessant macht.

Im siebten Kapitel werden wir noch einmal auf die Anwendung des zentralen Satzes aus Kapitel 3 in der speziellen Situation der Bewertung von Zinsderivaten im Libor Marktmodell eingehen und anschließend die zur Bewertung der verschiedenen vorgestellten Produkte notwendigen Algorithmen vorstellen.

Die zugehörigen numerischen Ergebnisse werden dann im achten Kapitel präsentiert, um nachfolgend die verschiedenen in dieser Arbeit vorgestellten Simulationsverfahren zu vergleichen und numerisch zu zeigen, dass mithilfe des Multilevel Verfahrens eine erhebliche Verringerung des Rechenaufwands erzielt werden kann. Zum Abschluss der Arbeit werden wir ein Fazit der erhaltenen Ergebnisse ziehen sowie Forschungsansätze zur weiteren Verbesserung der Verfahren aufzeigen.

Die vorliegende Arbeit wurde in Kooperation mit dem Unternehmen „zeb/information.technology“ geschrieben. Deshalb gilt vor allem Herrn Christoph Moll ein besonderer Dank für die hervorragende Betreuung und Unterstützung. Des Weiteren möchte ich Herrn PD Dr. Volkert Paulsen für die sehr gute universitäre Betreuung der Arbeit danken.

## 2 Eine Einführung in Monte Carlo Simulationen

Zu Beginn dieser Arbeit wollen wir eine kurze Einführung in die generelle Idee und Vorgehensweise von Monte Carlo Simulationen geben, um daran anknüpfend eine verbesserte Variante dieser Methoden, das Multilevel Monte Carlo Verfahren, vorzustellen und analysieren zu können.

Für die gesamte Arbeit werden wir davon ausgehen, dass ein Wahrscheinlichkeitsraum  $(\Omega, (\mathcal{F}_t)_t, \mathbb{Q})$  mit einer Wiener-Filtration  $\mathcal{F}$  vorliegt. Ferner soll der Preisprozess eines zu untersuchenden „underlying assets“  $(X(t))_t$  in diesem Abschnitt und obiger Situation durch die allgemeine  $d$ -dimensionale stochastische Differentialgleichung (SDE)

$$dX(t) = a(X(t), t)dt + b(X(t), t)dW(t), 0 \leq t \leq T \quad (2.1)$$

beschrieben werden können.  $(W(t))_{0 \leq t \leq T}$  stelle dabei einen  $m$ -dimensionalen Wiener-Prozess bzgl. des Wahrscheinlichkeitsmaßes  $\mathbb{Q}$  dar und erzeuge die obige Wiener-Filtration  $\mathcal{F}$ , sodass diese die „usual conditions“ (siehe z.B. [Sch11]) erfüllt.  $a : \mathbb{R}^d \times \mathbb{R} \rightarrow \mathbb{R}^d$  sowie  $b : \mathbb{R}^d \times \mathbb{R} \rightarrow \mathbb{R}^{d \times m}$  seien messbare Funktionen, die vom „underlying asset“  $X$  und der Zeitvariablen  $t$  abhängig sind.

Davon ausgehend ist es das Ziel, einen Erwartungswert der Form  $\mathbb{E}[f(X(T))]$  zu bestimmen, wobei  $f$  eine Funktion mit Werten in  $\mathbb{R}$  darstelle und der Erwartungswert  $\mathbb{E}$  bzgl. eines zu  $\mathbb{Q}$  äquivalenten Martingalmaßes  $\mathbb{P}$  bestimmt werden soll. Können wir diesen Erwartungswert ohne zu großen Aufwand analytisch bestimmen, werden dazu keine Monte Carlo Methoden benötigt; allerdings ist diese analytische Erwartungswertbestimmung nicht immer möglich. Wegen der vergleichsweise relativ einfachen numerischen Umsetzung sind folglich Monte Carlo Methoden, welche in diesen Situationen zur Bestimmung der Erwartungswerte herangezogen werden können, in der Finanzmathematik sehr beliebt.

Wollen wir den Preis eines Derivats bestimmen, dessen Payoff vom Wert des „underlying assets“ zum Zeitpunkt  $T$  abhängt, so ist dies oft über die Bestimmung eines Erwartungswerts der Form  $\mathbb{E}[e^{-rT}\bar{f}(X(T), T)]$  möglich. Dabei bezeichne  $\bar{f}(X(T), T)$  den Payoff des Derivats zum Zeitpunkt  $T$ , welcher mittels Multiplikation mit  $e^{-rT}$  und einem (hier der Einfachheit halber konstanten) Zinssatz  $r$  auf den heutigen Zeitpunkt „abgezinst“ wird.

Indem dann mithilfe der Erzeugung von Zufallszahlen, sogenannten Samples, Werte für  $X(T)$  simuliert werden, kann der obige Erwartungswert durch den Monte Carlo Schätzer

$$\hat{E}_N = e^{-rT} \frac{1}{N} \sum_{i=1}^N \bar{f}(X_T^{(i)}, T)$$

für ein  $N \in \mathbb{N}$  approximiert werden, wobei  $X_T^{(i)}$  den Wert des „underlying assets“ der  $i$ -ten Simulation zum Zeitpunkt  $T$  darstelle. Wie wir  $X_T^{(i)}$  simulieren, hängt dabei natürlich sowohl von der Wahl des Modells und somit von der den Werteprozess beschreibenden SDE als auch vom Diskretisierungsverfahren ab, worauf wir später näher eingehen werden. Die Konvergenz von  $\hat{E}_N$  gegen  $\mathbb{E}[e^{-rT}\bar{f}(X(T), T)]$

---

für  $N \rightarrow \infty$  sichert uns dabei für integrierbares  $\bar{f}$  das starke Gesetz der großen Zahlen zu.

Darüber hinaus können wir bei Monte Carlo Methoden zwischen statischen und dynamischen Monte Carlo Simulationen unterscheiden. Bei statischen Methoden werden – wie oben – Zufallsvariablen, bei dynamischen Methoden hingegen ganze Prozesse bzw. diskretisierte Versionen von stochastischen Prozessen über ein gewisses Zeitintervall  $[0, T]$  hinweg simuliert. Dies bedeutet, dass der Payoff eines Derivats nicht nur vom Wert des „underlying assets“ zu einem bestimmten Zeitpunkt, sondern auch von einer Vielzahl von Werten des „underlying assets“ zu verschiedenen Zeitpunkten oder sogar von einem kompletten Pfad  $(X(t))_t$  des stochastischen Prozesses abhängen kann.

Kommen wir nun wieder zur SDE (2.1) zurück, durch welche dann auch das Modell, welches den Verlauf des „underlying assets“ beschreibt, festgelegt wird.

Für die Durchführung der Pfadsimulationen benötigen wir zunächst ein Diskretisierungsverfahren, zum Beispiel das Euler-Maruyama Verfahren, bei welchem der Pfad der  $i$ -ten Komponente eines stochastischen Prozesses ausgehend von einem Startwert  $X_0^{(i)} := X^{(i)}(t_0)$  an  $\eta \in \mathbb{N}$  vorher festgelegten Diskretisierungsstellen des zu betrachtenden Zeitintervalls durch

$$X_{n+1}^{(i)} = X_n^{(i)} + a^{(i)}(X_n, t_n)(t_{n+1} - t_n) + \sum_{k=1}^m b^{(i,k)}(X_n, t_n) \Delta W_n^{(k)}, \quad n \in \{0, 1, \dots, \eta-1\}$$

approximiert wird. Dabei bezeichne  $\Delta W_n^{(k)}$  die  $k$ -te Komponente eines Wiener-Inkrementes  $\Delta W_n = W(t_{n+1}) - W(t_n)$ . Diskretisieren können wir bei diesem (Standard-) Diskretisierungsverfahren beispielsweise mit einer konstanten Schrittweite  $t_{n+1} - t_n = h$  für alle  $n \geq 0$ .

Ein einfacher Schätzer  $\hat{Y}$  für  $Y := \mathbb{E}[f(X(T), T)]$  ist dann gegeben durch den Mittelwert der Auszahlungswerte  $f(X_{T/h}, T)$ , welche von  $N$  unabhängigen Pfadsimulationen stammen, also durch

$$\hat{Y} = \frac{1}{N} \sum_{i=1}^N f(X_{T/h}^{(i)}, T).$$

Dafür nehmen wir zunächst an, dass  $f$  eine skalare Lipschitz-Funktion ist, d.h., dass ein  $c > 0$  existiert mit  $|f(x) - f(y)| \leq c\|x - y\|$  für alle  $x, y \in \mathbb{R}$ . Wenn  $a(X(t), t)$  und  $b(X(t), t)$  in dieser Situation gewisse Bedingungen erfüllen, auf die wir in Kapitel 4.2 eingehen werden, dann ist der asymptotische Fehler, der „mean square error“ (MSE), des Schätzers  $\hat{Y}$  ungefähr

$$\text{MSE}(\hat{Y}) \approx c_1 N^{-1} + c_2 h^2$$

mit Konstanten  $c_1, c_2 > 0$ . Um das Verhalten dieses Approximationsfehlers weiter untersuchen zu können, wollen wir an die Definition des Landau-Symbols aus der Numerik erinnern sowie den Rechenaufwand definieren.

**Definition 2.1** (Landau-Symbol)

Seien  $g$  und  $h$  reellwertige Funktionen  $g, h : \mathbb{R} \rightarrow \mathbb{R}$ . Dann heißt  $g$  von der Ordnung  $h(t)$ , also  $g(t) = \mathcal{O}(h(t))$  für  $t \rightarrow 0$ , genau dann, wenn ein  $c > 0$  und ein  $\delta > 0$  existieren, so dass  $|g(t)| \leq c|h(t)|$  für alle  $|t| < \delta$  ist.

**Definition 2.2** (Rechenaufwand)

Wir definieren den numerischen Rechenaufwand als die Anzahl von insgesamt durchgeführten Simulationsschritten bei der Erzeugung von Samplepfaden.

Diese Definition des Rechenaufwands führt dazu, dass eine Simulation von  $N$  Pfaden mit jeweils  $\eta$  Diskretisierungsstellen beispielsweise einen Rechenaufwand von  $N \cdot \eta$  Simulationsschritten erfordert.

Betrachten wir den MSE der Schätzung nun noch einmal genauer, so sehen wir, dass der gesamte Fehler des Monte Carlo Schätzers  $\hat{Y}$  wegen

$$\text{MSE}(\hat{Y}) = \mathbb{E}[|Y - \hat{Y}|^2] = \mathbb{V}[Y - \hat{Y}] + \mathbb{E}[Y - \hat{Y}]^2 = \mathbb{V}[\hat{Y}] + \mathbb{E}[Y - \hat{Y}]^2$$

in zwei verschiedene Fehlerterme aufgeteilt werden kann.

Der erste Summand im obigen MSE des Schätzers  $\hat{Y}$  entsteht durch die Varianz von  $\hat{Y}$  aufgrund der Monte Carlo Simulation. Mit dem zentralen Grenzwertsatz gilt nämlich für eine quadratisch integrierbare Funktion  $f$

$$\sqrt{N}(\hat{Y} - \mathbb{E}[f(X(T), T)]) \xrightarrow{d} \mathcal{N}(0, \sigma_f^2),$$

wobei  $\sigma_f^2$  die Varianz von  $f(X(T), T)$  darstelle und die Konvergenz in Verteilung gemeint ist. Daraus folgt dann wiederum

$$\hat{Y} - \mathbb{E}[f(X(T), T)] \xrightarrow{d} \frac{\sigma_f}{\sqrt{N}} \mathcal{N}(0, 1).$$

Für die Varianz von  $\hat{Y}$  erhalten wir damit

$$\mathbb{V}[\hat{Y}] = \mathbb{V}[\hat{Y} - \mathbb{E}[f(X(T), T)]] \approx \frac{\sigma_f^2}{N},$$

was folglich für  $N \rightarrow \infty$  der Ordnung  $\mathcal{O}(\frac{1}{N})$  entspricht.

Der zweite Fehlerterm entsteht wegen des notwendigen Einsatzes des Diskretisierungsverfahrens, also der Approximation des eigentlichen stochastischen Prozesses mittels des Euler-Maruyama Verfahrens. Dieses liefert, wie sich mithilfe einer (stochastischen) Taylorentwicklung (siehe [KP92]) zeigen lässt, einen Fehler – meist als Bias bezeichnet – der Ordnung  $\mathcal{O}(h)$ , wobei  $h$  wie oben die fest gewählte Schrittweite der Diskretisierung bezeichnet. Da wir hier den quadratischen Fehler betrachten, ergibt sich für diesen Fehlerterm folglich eine Ordnung von  $\mathcal{O}(h^2)$ .

Um nun insgesamt einen MSE der Ordnung  $\mathcal{O}(\epsilon^2)$  für ein gegebenes  $\epsilon > 0$  zu erreichen, müssen wir  $N$  von der Größenordnung  $\mathcal{O}(\epsilon^{-2})$  sowie  $h$  von der Größenordnung  $\mathcal{O}(\epsilon)$  wählen. Definition 2.2 liefert für das (Standard) Monte Carlo Verfahren bei Benutzung des Euler-Maruyama Diskretisierungsverfahrens dann einen Rechenaufwand von  $N \cdot \frac{1}{h}$ , was der Ordnung  $\mathcal{O}(\epsilon^{-3})$  entspricht.

---

### 3 Das Multilevel Monte Carlo Verfahren

#### 3.1 Die Idee des Verfahrens

Wie wir im vorherigen Kapitel gesehen haben, liefert ein klassisches Monte Carlo Verfahren einen Rechenaufwand der Ordnung  $\mathcal{O}(\epsilon^{-3})$ , wenn wir davon ausgehen, dass ein MSE der Ordnung  $\mathcal{O}(\epsilon^2)$  eingehalten werden soll. Das Ziel dieser Arbeit besteht darin, diesen im Vergleich zu anderen Verfahren relativ hohen Rechenaufwand, welcher folglich den Schwachpunkt von Monte Carlo Simulationen darstellt, durch den Einsatz eines verbesserten Verfahrens zu verringern.

Für einen einfachen Anwendungsfall, bei dem eine Lipschitz-stetige Payoff-Funktion  $f$  vorliegt und die SDE mithilfe des Euler-Maruyama Verfahrens diskretisiert wird, ist es beispielsweise möglich, den Rechenaufwand auf die Ordnung  $\mathcal{O}(\epsilon^{-2} \ln(\epsilon)^2)$  zu verringern, wenn wir das sogenannte Multilevel Monte Carlo Verfahren benutzen. Wie genau dies funktionieren kann, werden wir im Folgenden in Anlehnung an [Gil08a] zeigen.

Die Idee des Verfahrens, welches Giles in [Gil08a] vorstellt, ist es, die Samplepfade für die Simulation nicht alle mit derselben festen Schrittweite  $h$  zu diskretisieren, sondern die Schrittweite variabel zu machen. Dies bedeutet, dass verschiedene Diskretisierungslevel für die Pfadsimulationen genutzt werden, also, dass Samplepfade mit verschiedenen Diskretisierungsschrittweiten  $h_l := M^{-l}T$ ,  $l = 0, \dots, L$ ,  $M \in \mathbb{N}^{\geq 2}$ , simuliert werden, wobei  $L \in \mathbb{N}^{>0}$  später weiter spezifiziert wird.

Um dies näher zu betrachten, werden wir den Payoff  $f(X(T), T)$  eines Derivats für eine zufällige Realisierung des zugrunde liegenden stochastischen Prozesses  $X$ , welcher im Allgemeinen als Lösung der SDE (2.1) gegeben sei und somit von einer zufälligen Wiener-Trajektorie  $(W(t))_{0 \leq t \leq T}$  abhängt, der Einfachheit halber mit  $P$  bezeichnen. Auf die Voraussetzungen für die Existenz und Eindeutigkeit einer solchen Lösung  $X$  der SDE (2.1) werden wir später näher eingehen und diese hier zunächst annehmen.

Die Zufallsvariablen, die je eine Approximation für  $X(T)$  bzw. für  $P$  darstellen und durch numerische Pfaddiskretisierungen mit Schrittweite  $h_l = M^{-l}T$  bestimmt werden, bezeichnen wir im Weiteren mit  $\tilde{X}_T^l$  bzw.  $\tilde{P}_l$ .

Das Besondere an dem von Giles vorgestellten Verfahren ist nun, für jeden der Simulationslevel  $l > 0$  nicht nur einen einfachen Monte Carlo Schätzer für die Payoffs zu verwenden, sondern einen Schätzer für die Differenz der Payoffs von Pfadapproximationen zweier aufeinanderfolgender Level zu nutzen. Dabei wird für jeden Level  $l > 0$  zunächst – wie bei Monte Carlo Simulationen üblich – eine Wiener-Trajektorie  $(W_l^f(t))_t$  bzgl. der Schrittweite  $h_l$  simuliert, welche dann zur Bestimmung eines Samplepayoffs, d.h. einer Realisierung von  $\tilde{P}_l$ , genutzt werden kann. Anschließend wird dieselbe Wiener-Trajektorie jedoch zusätzlich genutzt, um eine Realisierung des Payoffs  $\tilde{P}_{l-1}$  einer Pfadapproximation bzgl. der Schrittweite  $h_{l-1}$  zu bestimmen. Dazu werden je  $M$  aufeinanderfolgende Wiener-Inkremeante, welche bzgl. der feineren der beiden Schrittweiten bestimmt wurden, zusammengefasst und als Wiener Inkrement bzgl. der größeren Diskretisierungsschrittweite genutzt, um so eine diskretisierte Trajektorie  $(W_{l-1}^g(t))_t$  zu erhalten. Die Indizes

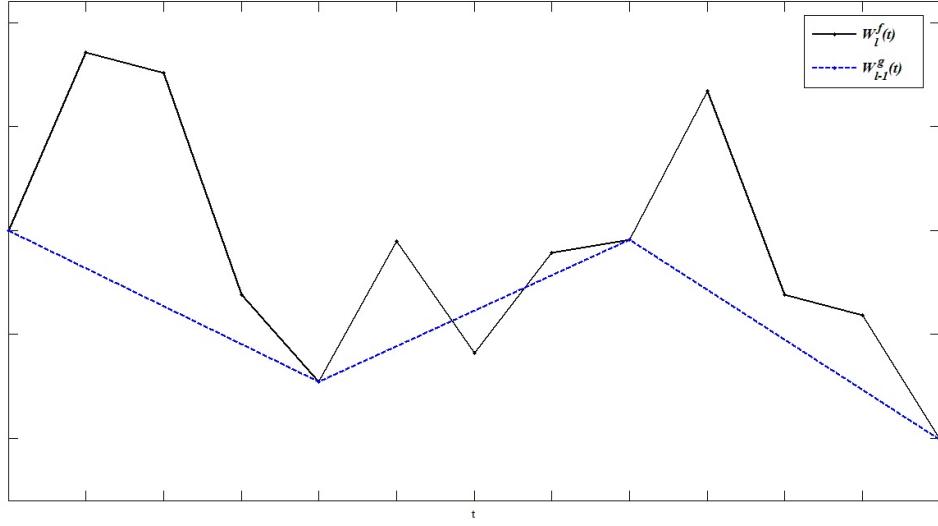


Abbildung 1: Diskretisierte Wiener-Prozesse bzgl. zweier aufeinanderfolgender Diskretisierungsschrittweiten eines Simulationslevels (Verfeinerungsfaktor  $M=4$ )

$f$  bzw.  $g$  geben somit an, ob es sich bei der jeweiligen Wiener-Trajektorie um die bzgl. der feineren Schrittweite erzeugte oder um die davon abgeleitete Trajektorie bzgl. der größeren der beiden Schrittweiten handelt. Dieses Vorgehen bei der Bestimmung der Wiener-Trajektorien eines Simulationslevels soll in Abbildung 1 verdeutlicht werden.

Offensichtlich gilt nun wegen der Linearität des Erwartungswertes

$$\mathbb{E}[\tilde{P}_L] = \mathbb{E}[\tilde{P}_0] + \sum_{l=1}^L E[\tilde{P}_l - \tilde{P}_{l-1}], \quad (3.1)$$

wobei  $\tilde{P}_L$  den bzgl. der Approximation mit der kleinsten Schrittweite  $h_L$  bestimmten zufälligen Payoff eines zu bewertenden Derivats bezeichne.

Mit  $\hat{Y}_0$  werden wir nun zukünftig einen Monte Carlo Schätzer für  $\mathbb{E}[\tilde{P}_0]$ , für den  $N_0$  Pfadsimulationen genutzt werden, mit  $\hat{Y}_l$  je einen Monte Carlo Schätzer für  $\mathbb{E}[\tilde{P}_l - \tilde{P}_{l-1}]$ , für den  $N_l$  Pfadsimulationen verwendet werden, bezeichnen. Dabei werden die Realisierungen der Ausdrücke  $\tilde{P}_l - \tilde{P}_{l-1}$ , welche im Folgenden auch Korrekturterme genannt werden, für  $l > 0$  mithilfe der gerade beschriebenen Vorgehensweise bestimmt.

Auf diese Weise ergibt sich folglich

$$\hat{Y}_l = \frac{1}{N_l} \sum_{i=1}^{N_l} \tilde{P}_l^{(i)} \text{ für } l = 0$$

bzw.

$$\hat{Y}_l = \frac{1}{N_l} \sum_{i=1}^{N_l} (\tilde{P}_l^{(i)} - \tilde{P}_{l-1}^{(i)}) \text{ für } l > 0.$$

Den Gesamtschätzer für  $\mathbb{E}[\tilde{P}_L]$  wollen wir im Folgenden mit  $\hat{Y}$  bezeichnen. Wir erhalten damit den Multilevel Monte Carlo Schätzer

$$\hat{Y} = \hat{Y}_0 + \sum_{l=1}^L \hat{Y}_l.$$

### 3.2 Fehlerschranken und Rechenaufwand

In diesem Abschnitt wollen wir zeigen, dass ein auf obige Weise konstruierter Multilevel Monte Carlo Schätzer bei gleicher Genauigkeit wie der Standard Monte Carlo Schätzer tatsächlich einen geringeren Rechenaufwand erfordert. Dies wollen wir wieder am Beispiel einer Lipschitz-stetigen Payoff-Funktion  $f$  und unter Verwendung des Euler-Maruyama Diskretisierungsverfahrens exemplarisch durchführen. Dazu müssen wir u.a. zeigen, dass die geforderte Fehlerschranke – hier ein MSE der Ordnung  $\mathcal{O}(\epsilon^2)$  – eingehalten wird. Wie wir am Ende von Kapitel 2 gesehen haben, ist dies gleichbedeutend damit, dass die Varianz der Schätzung von der Ordnung  $\mathcal{O}(\epsilon^2)$  und deren Bias von der Ordnung  $\mathcal{O}(\epsilon)$  ist. Um dies zu erreichen, werden wir aufzeigen, wie die optimale Anzahl an Simulationen  $N_l^*$  für jeden Simulationslevel  $l$  und die Gesamtzahl an Simulationsleveln  $L$  zu wählen sind.

Um die Varianz der Schätzer und den Bias des Verfahrens abschätzen zu können, geben wir zunächst die folgenden Lemmata in allgemeiner Form und in Anlehnung an [KP92] an.

**Lemma 3.1** (Schwache Konvergenz des Euler-Maruyama Verfahrens)

Im Falle, dass die Koeffizienten-Funktionen  $a(x, t)$  und  $b(x, t)$  aus der SDE (2.1) nicht von der Zeitvariablen  $t$  abhängig sind, sei  $\tilde{X}^l$  die Euler-Maruyama Approximation für  $X$  mit Schrittweite  $h_l$ . Weiter seien  $a$  und  $b$  Lipschitz-stetig und ausreichend glatt, d.h. für alle Komponenten  $a^k$  von  $a$  und alle Komponenten  $b^{k,j}$  von  $b$  gelte  $a^k, b^{k,j} \in C_P^4(\mathbb{R}^d, \mathbb{R})$  für alle  $1 \leq k \leq d$  und  $1 \leq j \leq m$ , wo bei  $C_P^4(\mathbb{R}^d, \mathbb{R})$  die Menge aller viermal differenzierbaren Funktionen  $\varphi : \mathbb{R}^d \rightarrow \mathbb{R}$  bezeichnet, für die  $\varphi$  und alle seine Ableitungen bis zur vierten Ordnung eine polynomische Wachstumsbedingung erfüllen. Dies bedeutet, dass für alle  $x \in \mathbb{R}^d$  und alle partiellen Ableitungen  $\partial_x^j \varphi$  der Ordnung  $j \leq 4$  Konstanten  $K > 0$  und  $r \in \mathbb{N}^{>0}$  existieren, sodass

$$\|\partial_x^j \varphi(x)\| \leq K(1 + \|x\|^{2r})$$

gilt. Weiter gelten die linearen Wachstumsbeschränkungen  $\|a(x, t)\| \leq K(1 + \|x\|)$  und  $\|b(x, t)\| \leq K(1 + \|x\|)$  für alle  $x \in \mathbb{R}^d$  und eine Konstante  $K > 0$  sowie für alle  $t \in [0, T]$ .

Dann existiert für alle  $f \in C_P^4(\mathbb{R}^d, \mathbb{R})$  eine Konstante  $c_f$ , welche nicht von  $h_l$  abhängt, sodass

$$|\mathbb{E}[f(X(T)) - f(\tilde{X}_T^l)]| \leq c_f h_l, \quad (3.2)$$

d.h.  $\tilde{X}_T^l$  konvergiert schwach gegen  $X(T)$  mit Konvergenzrate 1.

*Beweis:* Für einen Beweis des Lemmas siehe [KP92], S. 473 ff.  $\square$

**Bemerkung 3.2**

Für den Fall, dass die Koeffizienten-Funktionen neben der Abhängigkeit von der Zustandsvariablen  $x$  auch zeitabhängig sind, ist das obige Lemma nach [KP92] ebenfalls anwendbar, falls wir  $t$  als weitere Komponente im Itô-Differential behandeln. In diesem Fall werden durch das Lemma allerdings stärkere Bedingungen an die Ableitungen der Koeffizienten-Funktionen  $a$  und  $b$  bzgl. der Zeitvariablen  $t$  gestellt als notwendig ist.

**Lemma 3.3** (Starke Konvergenz des Euler-Maruyama Verfahrens)

Es gelte

$$\begin{aligned} \mathbb{E}[\|X(0)\|^2] &< \infty, \\ \mathbb{E}[\|X(0) - \tilde{X}_0^l\|^2]^{1/2} &\leq K_1 h_l^{1/2}, \\ \|a(x, t) - a(y, t)\| + \|b(x, t) - b(y, t)\| &\leq K_2 \|x - y\|, \\ \|a(x, t)\| + \|b(x, t)\| &\leq K_3(1 + \|x\|) \text{ und} \\ \|a(x, s) - a(x, t)\| + \|b(x, s) - b(x, t)\| &\leq K_4(1 + \|x\|) \|s - t\|^{1/2} \end{aligned}$$

für alle  $s, t \in [0, T]$  und  $x, y \in \mathbb{R}^d$ , wobei die Konstanten  $K_1$  bis  $K_4$  nicht von  $h_l$  abhängen.

Dann gilt für die Euler-Maruyama Approximation  $\tilde{X}^l$  die Abschätzung

$$\mathbb{E}[\|X(T) - \tilde{X}_T^l\|] \leq K_5 h_l^{1/2}, \quad (3.3)$$

wobei auch die Konstante  $K_5$  nicht von  $h_l$  abhängt.

Dies bedeutet, dass  $\tilde{X}_T^l$  stark gegen  $X(T)$  mit Konvergenzrate 1/2 konvergiert.

*Beweis:* Für einen Beweis des Lemmas siehe [KP92], S. 342 ff.  $\square$

Aus dem Beweis dieses Satzes in [KP92] und Anmerkungen in [Gla03] oder [Gil08a] geht hervor, dass die Euler-Maruyama Approximation sogar einer leicht stärkeren Bedingung, nämlich der Bedingung

$$\mathbb{E}[\|X(T) - \tilde{X}_T^l\|^2] \leq K_5^2 h_l^2, \quad (3.4)$$

genügt, welche wir im Folgenden nutzen werden.

Wir haben nun also gesehen, welche schwache bzw. starke Konvergenzrate das Euler-Maruyama Verfahren im Allgemeinen besitzt, wenn gewisse Voraussetzungen erfüllt sind, was wir an dieser Stelle zunächst einmal annehmen wollen. Damit können wir nun die Varianz des Multilevel Monte Carlo Schätzers  $\hat{Y}$  abschätzen. Als Erstes betrachten wir deshalb die Varianz der Schätzer  $\hat{Y}_l$  für  $l > 0$ , welche sich zu

$$\mathbb{V}[\hat{Y}_l] = \frac{1}{N_l^2} \sum_{i=1}^{N_l} \underbrace{\mathbb{V}[\tilde{P}_l^{(i)} - \tilde{P}_{l-1}^{(i)}]}_{=: V_l} = \frac{1}{N_l} V_l$$

ergibt, da die  $\tilde{P}_l^{(i)} - \tilde{P}_{l-1}^{(i)}$  für alle  $1 \leq i \leq N_l$  identisch und unabhängig verteilt sind.

Für die Varianz des Gesamtschätzers  $\hat{Y} = \sum_{l=0}^L \hat{Y}_l$  folgt damit wiederum mithilfe der Unabhängigkeit der  $\hat{Y}_l$  für verschiedene Level  $l$

$$\mathbb{V}[\hat{Y}] = \mathbb{V}\left[\sum_{l=0}^L \hat{Y}_l\right] = \sum_{l=0}^L \frac{1}{N_l} V_l.$$

Mithilfe der Lipschitz-Stetigkeit der Payoff-Funktion  $f$  sowie der starken Konvergenzrate des Euler-Maruyama Verfahrens bzw. der Gleichung (3.4) können wir nun weiter zeigen, dass die Varianz  $V_l$  von der Ordnung  $\mathcal{O}(h_l)$  ist, denn für entsprechende positive Konstanten  $c_1$  und  $c_2$  gilt

$$\mathbb{V}[\tilde{P}_l - P] \leq \mathbb{E}[(\tilde{P}_l - P)^2] \leq \mathbb{E}[c_1^2 \|\tilde{X}_T^l - X(T)\|^2] \leq c_2 h_l. \quad (3.5)$$

Davon ausgehend erhalten wir zusammen mit der Ungleichung

$$|\text{Cov}[X, Y]| \leq \mathbb{V}[X]^{1/2} \mathbb{V}[Y]^{1/2} \quad (\text{vgl. Satz 17.8 in [Als07]})$$

und der binomischen Formel für eine weitere positive Konstante  $c_3$

$$\begin{aligned} V_l &= \mathbb{V}[\tilde{P}_l - \tilde{P}_{l-1}] \\ &= \mathbb{V}[\tilde{P}_l - P + P - \tilde{P}_{l-1}] \\ &\leq \mathbb{V}[\tilde{P}_l - P] + \mathbb{V}[P - \tilde{P}_{l-1}] + 2|\text{Cov}[\tilde{P}_l - P, P - \tilde{P}_{l-1}]| \\ &\leq \mathbb{V}[\tilde{P}_l - P] + \mathbb{V}[P - \tilde{P}_{l-1}] + 2(\mathbb{V}[\tilde{P}_l - P])^{1/2}(\mathbb{V}[P - \tilde{P}_{l-1}])^{1/2} \\ &= ((\mathbb{V}[\tilde{P}_l - P])^{1/2} + (\mathbb{V}[P - \tilde{P}_{l-1}])^{1/2})^2 \\ &\stackrel{(3.5)}{\leq} ((c_2 h_l)^{1/2} + (c_2 \underbrace{h_{l-1}}_{=Mh_l})^{1/2})^2 \\ &= c_3 h_l, \end{aligned}$$

also

$$V_l = \mathcal{O}(h_l). \quad (3.6)$$

Im nächsten Schritt wollen wir eine optimale Anzahl  $N_l^*$  an durchzuführenden Simulationen für die jeweiligen Level bestimmen. Optimal bedeutet hier, dass die Varianz des Schätzers  $\hat{Y}$  minimiert werden soll, wobei wir von einem festen Rechenaufwand  $C$  ausgehen. Dies bedeutet, dass  $\mathbb{V}[\hat{Y}]$  unter der Nebenbedingung  $\sum_{i=0}^L N_i^* h_i^{-1} = C$  zu minimieren ist. Dazu nutzen wir einen Lagrange-Ansatz und setzen

$$H(N^*) := g(N^*) + \lambda(\langle N^*, h^{-1} \rangle - C),$$

wobei  $N^* = (N_0^*, \dots, N_L^*)^T$ ,  $h = (h_0, \dots, h_L)^T$  und  $g$  durch  $g(N^*) = \sum_{i=0}^L \frac{1}{N_i^*} V_i$  definiert seien und  $\langle \cdot, \cdot \rangle$  hier das Skalarprodukt bezeichne. Indem wir dann die partiellen ersten Ableitungen gleich null setzen, erhalten wir für alle  $l \in \{0, 1, \dots, L\}$

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

und demzufolge

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

sowie

$$0 \stackrel{!}{=} \frac{\partial H(N^*)}{\partial \lambda} = \sum_{i=0}^L N_i^* h_i^{-1} - C = \sum_{i=0}^L \frac{1}{\sqrt{\lambda}} \sqrt{V_i h_i} h_i^{-1} - C$$

bzw.

$$\frac{1}{\sqrt{\lambda}} = C \left( \sum_{i=0}^L \sqrt{V_i / h_i} \right)^{-1}.$$

Wir sehen, dass  $\frac{1}{\sqrt{\lambda}}$  somit eine Normierungskonstante darstellt und können daher folgern, dass

$$N_l^* = c_4 \sqrt{V_l h_l} \stackrel{(3.6)}{\leq} c_5 \sqrt{h_l^2} = c_5 h_l \quad (3.7)$$

für entsprechend gewählte Konstanten  $c_4 > 0$  bzw.  $c_5 > 0$  gilt.

Demnach müssen wir  $N_l^*$  asymptotisch proportional zu  $\sqrt{V_l h_l}$  bzw.  $h_l$  wählen.

Wenn wir dann

$$N_l^* = c_6 \epsilon^{-2} L h_l \quad (3.8)$$

für eine weitere Konstante  $c_6 > 0$  setzen, erhalten wir die geforderte  $\mathcal{O}(\epsilon^2)$ -Varianz

$$\mathbb{V}[\hat{Y}] = \sum_{l=0}^L \frac{V_l}{N_l^*} \stackrel{(3.6)}{\leq} \sum_{l=0}^L \frac{c_3 h_l}{c_6 \epsilon^{-2} L h_l} = (L+1) \frac{c_3}{c_6} \frac{1}{L} \epsilon^2 = c_7 \epsilon^2$$

für eine Konstante  $c_7 > 0$ . Wie gewünscht erhalten wir demnach auf diese Weise

$$\mathbb{V}[\hat{Y}] = \mathcal{O}(\epsilon^2).$$

Nachdem wir gezeigt haben, dass die geforderte Fehlerschranke für die Varianz eines Multilevel Monte Carlo Schätzers eingehalten werden kann, betrachten wir nachfolgend den Bias, also die erwartete Abweichung der Approximation  $\tilde{P}_L$  von  $P$ . Unter Benutzung der schwachen Konvergenz (3.2) ergibt sich dieser dann mit passenden Konstanten  $c_8, c_9 > 0$  sowie der Wahl  $L := \left\lceil \frac{\ln(\epsilon^{-1})}{\ln(M)} \right\rceil = \frac{-\ln(\epsilon)}{\ln(M)} + c_8$  zu

$$\begin{aligned} |\mathbb{E}[\tilde{P}_L - P]| &\leq c_f h_L = c_f (M^{-L} T) = c_f (M^{\ln(\epsilon)/\ln(M)} M^{-c_8} T) \\ &= c_f (M^{(\log_M(\epsilon)/\log_M(e))/(log_M(M)/\log_M(e))} M^{-c_8} T) \\ &= c_f (M^{\log_M(\epsilon)/\log_M(M)} M^{-c_8} T) = c_f \epsilon M^{-c_8} T = c_9 \epsilon \\ &= \mathcal{O}(\epsilon), \end{aligned}$$

wobei im vierten Umformungsschritt ein Basiswechsel des Logarithmus stattfindet.

Dementsprechend wird auch hier die geforderte Genauigkeit eingehalten.

Mit der Definition (3.8) der optimalen Anzahl  $N_l^*$  an für das Level  $l$  durchzuführenden Simulationen sowie der Definition der Anzahl an Simulationsleveln  $L$  betrachten wir nun den Rechenaufwand eines so konstruierten Schätzers, welcher sich für eine entsprechend gewählte Konstante  $c_{10} > 0$  zu

$$\begin{aligned} C &= \sum_{l=0}^L N_l^* h_l^{-1} = \sum_{l=0}^L c_6 \epsilon^{-2} L h_l h_l^{-1} = (L+1) c_6 \epsilon^{-2} L \\ &= c_6 \epsilon^{-2} \left( \left( \frac{-\ln(\epsilon)}{\ln(M)} + c_8 \right) + 1 \right) \left( \frac{-\ln(\epsilon)}{\ln(M)} + c_8 \right) \\ &\leq c_{10} \epsilon^{-2} (\ln(\epsilon)^2 - \ln(\epsilon)) \end{aligned}$$

ergibt, wobei  $\ln(\epsilon)$  im Vergleich zu  $\ln(\epsilon)^2$  für  $\epsilon \rightarrow 0$  vernachlässigbar klein wird. Damit erhalten wir unter obigen Voraussetzungen, wie behauptet, einen Rechenaufwand von der Ordnung

$$\mathcal{O}(\epsilon^{-2} \ln(\epsilon)^2).$$

### 3.3 Zentraler Satz

Nachdem wir die Vorgehensweise bei der Anwendung der Multilevel Monte Carlo Methode kennengelernt und exemplarisch gesehen haben, inwieweit wir den Rechenaufwand bei Verwendung dieses Verfahrens verringern können, kommen wir nun zum zentralen Satz über das Multilevel Monte Carlo Verfahren.

Dieser trifft auch für einen deutlich allgemeineren Fall, bei dem wir uns weder auf ein konkretes Diskretisierungsverfahren beschränken noch die Lipschitz-Stetigkeit der Payoff-Funktion  $f$  voraussetzen, eine Aussage über den (im Vergleich zur Standard Monte Carlo Methode verringerten) Rechenaufwand.

Bei der Formulierung dieses Satzes sowie bei dessen Beweis orientieren wir uns weiterhin an [Gil08a].

#### Satz 3.4

Sei  $f$  ein Funktional abhängig von der Lösung  $X$  der obigen SDE (2.1) für einen gegebenen Pfad eines Wiener-Prozesses  $(W(t))_t$ . Weiter sei  $\tilde{P}_l$  für alle  $l = 0, 1, \dots, L$  wie oben eine Approximation von  $P$ , die mithilfe eines numerischen Diskretisierungsverfahrens mit Schrittweite  $h_l = M^{-l}T$  erstellt wird.

Existieren dann unabhängige Schätzer  $\hat{Y}_l$ , welche auf der Simulation von  $N_l$  Monte Carlo Samples basieren, sowie positive Konstanten  $\alpha \geq \frac{1}{2}, \beta, c_1, c_2, c_3$ , sodass die Bedingungen

$$(i) \quad |\mathbb{E}[\tilde{P}_l - P]| \leq c_1 h_l^\alpha \quad (\text{schwache Konvergenzrate } \alpha \geq \frac{1}{2})$$

$$(ii) \quad \mathbb{E}[\hat{Y}_l] = \begin{cases} \mathbb{E}[\tilde{P}_0] & , l = 0 \\ \mathbb{E}[\tilde{P}_l - \tilde{P}_{l-1}] & , l > 0 \end{cases}$$

$$(iii) \quad \mathbb{V}[\hat{Y}_l] \leq c_2 N_l^{-1} h_l^\beta$$

(iv) der Rechenaufwand  $C_l$  zur Bestimmung von  $\hat{Y}_l$  ist beschränkt durch

$$C_l \leq c_3 N_l h_l^{-1}$$

erfüllt sind, so existiert eine positive Konstante  $c_4$ , sodass für alle  $0 < \epsilon < e^{-1}$  sowohl ein  $L$  als auch  $N_l$  existieren, sodass der Multilevel Schätzer  $\hat{Y} = \sum_{l=0}^L \hat{Y}_l$  einen MSE  $\mathbb{E}[(\hat{Y} - \mathbb{E}[P])^2] < \epsilon^2$  besitzt.

Dabei ist der Rechenaufwand  $C = \sum_{l=0}^L C_l$  beschränkt durch

$$C \leq \begin{cases} c_4 \epsilon^{-2} & , \beta > 1 \\ c_4 \epsilon^{-2} \ln(\epsilon)^2 & , \beta = 1 \\ c_4 \epsilon^{-2-(1-\beta)/\alpha} & , \beta \in (0, 1). \end{cases}$$

*Beweis:* Für den Beweis setzen wir zunächst die Anzahl an Leveln  $L$ , für welche Simulationen durchgeführt werden sollen, als

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

sodass die Ungleichung

$$\frac{1}{\sqrt{2}} M^{-\alpha} \epsilon < c_1 h_L^\alpha \leq \frac{1}{\sqrt{2}} \epsilon \quad (3.9)$$

erfüllt ist. Dies gilt, da einerseits

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

und andererseits

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

ist. Da mit der Bedingung (ii) weiterhin  $\mathbb{E}[\hat{Y}] = \mathbb{E}[\tilde{P}_L]$  gilt, folgt mit der Bedingung (i) und der Ungleichung (3.9) dann für den quadratischen Bias

$$(\mathbb{E}[\hat{Y} - P])^2 \leq (c_1 h_L^\alpha)^2 \leq \frac{1}{2} \epsilon^2. \quad (3.10)$$

Im nächsten Schritt beweisen wir die Ungleichung

$$\sum_{l=0}^L h_l^{-1} < \frac{M^2}{M-1} (\sqrt{2}c_1)^{1/\alpha} \epsilon^{-2}. \quad (3.11)$$

Mit der Gleichung  $h_l = h_L M^{L-l}$  für beliebiges  $l \in \{0, 1, \dots, L\}$ , der Summenformel für die geometrische Reihe und der Äquivalenz des ersten Teils von Ungleichung (3.9) zur Ungleichung  $h_L^{-1} < c_1^{1/\alpha} (\frac{1}{\sqrt{2}} M^{-\alpha} \epsilon)^{-1/\alpha}$  erhalten wir nämlich

$$\begin{aligned} \sum_{l=0}^L h_l^{-1} &= h_L^{-1} \sum_{l=0}^L M^{l-L} = h_L^{-1} M^{-L} \sum_{l=0}^L M^l \\ &= h_L^{-1} M^{-L} \frac{M^{L+1} - 1}{M - 1} < h_L^{-1} \frac{M}{M - 1} \\ &\stackrel{(3.9)}{<} c_1^{1/\alpha} \left( \frac{1}{\sqrt{2}} M^{-\alpha} \epsilon \right)^{-1/\alpha} \frac{M}{M - 1} \\ &= \frac{M^2}{M - 1} (c_1 \sqrt{2} \epsilon^{-1})^{1/\alpha} \\ &\leq \frac{M^2}{M - 1} (\sqrt{2} c_1)^{1/\alpha} \epsilon^{-2}, \end{aligned}$$

da wir für  $\alpha \geq \frac{1}{2}$  und  $0 < \epsilon < e^{-1} (< 1)$  die Ungleichung  $\epsilon^{-1/\alpha} \leq \epsilon^{-2}$  ausnutzen können.

Im Folgenden werden wir den Beweis mithilfe einer Fallunterscheidung bzgl.  $\beta$  abschließen:

1.  $\beta > 1$ :

Setze

$$N_l = \lceil 2\epsilon^{-2} c_2 T^{(\beta-1)/2} (1 - M^{-(\beta-1)/2})^{-1} h_l^{(\beta+1)/2} \rceil. \quad (3.12)$$

Unter Verwendung dieser Definition und der Ungleichung

$$\sum_{l=0}^L h_l^{(\beta-1)/2} < T^{(\beta-1)/2} (1 - M^{-(\beta-1)/2})^{-1}, \quad (3.13)$$

welche wir mithilfe der Gleichung  $h_l = M^{-l} T$  und der Summenformel für die geometrische Reihe durch die Abschätzung

$$\begin{aligned} \sum_{l=0}^L h_l^{(\beta-1)/2} &= T^{(\beta-1)/2} \sum_{l=0}^L (M^{-(\beta-1)/2})^l \\ &= T^{(\beta-1)/2} \frac{1 - (M^{-(\beta-1)/2})^{L+1}}{1 - M^{-(\beta-1)/2}} \\ &< T^{(\beta-1)/2} (1 - M^{-(\beta-1)/2})^{-1} \end{aligned}$$

erhalten, ergibt sich für die Varianz des Schätzers  $\hat{Y}$  dann

$$\begin{aligned}\mathbb{V}[\hat{Y}] &= \sum_{l=0}^L \mathbb{V}[\hat{Y}_l] \stackrel{(iii)}{\leq} \sum_{l=0}^L c_2 N_l^{-1} h_l^\beta \\ &\stackrel{(3.12)}{\leq} \sum_{l=0}^L c_2 \frac{1}{2\epsilon^{-2}c_2} T^{-(\beta-1)/2} (1 - M^{-(\beta-1)/2}) h_l^{-(\beta+1)/2} h_l^\beta \\ &= \frac{1}{2} \epsilon^2 T^{-(\beta-1)/2} (1 - M^{-(\beta-1)/2}) \sum_{l=0}^L h_l^{(\beta-1)/2} \\ &\stackrel{(3.13)}{<} \frac{1}{2} \epsilon^2.\end{aligned}$$

Zusammen mit der Abschätzung (3.10) erhalten wir für den MSE des Multilevel Schätzers nun wie gefordert

$$\text{MSE}(\hat{Y}) = (\mathbb{E}[\hat{Y} - P])^2 + \mathbb{V}[\hat{Y}] < \frac{1}{2} \epsilon^2 + \frac{1}{2} \epsilon^2 = \epsilon^2.$$

Mithilfe der Abschätzung

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

welche nach Definition (3.12) von  $N_l$  gilt, ergibt sich für den Rechenaufwand  $C$  in diesem Fall wie behauptet

$$\begin{aligned}C &= \sum_{l=0}^L C_l \stackrel{(iv)}{\leq} \sum_{l=0}^L c_3 N_l h_l^{-1} \\ &\leq c_3 \sum_{l=0}^L (2\epsilon^{-2} c_2 T^{(\beta-1)/2} (1 - M^{-(\beta-1)/2})^{-1} h_l^{(\beta+1)/2} + 1) h_l^{-1} \\ &= c_3 (2\epsilon^{-2} c_2 T^{(\beta-1)/2} (1 - M^{-(\beta-1)/2})^{-1} \sum_{l=0}^L h_l^{(\beta-1)/2} + \sum_{l=0}^L h_l^{-1}) \\ &\stackrel{(3.11),(3.13)}{<} \underbrace{(2c_3 c_2 T^{(\beta-1)} (1 - M^{-(\beta-1)/2})^{-2} + c_3 \frac{M^2}{M-1} (\sqrt{2}c_1)^{1/\alpha}) \epsilon^{-2}}_{=:c_4}.\end{aligned}$$

2.  $\beta = 1$ :

Setze

$$N_l = \lceil 2\epsilon^{-2} c_2 (L+1) h_l \rceil. \quad (3.14)$$

Dann gilt für die Varianz von  $\hat{Y}$  auch hier

$$\mathbb{V}[\hat{Y}] = \sum_{l=0}^L \mathbb{V}[\hat{Y}_l] \stackrel{(iii)}{\leq} \sum_{l=0}^L c_2 N_l^{-1} h_l \stackrel{(3.14)}{\leq} \sum_{l=0}^L c_2 (c_2 2\epsilon^{-2} (L+1) h_l)^{-1} h_l = \frac{1}{2} \epsilon^2.$$

Nutzen wir wieder die Abschätzung (3.10) für den Bias der Schätzung, so erhalten wir auch in diesem Fall einen MSE, der beschränkt ist durch

$$\text{MSE}(\hat{Y}) < \epsilon^2.$$

Um den Rechenaufwand  $C$  abschätzen zu können, zeigen wir zunächst, dass

$$L + 1 \leq c_5 \ln(\epsilon^{-1}) \quad (3.15)$$

für eine Konstante  $c_5 > 0$  gilt. Dies folgt, da nach Definition von  $L$

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

ist und wir mit

$$(\epsilon < e^{-1}) \Leftrightarrow (\epsilon^{-1} > e) \Leftrightarrow (\ln(\epsilon^{-1}) > 1) \quad (3.16)$$

dann

$$\begin{aligned} L + 1 &\leq \frac{\ln(\epsilon^{-1})}{\alpha \ln(M)} + \frac{\ln(\sqrt{2}c_1 T^\alpha)}{\alpha \ln(M)} + 2 \\ &\stackrel{(3.16)}{<} \underbrace{\ln(\epsilon^{-1}) \left( \frac{1}{\alpha \ln(M)} + \max\{0, \frac{\ln(\sqrt{2}c_1 T^\alpha)}{\alpha \ln(M)}\} + 2 \right)}_{=:c_5} \end{aligned}$$

erhalten. Für  $N_l$  gilt nach Definition (3.14)  $N_l \leq 2\epsilon^{-2}(L+1)c_2h_l + 1$  und damit folgt für den Rechenaufwand  $C$  nun

$$\begin{aligned} C &\stackrel{(iv)}{\leq} c_3 \sum_{l=0}^L N_l h_l^{-1} \leq c_3 \left( \sum_{l=0}^L 2\epsilon^{-2}(L+1)c_2 + h_l^{-1} \right) \\ &\leq c_3 (2\epsilon^{-2}(L+1)^2 c_2 + \sum_{l=0}^L h_l^{-1}) \\ &\stackrel{(3.15),(3.11)}{\leq} c_3 2\epsilon^{-2} \underbrace{\ln(\epsilon^{-1})^2 c_5^2 c_2}_{=\ln(\epsilon)^2} + c_3 \frac{M^2}{M-1} (\sqrt{2}c_1)^{1/\alpha} \epsilon^{-2} \\ &\stackrel{(3.16)}{<} \underbrace{(c_3 2c_5^2 c_2 + c_3 \frac{M^2}{M-1} (\sqrt{2}c_1)^{1/\alpha}) \epsilon^{-2} \ln(\epsilon)^2}_{=:c_4}. \end{aligned}$$

3.  $\beta < 1$ :

Setze

$$N_l = \lceil 2\epsilon^{-2} c_2 h_L^{-(1-\beta)/2} (1 - M^{-(1-\beta)/2})^{-1} h_l^{(\beta+1)/2} \rceil. \quad (3.17)$$

Zusammen mit der Ungleichung

$$\begin{aligned}
 \sum_{l=0}^L h_l^{-(1-\beta)/2} &= \sum_{l=0}^L (M^{L-l} h_L)^{-(1-\beta)/2} \\
 &= (M^L h_L)^{-(1-\beta)/2} \sum_{l=0}^L (M^{(1-\beta)/2})^l \\
 &= h_L^{-(1-\beta)/2} M^{-L(1-\beta)/2} \frac{(M^{(1-\beta)/2})^{L+1} - 1}{M^{(1-\beta)/2} - 1} \\
 &= h_L^{-(1-\beta)/2} M^{-(L+1)(1-\beta)/2} \frac{M^{(L+1)(1-\beta)/2} - 1}{1 - M^{-(1-\beta)/2}} \\
 &< h_L^{-(1-\beta)/2} (1 - M^{-(1-\beta)/2})^{-1}
 \end{aligned} \tag{3.18}$$

gilt dann auch hier

$$\begin{aligned}
 \mathbb{V}[\hat{Y}] &= \sum_{l=0}^L \mathbb{V}[\hat{Y}_l] \stackrel{(iii)}{\leq} \sum_{l=0}^L c_2 N_l^{-1} h_l \\
 &\stackrel{(3.17)}{\leq} \sum_{l=0}^L c_2 c_2^{-1} \frac{1}{2} \epsilon^2 h_L^{(1-\beta)/2} (1 - M^{-(1-\beta)/2}) h_l^{-(1-\beta)/2} \\
 &\stackrel{(3.18)}{<} \frac{1}{2} \epsilon^2.
 \end{aligned}$$

Da für  $N_l$  nach Definition (3.17)

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

gilt und für  $\epsilon < e^{-1} (< 1)$

$$\epsilon^{-2} < \epsilon^{-2-(1-\beta)/\alpha} \tag{3.19}$$

ist, erhalten wir in diesem Fall für den Rechenaufwand abschließend

$$\begin{aligned}
 C &\stackrel{(iv)}{\leq} \sum_{l=0}^L c_3 N_l h_l^{-1} \\
 &< c_3 \left( \sum_{l=0}^L 2\epsilon^{-2} c_2 h_L^{-(1-\beta)/2} (1 - M^{-(1-\beta)/2})^{-1} h_l^{-(1-\beta)/2} + h_l^{-1} \right) \\
 &\stackrel{(3.18),(3.11)}{<} 2c_3 c_2 \epsilon^{-2} h_L^{-(1-\beta)} (1 - M^{-(1-\beta)/2})^{-2} + c_3 \frac{M^2}{M-1} (\sqrt{2}c_1)^{1/\alpha} \epsilon^{-2} \\
 &\stackrel{(3.9)}{<} 2c_3 c_2 \epsilon^{-2} (\sqrt{2}c_1)^{\frac{1-\beta}{\alpha}} M^{(1-\beta)} \epsilon^{-(1-\beta)/\alpha} (1 - M^{-(1-\beta)/2})^{-2} \\
 &\quad + c_3 \frac{M^2}{M-1} (\sqrt{2}c_1)^{1/\alpha} \epsilon^{-2} \\
 &\stackrel{(3.19)}{<} \underbrace{(2c_3 c_2 (\sqrt{2}c_1)^{\frac{1-\beta}{\alpha}} M^{(1-\beta)} (1 - M^{-(1-\beta)/2})^{-2} + c_3 \frac{M^2}{M-1} (\sqrt{2}c_1)^{1/\alpha})}_{=:c_4} \\
 &\quad \epsilon^{-2-(1-\beta)/\alpha}.
 \end{aligned}$$

□

### 3.4 Anwendung in der Praxis

#### 3.4.1 Biasschätzung

Der im vorherigen Abschnitt bewiesene Satz 3.4 sichert uns theoretisch zu, dass das Multilevel Monte Carlo Verfahren einen im Vergleich zum klassischen Monte Carlo Verfahren verringerten Rechenaufwand besitzt. Trotzdem können wir dies in der Praxis nicht auf diese Weise garantieren. Im Satz wird nämlich u.a. die Kenntnis der Konstante  $c_1$ , welche die schwache Konvergenz des Diskretisierungsverfahrens beeinflusst, a priori vorausgesetzt. Diese Konstante wird in den meisten Fällen allerdings erst implizit während der Durchführung der Simulationsalgorithmen bestimmt (vgl. [Gil08a]).

Wie wir im Beweis dieses zentralen Satzes sehen, ist die Kenntnis der Konstanten  $c_1$  ebenfalls notwendig, um zu Beginn der Ausführung der Algorithmen die notwendige Anzahl an durchzuführenden Simulationslevels  $L$  festlegen zu können. Da dies nun in der Praxis nicht möglich ist, müssen wir nach jedem Level, für welchen wir Pfade simuliert haben, einen Konvergenztest durchführen, welcher überprüft, ob der Bias des Schätzers bereits ausreichend klein ist.

Nachfolgend wollen wir – weiterhin in Anlehnung an [Gil08a] – zeigen, auf welche Weise wir den Bias der Schätzung sinnvoll abschätzen können und damit die in einem Konvergenztest zu verwendende Abbruchbedingung herleiten.

Zu diesem Zweck werden wir sehen, dass uns die einzelnen Schätzer  $\hat{Y}_l$  für die Korrekturterme  $\mathbb{E}[\tilde{P}_l - \tilde{P}_{l-1}]$  für  $l > 1$  bei der Schätzung des verbleibenden Bias weiterhelfen können. Gehen wir – wie in der gesamten Arbeit – von der Benutzung eines Diskretisierungsverfahrens mit einer schwachen Konvergenzrate von 1 aus, so liefert uns dies für  $l \rightarrow \infty$  und eine (unbekannte) Konstante  $c_1$  einen asymptotischen Fehler

$$\mathbb{E}[P - \tilde{P}_l] \approx c_1 h_l.$$

Mithilfe der obigen Korrekturterme können wir diese Konstante nun eliminieren und erhalten

$$\begin{aligned} \mathbb{E}[\tilde{P}_l - \tilde{P}_{l-1}] &= \mathbb{E}[\tilde{P}_l - P] + \mathbb{E}[P - \tilde{P}_{l-1}] \approx -\mathbb{E}[P - \tilde{P}_l] + c_1 h_{l-1} \\ &= -\mathbb{E}[P - \tilde{P}_l] + c_1 M h_l \approx -\mathbb{E}[P - \tilde{P}_l] + M \mathbb{E}[P - \tilde{P}_l] \\ &= (M-1) \mathbb{E}[P - \tilde{P}_l]. \end{aligned} \tag{3.20}$$

Diese Approximation wollen wir im Folgenden als ungefähre Schranke für den verbleibenden Bias nutzen, so dass wir einen Bias kleiner als  $\frac{1}{\sqrt{2}}\epsilon$  erhalten. Dazu erhöhen wir im Algorithmus den Wert für  $L$  so lange, bis der Schätzer  $\hat{Y}_L$  für den Korrekturterm  $\mathbb{E}[\tilde{P}_L - \tilde{P}_{L-1}]$  die Ungleichung

$$|\hat{Y}_L| < \frac{1}{\sqrt{2}}(M-1)\epsilon \tag{3.21}$$

erfüllt.

Dies gewährleistet, dass der approximierte quadratische Bias  $|\mathbb{E}[P - \tilde{P}_L]|^2$  gerade die erforderliche Fehlerschranke unterschreitet, da mit obiger Ungleichung (3.21)

$$|\mathbb{E}[P - \tilde{P}_L]|^2 \stackrel{(3.20)}{\approx} (M-1)^{-2} |\mathbb{E}[\tilde{P}_L - \tilde{P}_{L-1}]|^2 \approx (M-1)^{-2} |\hat{Y}_L|^2 \stackrel{(3.21)}{<} \frac{1}{2}\epsilon^2$$

gilt. Da mit (3.20) weiterhin

$$\begin{aligned}\hat{Y}_L - \hat{Y}_{L-1} &= \mathbb{E}[\tilde{P}_L - \tilde{P}_{L-1}] - [\tilde{P}_{L-1} - \tilde{P}_{L-2}] \\ &\approx (M-1)\mathbb{E}[P - \tilde{P}_L] - (M-1)\mathbb{E}[P - \tilde{P}_{L-1}] \\ &= (M-1)\mathbb{E}[\tilde{P}_{L-1} - \tilde{P}_L] = (1-M)\hat{Y}_L\end{aligned}$$

gilt, woraus wiederum

$$\hat{Y}_L \approx M^{-1}\hat{Y}_{L-1}$$

folgt, werden wir für die numerische Implementierung die Bedingung (3.21) etwas schärfner formulieren und

$$\max\{M^{-1}|\hat{Y}_{L-1}|, |\hat{Y}_L|\} < \frac{1}{\sqrt{2}}(M-1)\epsilon \quad (3.22)$$

fordern.

Auf diese Weise ist gewährleistet, dass sowohl der aktuell letzte als auch der vorletzte Level die Konvergenzbedingung erfüllen und Probleme durch einen Vorzeichenwechsel beim Korrekturterm  $\mathbb{E}[\tilde{P}_l - \tilde{P}_{l-1}]$  von aufeinanderfolgenden Simulationsleveln vermieden werden (vgl. [Gil08a]).

### 3.4.2 Die optimale Anzahl an Pfadsimulationen

Eine andere Stelle, an der die Unkenntnis der Konstanten im Satz 3.4 in der Praxis Probleme hervorruft, ist die Bestimmung der für jeden Level  $l$  (optimalen) Anzahl an durchzuführenden Pfadsimulationen  $N_l^*$ . Der zentrale Satz 3.4 liefert dazu nur die Existenz gewisser  $N_l$  und lässt dabei zunächst offen, wie genau diese gewählt werden müssen. Im Beweis des Satzes wird dann abhängig vom Parameter  $\beta$  eine optimale Anzahl an Simulationen für jedes  $l$  definiert.

Bei dieser Definition wird allerdings die Kenntnis der Konstante  $c_2$  und für  $\beta \leq 1$  auch die Kenntnis des Parameters  $L$  vorausgesetzt. Bevor wir jedoch die Simulationen mithilfe des nachfolgenden Algorithmus durchführen, kennen wir diese Parameter in der Praxis meist nicht und auch sie werden erst während der Berechnungen mitbestimmt. Dies führt dazu, dass wir die optimale Anzahl an Simulationen nicht a priori festlegen können, wie es im Beweis des Satzes geschieht. Wir werden deshalb zunächst eine festgelegte Anzahl  $N_{\text{Start}}$  (siehe Kapitel 8) an Pfaden simulieren und anschließend, abhängig von der Stichprobenvarianz dieser Samples, die optimale Anzahl  $N_l^*$  an zu simulierenden Pfaden für jeden bereits durchgeföhrten Simulationslevel  $l$  ermitteln.

In Gleichung (3.7) hatten wir gesehen, dass die optimale Anzahl an Simulationen proportional zur Größe  $\sqrt{V_l h_l}$  gewählt werden sollte. Zusammen mit der von der Varianz des Schätzers einzuhaltenen Fehlerschranke  $\mathbb{V}[\hat{Y}] \leq \frac{1}{2}\epsilon^2$  gelangen wir damit zu

$$N_l^* = \lceil 2\epsilon^{-2} \sqrt{V_l h_l} \left( \sum_{i=0}^L \sqrt{V_i/h_i} \right) \rceil, \quad (3.23)$$

denn es gilt dann wie gefordert

$$\mathbb{V}[\hat{Y}] = \mathbb{V}\left[\sum_{l=0}^L \hat{Y}_l\right] = \sum_{l=0}^L \frac{1}{N_l^*} V_l \leq \frac{1}{2} \epsilon^2 \left(\sum_{l=0}^L \sqrt{V_l/h_l}\right) \left(\sum_{i=0}^L \sqrt{V_i/h_i}\right)^{-1} = \frac{1}{2} \epsilon^2.$$

### 3.5 Der Multilevel Monte Carlo Algorithmus

Für eine Implementierung des obigen Multilevel Monte Carlo Verfahrens werden wir somit den folgenden Algorithmus verwenden, welchen wir in Kapitel 7 noch detaillierter betrachten. Wie in [Gil08a] vorgeschlagen, verwenden wir dabei im Folgenden einen Verfeinerungsfaktor  $M = 4$ .

#### Algorithmus 3.5 (Multilevel Monte Carlo Simulation)

1. Setze  $L = 0$
2. Solange  $L < 2$  oder das Konvergenzkriterium (3.22) noch nicht erfüllt ist:
  - (a) Setze  $N_L = N_{Start}$  (initiale Anzahl zu simulierender Pfade)
  - (b) Setze  $h_L = M^{-L}T$  (Diskretisierungsschrittweite)
  - (c) Für alle  $j = 1, 2, \dots, N_L$ :  
Simuliere die  $\tilde{P}_0^{(j)}$  (für  $L = 0$ ) bzw. die  $\tilde{P}_L^{(j)} - \tilde{P}_{L-1}^{(j)}$  (für  $L > 0$ ) (Simulation je nach Modell- und Produktwahl (siehe Kapitel 7))
  - (d) Bestimme  $\hat{Y}_L$  und die Stichprobenvarianz  $V_L$  anhand der simulierten  $\tilde{P}_0^{(j)}$  (für  $L = 0$ ) bzw.  $\tilde{P}_L^{(j)} - \tilde{P}_{L-1}^{(j)}$  (für  $L > 0$ )
  - (e) Für alle  $l = 0, \dots, L$ 
    - i. Bestimme die optimale Anzahl an Pfaden  $N_l^*$  mithilfe der neuen „Information“  $V_L$  als

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

- ii. Falls  $N_l^* > N_l$ 
    - A. Für alle  $j = N_l + 1, \dots, N_l^*$ :  
Simuliere weitere Samples  $\tilde{P}_0^{(j)}$  (für  $l = 0$ ) bzw.  $\tilde{P}_l^{(j)} - \tilde{P}_{l-1}^{(j)}$  (für  $l > 0$ ) und aktualisiere  $\hat{Y}_l$  und  $V_l$
    - B. Setze  $N_l = N_l^*$
  - (f) Setze  $L = L + 1$
3. Setze  $\hat{Y} = \sum_{l=0}^L \hat{Y}_l$  (Multilevel Monte Carlo Schätzwert für den Payoff  $P$ )

## 4 Das Libor Marktmodell

### 4.1 Einleitung

Das Libor Marktmodell ist ein Zinsstrukturmodell, in dem die sogenannten „London Interbank Offered Rates“, also gewisse Interbanken-Zinssätze, beschrieben werden. Die modellierten Libor-Raten sind dabei Forward-Zinsraten, welche errechnete durchschnittliche Kreditzinssätze darstellen, die von Londoner Banken mit hoher Bonität untereinander gehandelt werden. Zusammen mit den Euribor-Zinssätzen, den „Euro Interbank Offered Rates“, stellen die Libor-Zinssätze einen weltweiten Benchmark für Zinsraten mit kurzen Laufzeiten dar.

Da das Modell auf beobachtbaren Marktpreisen basiert, wird es als Marktmodell bezeichnet. Dies stellt einen wesentlichen Unterschied zu gängigen Zinsstrukturmodellen wie dem Zinsstrukturmodell von Heath, Jarrow und Morton (HJM), bei dem die Forward-Zinskurve modelliert wird, oder den Short-Rate-Modellen, bei denen der Momentanzins modelliert wird, dar. Diese Modelle sind zwar mathematisch gut handhabbar, allerdings sind die modellierten Größen nicht am Markt beobachtbar.

Bei der Darstellung der mathematischen Modellierung in diesem Kapitel orientieren wir uns an [Has12].

### 4.2 Mathematisches Modell

Für die mathematische Modellierung der Libor-Raten benötigen wir zunächst eine Tenorstruktur, d.h. eine diskrete Folge aufeinanderfolgender (zukünftiger) Zeitpunkte  $T_0 < T_1 < T_2 < \dots < T_N$  mit Intervalllängen  $\delta_i := T_{i+1} - T_i, 1 \leq i \leq N-1$ . Im weiteren Verlauf dieser Arbeit gehen wir davon aus, dass alle Tenorintervalle dieselbe Länge besitzen und definieren folglich  $\delta_i := \delta$  für alle  $1 \leq i \leq N-1$  und ein  $\delta > 0$ .

Als Basisfinanzgüter im Libor Marktmodell dienen Zerobondpreise  $(B(t, T_i))_{0 \leq t \leq T_i}$ , wobei zu beachten ist, dass ein solcher  $T_i$ -Bond jeweils nur bis zum Zeitpunkt  $T_i$  gehandelt wird. Als Numeraire Asset Preisprozess dieses Modells wählen wir den Preisprozess des  $T_N$ -Bonds  $(B(t, T_N))_{0 \leq t \leq T_N}$ , was insbesondere Sinn macht, weil dieser bis zum Ende des gesamten betrachteten Handelszeitraums  $[0, T_N]$  gehandelt wird und somit der Preisprozess eines jeden  $T_i$ -Bonds ( $1 \leq i \leq N-1$ ) in Einheiten des Numeraire Assets dargestellt werden kann.

Für die Modellierung des Zufalls im Libor Marktmodell können wir zwischen zwei Ansätzen, welche sich jeweils in den anderen überführen lassen, unterscheiden. Zum einen kann ein  $m$ -dimensionaler Standard Wiener-Prozess  $(\bar{W}(t))_t$  als Quelle des Zufalls gewählt werden, welcher alle Libor-Raten treibt. Zum anderen können wir einen  $(N-1)$ -dimensionalen Wiener-Prozess, dessen Komponenten untereinander korreliert sind, nutzen, um die Quelle des Zufalls darzustellen. In dieser Variante wird jede Libor-Rate dann von einer Komponente dieses Wiener-Prozesses getrieben.

Bei der Darstellung dieser beiden Ansätze wollen wir mit der ersten Variante

beginnen und die Dynamiken der Libor-Raten unter Verwendung eines unkorrelierten Standard Wiener-Prozesses herleiten, um diese anschließend in die zweite Darstellungsvariante zu überführen.

Wir wollen annehmen, dass das Modell arbitragefrei ist beziehungsweise, dass ein (zu  $\mathbb{Q}$ ) äquivalentes Martingalmaß  $\mathbb{P}_{T_N}$  existiert, so dass  $(\frac{B(t, T_i)}{B(t, T_N)})_{0 \leq t \leq T_i}$  für alle  $1 \leq i \leq N-1$  ein  $\mathbb{P}_{T_N}$ -Martingal ist. Dieses Forwardmartingalmaß  $\mathbb{P}_{T_N}$  wird auch als „terminal measure“ bezeichnet, da  $T_N$  den letzten Zeitpunkt auf der Tenorstruktur darstellt. Weiterhin sei  $\bar{W}$  nun ein  $m$ -dimensionaler Standard Wiener-Prozess bzgl. dieses äquivalenten Martingalmaßes  $\mathbb{P}_{T_N}$ .

Die  $i$ -te Libor-Rate soll in diesem Kontext die Forward-Zinsrate zum Zeitpunkt  $t$  für eine Verzinsung während des Zeitintervalls  $[T_i, T_{i+1}]$  darstellen. Dazu definieren wir die Libor-Raten-Prozesse  $(L^{(i)}(t))_{0 \leq t \leq T_i}$  durch

$$L^{(i)}(t) = \frac{1}{\delta} \left( \frac{B(t, T_i)}{B(t, T_{i+1})} - 1 \right) \forall 1 \leq i \leq N-1. \quad (4.1)$$

Nehmen wir nun sinnvollerweise an, dass für alle  $t > 0$  und für alle  $1 \leq i \leq N-1$   $B(t, T_i) > B(t, T_{i+1})$  ist, so sind die Libor-Raten-Prozesse strikt positiv.

Mithilfe dieser Voraussetzungen wollen wir nun die Dynamiken der Libor-Raten-Prozesse herleiten. Zuerst bestimmen wir dazu die Dynamik der  $(N-1)$ -ten Libor-Rate und gehen anschließend induktiv von hinten nach vorne vor, um die weiteren Libor-Raten-Dynamiken zu erklären.

Wir beginnen mit der Feststellung, dass  $L^{(N-1)}(t) = \frac{1}{\delta} \left( \frac{B(t, T_{N-1})}{B(t, T_N)} - 1 \right)$  nach Voraussetzung ein strikt positives  $\mathbb{P}_{T_N}$ -Martingal ist. Nach dem Martingaldarstellungssatz existiert dann ein bzgl.  $(\mathcal{F}_t)_t$  previsibler  $\mathbb{R}^m$ -dimensionaler Prozess  $(M^{(N-1)}(s))_{0 \leq s \leq T_{N-1}}$  mit

$$\begin{aligned} L^{(N-1)}(t) &= L^{(N-1)}(0) + \sum_{j=1}^m \int_0^t M^{(N-1,j)}(s) d\bar{W}^{(j)}(s) \\ &= L^{(N-1)}(0) + \sum_{j=1}^m \int_0^t \frac{M^{(N-1,j)}(s)}{L^{(N-1)}(s)} L^{(N-1)}(s) d\bar{W}^{(j)}(s) \\ &= L^{(N-1)}(0) + \sum_{j=1}^m \int_0^t \bar{\sigma}^{(N-1,j)}(s) L^{(N-1)}(s) d\bar{W}^{(j)}(s), \end{aligned}$$

wobei  $\bar{\sigma}^{(N-1,j)}(s) := \frac{M^{(N-1,j)}(s)}{L^{(N-1)}(s)}$  definiert wird und  $\bar{W}^{(j)}$  für alle  $1 \leq j \leq m$  die  $j$ -te Komponente des Wiener-Prozesses  $\bar{W}$  bezeichne.

In Differentialschreibweise lautet dies also

$$dL^{(N-1)}(t) = L^{(N-1)}(t) \sum_{j=1}^m \bar{\sigma}^{(N-1,j)} d\bar{W}^{(j)}(t).$$

Für die nächsten Schritte ist zu beachten, dass die Libor-Raten  $(L^{(i)}(t))_{0 \leq t \leq T_i}$  für  $1 \leq i \leq N-2$  als Quotienten von  $\mathbb{P}_{T_N}$ -Martingalen jeweils  $\mathbb{P}_{T_N}$ -Semimartingale

sind, da sich die Quotienten der aufeinanderfolgenden  $T_i$ -Bonds als

$$\frac{B(t, T_i)}{B(t, T_{i+1})} = \frac{B(t, T_i)/B(t, T_N)}{B(t, T_{i+1})/B(t, T_N)}$$

darstellen lassen.

Wiederum mithilfe des Martingaldarstellungssatzes erhalten wir, was wir hier unter Verweis auf [Has12] ohne explizite Herleitung angeben wollen, für die übrigen Libor-Raten  $(L^{(i)}(t))_{0 \leq t \leq T_i}$  für  $1 \leq i \leq N-2$  bzgl. des „terminal measures“ die Darstellung

$$dL^{(i)}(t) = L^{(i)}(t) \sum_{j=1}^m \bar{\sigma}^{(i,j)}(t) \left( - \sum_{k=i+1}^{N-1} \frac{\delta L^{(k)}(t) \bar{\sigma}^{(k,j)}(t)}{\delta L^{(k)}(t) + 1} \right) dt + d\bar{W}^{(j)}(t). \quad (4.2)$$

Wie bereits angekündigt, wollen wir ausgehend von dieser Darstellung der Libor-Raten bzgl. eines  $m$ -dimensionalen Wiener-Prozesses mit unkorrelierten Komponenten nun eine alternative Libor-Raten-Darstellung bzgl. eines  $(N-1)$ -dimensionalen korrelierten Wiener-Prozesses  $\bar{W}$  herleiten.

Wir betrachten dazu die Korrelationsmatrix

$$\Sigma = \begin{pmatrix} 1 & \varrho_{1,2} & \cdots & \varrho_{1,(N-1)} \\ \varrho_{2,1} & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \varrho_{(N-2),(N-1)} \\ \varrho_{(N-1),1} & \cdots & \varrho_{(N-1),(N-2)} & 1 \end{pmatrix}$$

des korrelierten Wiener-Prozesses  $\bar{W}$ , welche wir in der Praxis über eine Markt-kalibrierung (siehe Kapitel 8.1) erhalten können.

Im Folgenden gehen wir davon aus, dass  $m = N-1$  gilt. Dies ist sinnvoll, da wir zum einen die Vollständigkeit des Modells fordern können, was dazu führt, dass  $m \leq N-1$  gelten muss und wir zum anderen gegebenenfalls „künstlich“ weitere Wiener-Prozesse zum Modell hinzufügen können, welche wir über die Definition der Faktoren  $\bar{\sigma}$  kontrollieren bzw. wieder eliminieren können, um dem Modell dadurch letztlich kein zusätzliches Risiko hinzuzufügen.

Betrachten wir nun die Cholesky-Zerlegung  $AA^T (= \Sigma)$  dieser symmetrischen und positiv definiten  $(N-1) \times (N-1)$ -Matrix  $\Sigma$ , auf die wir später näher eingehen werden, dann können wir den korrelierten Wiener-Prozess  $\bar{W}$  mithilfe des unkorrelierten Wiener-Prozesses  $\bar{W}$  aus der ersten Variante definieren als

$$W(t) = A\bar{W}(t) \text{ bzw. } W^{(i)}(t) = \sum_{j=1}^{N-1} a_{i,j} \bar{W}^{(j)}(t), \quad 1 \leq i \leq N-1, \quad 0 \leq t \leq T_i.$$

Dass  $(W(t))_t$  wieder ein Wiener-Prozess ist, sieht man mithilfe des Satzes von Lévy, welchen wir unter Verweis auf [KS97] angeben wollen.

**Satz 4.1** (Satz von Lévy)

Sei  $W = (W^{(1)}(t), \dots, W^{(m)}(t))$ ,  $0 \leq t < \infty$ , ein stetiger, adaptierter Prozess im  $\mathbb{R}^m$ , so dass der Prozess

$$\Delta W^{(i)}(t) := W^{(i)}(t) - W^{(i)}(0), \quad 0 \leq t < \infty$$

für jede Komponente  $1 \leq i \leq m$  ein stetiges, adaptiertes lokales Martingal ist. Ferner seien die Kovariationsprozesse gegeben durch

$$\langle \Delta W^{(i)}(\cdot), \Delta W^{(k)}(\cdot) \rangle_t = \varrho_{i,k} t, \quad 1 \leq i, k \leq m.$$

Dann ist  $W$  ein  $m$ -dimensionaler Wiener-Prozess.

Die Voraussetzungen des Satzes sind hier erfüllt, da die einzelnen Komponenten von  $W$  als Linearkombinationen von Komponenten von  $\bar{W}$  wieder adaptiert sind und stetige Pfade besitzen. Auch die Martingaleigenschaft ist aufgrund der Martingaleigenschaft der Komponenten von  $\bar{W}$  für jede Komponente von  $W$  erfüllt. Es reicht also zu zeigen, dass die Kovariationsterme die geforderten Eigenschaften erfüllen. Da  $\bar{W}$  ein Wiener-Prozess ist, gilt  $\bar{W}^{(j)}(0) = 0$  für alle  $1 \leq j \leq N-1$  und somit auch  $W^{(j)}(0) = 0$  für alle  $1 \leq j \leq N-1$ . Damit gilt hier  $\Delta W^{(j)}(t) = W^{(j)}(t)$  für alle  $1 \leq j \leq N-1$ . Wegen der Unabhängigkeit der Komponenten von  $\bar{W}$  und der Definition der Cholesky-Matrix  $A$  gilt dann weiter

$$\begin{aligned} \langle W^{(i)}(\cdot), W^{(k)}(\cdot) \rangle_t &= \langle (A\bar{W}(\cdot))_i, (A\bar{W}(\cdot))_k \rangle_t = \left\langle \sum_{j=1}^{N-1} a_{i,j} \bar{W}^{(j)}(\cdot), \sum_{j=1}^{N-1} a_{k,j} \bar{W}^{(j)}(\cdot) \right\rangle_t \\ &= \sum_{j=1}^{N-1} a_{i,j} a_{k,j} \langle \bar{W}^{(j)}(\cdot) \rangle_t + 2 \sum_{j=1}^{N-1} \sum_{q=1, q>j}^{N-1} a_{i,j} a_{k,q} \langle \bar{W}^{(j)}(\cdot), \bar{W}^{(q)}(\cdot) \rangle_t \\ &= \sum_{j=1}^{N-1} a_{i,j} a_{k,j} t = \varrho_{i,k} t. \end{aligned}$$

Also ist  $W$  ein Wiener-Prozess mit  $d\langle W^{(i)}(\cdot), W^{(k)}(\cdot) \rangle_t = \varrho_{i,k} dt$  für alle  $1 \leq i, k \leq N-1$ .

Definieren wir nun  $(\sigma^{(i)}(t))_t$  mittels  $\bar{\sigma}^{(i,j)}(t) = \sigma^{(i)}(t) a_{i,j}$ , so können wir die erste Darstellungsvariante (4.2) der Libor-Raten-Dynamiken mithilfe des neuen Wiener-Prozesses  $W$  nun wie folgt in die Darstellungsvariante mit korrelierten Komponenten des Wiener-Prozesses überführen, denn wir erhalten

$$\begin{aligned} dL^{(i)}(t) &= L^{(i)}(t) \sum_{j=1}^{N-1} \bar{\sigma}^{(i,j)}(t) \left( \left( - \sum_{k=i+1}^{N-1} \frac{\delta L^{(k)}(t) \bar{\sigma}^{(k,j)}(t)}{\delta L^{(k)}(t) + 1} \right) dt + d\bar{W}^{(j)}(t) \right) \\ &= L^{(i)}(t) \sum_{j=1}^{N-1} \sigma^{(i)}(t) a_{i,j} \left( \left( - \sum_{k=i+1}^{N-1} \frac{\delta L^{(k)}(t) \sigma^{(k)}(t) a_{k,j}}{\delta L^{(k)}(t) + 1} \right) dt + d\bar{W}^{(j)}(t) \right) \\ &= L^{(i)}(t) \sigma^{(i)}(t) \left( \left( - \sum_{k=i+1}^{N-1} \frac{\delta L^{(k)}(t) \sigma^{(k)}(t)}{\delta L^{(k)}(t) + 1} \right) \sum_{j=1}^{N-1} a_{i,j} a_{k,j} dt + \sum_{j=1}^{N-1} a_{i,j} d\bar{W}^{(j)}(t) \right) \end{aligned}$$

$$= L^{(i)}(t) \sigma^{(i)}(t) \left( \left( - \sum_{k=i+1}^{N-1} \frac{\delta L^{(k)}(t) \sigma^{(k)}(t) \varrho_{i,k}}{\delta L^{(k)}(t) + 1} \right) dt + dW^{(i)}(t) \right). \quad (4.3)$$

Dieses Ergebnis können wir auch für die Simulation der Wiener-Prozesse ausnutzen. Für die Bewertung der Zinsderivate wollen wir später mit der Variante des Modells mit korrelierten Wiener-Prozessen arbeiten. Um für die Monte Carlo Simulation dann die Inkremente dieser Wiener-Prozesse zu simulieren, erzeugen wir zunächst einen  $(N-1)$ -dimensionalen Standard Wiener-Prozess  $(\bar{W}(t))_t$ , dessen Komponenten unkorreliert sind. Für die Korrelation betrachten wir dann, wie oben beschrieben, die Cholesky-Zerlegung  $A A^T$  der Korrelationsmatrix  $\Sigma$ . Die Einträge der Cholesky-Matrix  $A$ , einer unteren Dreiecksmatrix, ergeben sich zu

$$a_{i,j} = \begin{cases} \sqrt{1 - \sum_{k=1}^{i-1} a_{i,k}^2} & , 1 \leq i = j \leq N-1 \\ \frac{1}{a_{j,j}} (\varrho_{i,j} - \sum_{k=1}^{j-1} a_{i,k} a_{j,k}) & , 1 \leq j < i \leq N-1 \\ 0 & , \text{ sonst.} \end{cases}$$

Wir können anschließend also wieder  $(W(t))_t = (A \bar{W}(t))_t$  setzen und erhalten, wie gezeigt, einen  $(N-1)$ -dimensionalen Wiener-Prozess, dessen Korrelationsmatrix durch  $\Sigma$  gegeben ist.

Zu klären ist an dieser Stelle noch, in welcher Form die Volatilitäten  $\sigma(t)$  gegeben sind, worüber die nachfolgende Bemerkung Auskunft geben soll.

#### Bemerkung 4.2

In dieser Arbeit nehmen wir an, dass die Volatilitäten  $\sigma^{(i)}(t)$  für alle  $1 \leq i \leq N-1$  deterministische, zeitabhängige Funktionen sind. Des Weiteren seien diese Volatilitätsfunktionen für alle  $1 \leq i \leq N-1$  beschränkt.

Mithilfe dieser Annahmen und dem folgenden Satz aus [KP92] erhalten wir dann zusätzlich die Existenz einer sogenannten „starken Lösung“, also die schon oben angesprochene Existenz und Eindeutigkeit einer Lösung der zugrunde liegenden SDE (4.3).

#### Satz 4.3 (Existenz einer Lösung einer SDE)

Gelten für die SDE  $dX_t = a(t, X_t)dt + b(t, X_t)dW_t$  die Bedingungen

1. (Messbarkeit)  $a(t, x), b(t, x)$  sind  $(\mathcal{L}^2)$ -messbar in  $(t, x) \in [t_0, T] \times \mathbb{R}$
2. (Lipschitz Bedingung) Es existiert eine Konstante  $K_1 > 0$ , s.d.  $|a(t, x) - a(t, y)| \leq K_1|x - y|$  und  $|b(t, x) - b(t, y)| \leq K_1|x - y|$   
 $\forall t \in [t_0, T]$  und  $x, y \in \mathbb{R}$
3. (Lineare Wachstumsbeschränktheit) Es existiert eine Konstante  $K_2 > 0$ , s.d.  $|a(t, x)|^2 \leq K_2^2(1 + |x|^2)$  und  $|b(t, x)|^2 \leq K_2^2(1 + |x|^2) \forall t \in [t_0, T]$  und  $x \in \mathbb{R}$
4. (Anfangswert)  $X_{t_0}$  ist  $\mathcal{F}_{t_0}$ -messbar mit  $\mathbb{E}[|X_{t_0}|^2] < \infty$ ,

### 4.3 Die Euler-Maruyama Diskretisierung im Libor Marktmodell

---

so hat diese eine pfadweise eindeutige „starke Lösung“  $X_t$  auf  $[t_0, T]$  mit

$$\sup_{t_0 \leq t \leq T} \mathbb{E}[|X_t|^2] < \infty.$$

Zusammen mit der obigen Bemerkung 4.2 über die Beschränktheit der Volatilitätsfunktionen  $\sigma^{(i)}(t)$  und der Beobachtung, dass für den Ausdruck  $\frac{\delta L^{(k)}(t)\varrho_{i,k}}{\delta L^{(k)}(t)+1}$  aus dem Driftterm

$$a(t, L^{(i)}(t)) := L^{(i)}(t)\sigma^{(i)}(t)(-\sum_{k=i+1}^{N-1} \frac{\delta L^{(k)}(t)\sigma^{(k)}(t)\varrho_{i,k}}{\delta L^{(k)}(t)+1})$$

der SDE (4.3) die Ungleichung

$$\frac{\delta L^{(k)}(t)\varrho_{i,k}}{\delta L^{(k)}(t)+1} \leq 1$$

gilt, sehen wir leicht, dass alle Bedingungen des Satzes, also insbesondere die Bedingungen 2 und 3, erfüllt sind und erhalten somit für alle  $1 \leq i \leq N-1$  die Existenz einer Lösung der SDE (4.3), welche das Libor Marktmodell definiert.

### 4.3 Die Euler-Maruyama Diskretisierung im Libor Marktmodell

Für die Simulation der Libor-Raten-Pfade, auf der das Multilevel Monte Carlo Verfahren basiert, müssen wir die oben hergeleitete mehrdimensionale SDE bzw. den  $(N-1)$ -dimensionalen Vektor von Libor-Raten, dessen Einträge für alle  $1 \leq i \leq N-1$  die SDEs

$$dL^{(i)}(t) = L^{(i)}(t)\sigma^{(i)}(t)((-\sum_{k=i+1}^{N-1} \frac{\delta L^{(k)}(t)\sigma^{(k)}(t)\varrho_{i,k}}{\delta L^{(k)}(t)+1})dt + dW^{(i)}(t))$$

erfüllen, zunächst diskretisieren. Dies ist notwendig, da wir in der Praxis keine zeitstetigen Prozesse simulieren können, sondern die Pfade der stochastischen Prozesse nur an festgelegten diskreten Zeitpunkten auswerten können. Wir legen dazu vorher fest, wie groß die Schrittweite für die Diskretisierung sein soll und erzeugen dann zufällige Wiener-Inkremente, welche die diskretisierten Pfade der zugrunde liegenden SDE treiben. Nehmen wir an, dass die Diskretisierungsschrittweite der  $\eta$  Diskretisierungsschritte  $\Delta t$  beträgt, so erhalten wir nach [KP92] die Euler-Maruyama Approximation

$$L_{n+1}^{(i)} = L_n^{(i)} + L_n^{(i)}\sigma^{(i)}(t_n)(-\sum_{k=i+1}^{N-1} \frac{\delta L_n^{(k)}\sigma^{(k)}(t_n)\varrho_{i,k}}{\delta L_n^{(k)}+1})\Delta t + L_n^{(i)}\sigma^{(i)}(t_n)\Delta W_n^{(i)}$$

für  $0 \leq n \leq \eta-1$ , einen Anfangsvektor  $L_0 = L(t_0)$  und Wiener-Inkremente  $\Delta W_n^{(i)} = W^{(i)}(t_{n+1}) - W^{(i)}(t_n)$ .

## 5 Das Milstein Verfahren

### 5.1 Das Milstein Verfahren im Allgemeinen

Wie wir in Kapitel 3 gesehen haben, können wir durch die Verwendung des Multilevel Monte Carlo Verfahrens den Rechenaufwand im Vergleich zum Standard Monte Carlo Verfahren beispielsweise von der Ordnung  $\mathcal{O}(\epsilon^{-3})$  auf die Ordnung  $\mathcal{O}(\epsilon^{-2} \ln(\epsilon)^2)$  reduzieren, falls die Payoff-Funktion Lipschitz-stetig ist. Für die Simulationen haben wir die Pfade des „underlying assets“ dabei bisher stets mithilfe des Euler-Maruyama Verfahrens diskretisiert.

Im folgenden Abschnitt wollen wir nun zeigen, dass wir die Konvergenzgeschwindigkeit verbessern und damit auch den Rechenaufwand weiter reduzieren können, falls wir ein anderes Diskretisierungsverfahren, nämlich das Milstein Verfahren, nutzen. Dabei fügen wir zur Euler-Maruyama Diskretisierung Terme höherer Ordnung aus der (stochastischen) Taylor-Approximation hinzu (vgl. [KP92], [Gla03]). Anschließend zeigen wir, dass der Rechenaufwand in obiger Situation auf diese Weise weiter auf  $\mathcal{O}(\epsilon^{-2})$  abgesenkt werden kann. In [GDR13] und [Gil08b] zeigen Giles et al., dass dies bei dem Modell zugrunde liegenden skalaren SDEs für verschiedene ausgewählte Derivate funktioniert.

Betrachten wir nun erst einmal wieder eine allgemeine  $d$ -dimensionale SDE der Form

$$dX(t) = \mu(X(t), t)dt + \sigma(X(t), t)dW(t), 0 \leq t \leq T \quad (5.1)$$

mit  $\mu : \mathbb{R}^d \times \mathbb{R} \rightarrow \mathbb{R}$  und  $\sigma : \mathbb{R}^d \times \mathbb{R} \rightarrow \mathbb{R}^d \times \mathbb{R}^m$  und einem  $m$ -dimensionalen Standard Wiener-Prozess  $W$ . Dies bedeutet für jede Koordinate von  $X$ , d.h. für  $1 \leq i \leq d$ , gilt

$$dX^{(i)}(t) = \mu^{(i)}(X(t), t)dt + \sum_{j=1}^m \sigma^{(i,j)}(X(t), t)dW^{(j)}(t). \quad (5.2)$$

Nach [KP92] erhalten wir dazu für  $m = 1$  die Milstein Diskretisierung  $(X_n)_n$  für  $1 \leq i \leq d$  und alle  $1 \leq n \leq \eta - 1$  als

$$\begin{aligned} X_{n+1}^{(i)} &= X_n^{(i)} + \mu^{(i)}(X_n, t_n)\Delta t + \sigma^{(i,1)}(X_n, t_n)\Delta W_n^{(1)} \\ &\quad + \frac{1}{2} \left( \sum_{k=1}^d \sigma^{(k,1)}(X_n, t_n) \frac{\partial \sigma^{(i,1)}(X_n, t_n)}{\partial X_n^{(k)}} \right) ((\Delta W_n^{(1)})^2 - \Delta t), \end{aligned} \quad (5.3)$$

wobei  $X_n$  je eine Approximation für  $X(t_n)$  zu festgelegten Zeitpunkten  $0 \leq t_n \leq T$  mit Schrittweite  $\Delta t$  darstelle.

Für den allgemeinen Fall mit  $m > 1$  ist die Milstein Diskretisierung für  $1 \leq i \leq d$  und alle  $1 \leq n \leq \eta - 1$  ebenfalls nach [KP92] gegeben durch

$$\begin{aligned} X_{n+1}^{(i)} &= X_n^{(i)} + \mu^{(i)}(X_n, t_n)\Delta t + \sum_{j=1}^m \sigma^{(i,j)}(X_n, t_n)\Delta W_n^{(j)} \\ &\quad + \sum_{j_1, j_2=1}^m \sum_{k=1}^d \sigma^{(k, j_1)}(X_n, t_n) \frac{\partial \sigma^{(i, j_2)}(X_n, t_n)}{\partial X_n^{(k)}} I_{(j_1, j_2), n}, \end{aligned} \quad (5.4)$$

wobei  $I_{(j_1, j_2), n}$  das mehrfache Itô-Integral

$$I_{(j_1, j_2), n} = \int_{t_n}^{t_{n+1}} \int_{t_n}^{s_2} dW^{(j_1)}(s_1) dW^{(j_2)}(s_2) \quad (5.5)$$

bezeichne.

Dies bedeutet, dass für die Simulation von höherdimensionalen SDEs oft eine Betrachtung dieser iterierten Integrale erforderlich ist. Für die Simulation von den damit in Verbindung stehenden sogenannten „Lévy areas“, die wir nachfolgend definieren werden, gibt es, abgesehen von der Dimension 2, bislang keine effizienten Verfahren. Auf diese Problematik wollen wir nun etwas weiter eingehen.

Dazu definieren wir die „Lévy areas“ wie in [Wik01] mithilfe der iterierten Integrale aus Gleichung (5.5) für alle  $1 \leq j_1, j_2 \leq m$  und alle  $1 \leq n \leq \eta - 1$  als

$$A_{j_1, j_2, n} := \frac{1}{2}(I_{(j_1, j_2), n} - I_{(j_2, j_1), n}), \quad (5.6)$$

sodass dann offensichtlich  $A_{j_1, j_2, n} = -A_{j_2, j_1, n}$  gilt.

Weiterhin benötigen wir das folgende

**Lemma 5.1**

Für  $j_1 \neq j_2$  gilt

$$I_{(j_1, j_2), n} + I_{(j_2, j_1), n} = \Delta W_n^{(j_1)} \Delta W_n^{(j_2)}.$$

*Beweis:* Für den Beweis zeigen wir zunächst, dass die Gleichung

$$\begin{aligned} & \int_{t_n}^{t_{n+1}} W^{(j_1)}(s_2) dW^{(j_2)}(s_2) + \int_{t_n}^{t_{n+1}} W^{(j_2)}(s_1) dW^{(j_1)}(s_1) \\ &= W^{(j_1)}(t_{n+1}) W^{(j_2)}(t_{n+1}) - W^{(j_1)}(t_n) W^{(j_2)}(t_n), \end{aligned} \quad (5.7)$$

die in Differentialschreibweise die Form

$$W^{(j_1)}(t) dW^{(j_2)}(t) + W^{(j_2)}(t) dW^{(j_1)}(t) = d(W^{(j_1)} W^{(j_2)})(t)$$

besitzt, hier erfüllt ist. Dies ist der Fall, da wir mit der Itô-Formel

$$d(W^{(j_1)} W^{(j_2)})(t) = W^{(j_1)}(t) dW^{(j_2)}(t) + W^{(j_2)}(t) dW^{(j_1)}(t) + d\langle W^{(j_1)}(\cdot), W^{(j_2)}(\cdot) \rangle_t$$

erhalten und wegen der Unabhängigkeit der Komponenten des Wiener-Prozesses  $W$  (für  $j_1 \neq j_2$ )  $\langle W^{(j_1)}(\cdot), W^{(j_2)}(\cdot) \rangle_t = 0$  für alle  $t \geq 0$  gilt.

Mit dieser Gleichung gilt dann, wie behauptet,

$$\begin{aligned} & I_{(j_1, j_2), n} + I_{(j_2, j_1), n} \\ &= \int_{t_n}^{t_{n+1}} \int_{t_n}^{s_2} dW^{(j_1)}(s_1) dW^{(j_2)}(s_2) + \int_{t_n}^{t_{n+1}} \int_{t_n}^{s_1} dW^{(j_2)}(s_2) dW^{(j_1)}(s_1) \\ &= \int_{t_n}^{t_{n+1}} W^{(j_1)}(s_2) dW^{(j_2)}(s_2) + \int_{t_n}^{t_{n+1}} W^{(j_2)}(s_1) dW^{(j_1)}(s_1) \\ &\quad - W^{(j_1)}(t_n)(W^{(j_2)}(t_{n+1}) - W^{(j_2)}(t_n)) - W^{(j_2)}(t_n)(W^{(j_1)}(t_{n+1}) - W^{(j_1)}(t_n)) \end{aligned}$$

$$\begin{aligned}
 &\stackrel{(5.7)}{=} W^{(j_1)}(t_{n+1})W^{(j_2)}(t_{n+1}) - W^{(j_1)}(t_n)W^{(j_2)}(t_n) \\
 &\quad - W^{(j_1)}(t_n)\Delta W_n^{(j_2)} - W^{(j_2)}(t_n)W^{(j_1)}(t_{n+1}) + W^{(j_2)}(t_n)W^{(j_1)}(t_n) \\
 &= W^{(j_1)}(t_{n+1})\Delta W_n^{(j_2)} - W^{(j_1)}(t_n)\Delta W_n^{(j_2)} = \Delta W_n^{(j_1)}\Delta W_n^{(j_2)}.
 \end{aligned}$$

□

Mithilfe dieses Lemmas und der Definition (5.6) der „Lévy areas“  $A_{j_1, j_2, n}$  erhalten wir für  $I_{(j_1, j_2), n}$  nun eine alternative Darstellung.

**Lemma 5.2**

Für alle  $1 \leq j_1 \leq m$  gilt

$$I_{(j_1, j_1), n} = \frac{(\Delta W_n^{(j_1)})^2 - \Delta t}{2},$$

für alle  $1 \leq j_1 \neq j_2 \leq m$  gilt

$$I_{(j_1, j_2), n} = \frac{\Delta W_n^{(j_1)}\Delta W_n^{(j_2)}}{2} + A_{j_1, j_2, n}.$$

*Beweis:* Für den Beweis benötigen wir zunächst die Gleichung

$$\begin{aligned}
 &\int_{t_n}^{t_{n+1}} W^{(j_1)}(t)dW^{(j_1)}(t) \\
 &= \frac{1}{2}((\Delta W_n^{(j_1)})^2 + 2W^{(j_1)}(t_{n+1})W^{(j_1)}(t_n) - 2(W^{(j_1)}(t_n))^2 - \Delta t), \quad (5.8)
 \end{aligned}$$

die erfüllt ist, da wieder mit der Itô-Formel

$$d((W^{(j_1)})^2)(t) = 2W^{(j_1)}(t)dW^{(j_1)}(t) + d\langle W^{(j_1)}(\cdot) \rangle_t = 2W^{(j_1)}(t)dW^{(j_1)}(t) + dt$$

gilt, was sich in Integralschreibweise zu

$$\begin{aligned}
 &\int_{t_n}^{t_{n+1}} W^{(j_1)}(t)dW^{(j_1)}(t) \\
 &= \frac{1}{2}((W^{(j_1)}(t_{n+1}))^2 - (W^{(j_1)}(t_n))^2 - \Delta t) \\
 &= \frac{1}{2}((\Delta W_n^{(j_1)})^2 + 2W^{(j_1)}(t_{n+1})W^{(j_1)}(t_n) - 2(W^{(j_1)}(t_n))^2 - \Delta t)
 \end{aligned}$$

umformen lässt.

Die erste Gleichung des Lemmas erhalten wir nun mit

$$\begin{aligned}
 I_{(j_1, j_1), n} &= \int_{t_n}^{t_{n+1}} \int_{t_n}^{s_2} dW^{(j_1)}(s_1)dW^{(j_1)}(s_2) \\
 &= \int_{t_n}^{t_{n+1}} W^{(j_1)}(s_2)dW^{(j_1)}(s_2) - \int_{t_n}^{t_{n+1}} W^{(j_1)}(t_n)dW^{(j_1)}(s_2) \\
 &\stackrel{(5.8)}{=} \frac{1}{2}((\Delta W_n^{(j_1)})^2 - \Delta t + 2W^{(j_1)}(t_{n+1})W^{(j_1)}(t_n) - 2(W^{(j_1)}(t_n))^2) \\
 &\quad - W^{(j_1)}(t_n)(W^{(j_1)}(t_{n+1}) - W^{(j_1)}(t_n))
 \end{aligned}$$

## 5.2 Das Milstein Verfahren ohne „Lévy area“-Simulation

---

$$= \frac{1}{2}((\Delta W_n^{(j_1)})^2 - \Delta t).$$

Weiter gilt für  $j_1 \neq j_2$

$$\begin{aligned} I_{(j_1,j_2),n} + I_{(j_2,j_1),n} &\stackrel{\text{Lemma 5.1}}{=} \Delta W_n^{(j_1)} \Delta W_n^{(j_2)} \\ \Leftrightarrow 2I_{(j_1,j_2),n} &= \Delta W_n^{(j_1)} \Delta W_n^{(j_2)} + I_{(j_1,j_2),n} - I_{(j_2,j_1),n} \\ \Leftrightarrow I_{(j_1,j_2),n} &= \frac{1}{2}(\Delta W_n^{(j_1)} \Delta W_n^{(j_2)}) + A_{j_1,j_2,n}, \end{aligned}$$

was den Beweis abschließt.  $\square$

Betrachten wir nun noch einmal die obige SDE (5.2) und setzen

$$h_{i,j_1,j_2}(X_n) = \sum_{k=1}^d \sigma^{(k,j_1)}(X_n, t_n) \frac{\partial \sigma^{(i,j_2)}(X_n, t_n)}{\partial X_n^{(k)}}. \quad (5.9)$$

Dann erhalten wir mithilfe des Lemmas 5.2 die Milstein Diskretisierung der SDE (5.2) für  $m > 1$  nun als

$$\begin{aligned} X_{n+1}^{(i)} = & X_n^{(i)} + \mu^{(i)}(X_n, t_n) \Delta t + \sum_{j=1}^m \sigma^{(i,j)}(X_n, t_n) \Delta W_n^{(j)} \\ & + \sum_{j_1,j_2=1}^m h_{i,j_1,j_2}(X_n) \left( \frac{1}{2}(\Delta W_n^{(j_1)} \Delta W_n^{(j_2)} - \mathbf{1}_{\{j_1=j_2\}} \Delta t) + A_{j_1,j_2,n} \right). \quad (5.10) \end{aligned}$$

Analog dazu lässt sich zeigen, dass die Milstein Diskretisierung der SDE (5.1), für den Fall, dass  $W$  einen  $m$ -dimensionalen Vektor korrelierter Wiener-Prozesse darstellt, die Form

$$\begin{aligned} X_{n+1}^{(i)} = & X_n^{(i)} + \mu^{(i)}(X_n, t_n) \Delta t + \sum_{j=1}^m \sigma^{(i,j)}(X_n, t_n) \Delta W_n^{(j)} \\ & + \sum_{j_1,j_2=1}^m h_{i,j_1,j_2}(X_n) \left( \frac{1}{2}(\Delta W_n^{(j_1)} \Delta W_n^{(j_2)} - \varrho_{j_1,j_2} \Delta t) + A_{j_1,j_2,n} \right) \quad (5.11) \end{aligned}$$

besitzt, wobei  $\varrho_{j_1,j_2}$  die Korrelation zwischen den Komponenten  $j_1$  und  $j_2$  des Wiener-Prozesses  $W$  bezeichne.

## 5.2 Das Milstein Verfahren ohne „Lévy area“-Simulation

Wie schon oben erwähnt, ist es nach [GS12] bisher nur schwer und meist mit ineffektiven Methoden möglich, die für die obige Milstein Diskretisierung notwendigen iterierten Integrale bzw. „Lévy areas“ zu simulieren, was einen entscheidenden Nachteil von Diskretisierungsverfahren höherer Ordnung bei einer Anwendung für höherdimensionale SDEs darstellt.

Müller-Gronbach hat in [Mül02] gezeigt, dass es für allgemeine höherdimensionale SDEs ohne die Betrachtung von „Lévy areas“ jedoch nicht möglich ist, eine bessere starke Konvergenzrate als die des Euler-Maruyama Diskretisierungsverfahrens

zu erzielen. Dies bedeutet, dass mittels einer einfachen Änderung des Diskretisierungsverfahrens zur Milstein Diskretisierung ohne die Simulation der „Lévy areas“ auch keine Verbesserung des Rechenaufwands erzielt werden kann.

Giles und Szpruch haben allerdings in [GS12] mithilfe der Methode der „antithetischen Variablen“ einen Multilevel-Schätzer entwickelt, welcher auch ohne die Simulation von „Lévy areas“ eine Varianz besitzt, die mit derselben Ordnung konvergiert wie der Multilevel-Schätzer mit einer Milstein Diskretisierung für skalare SDEs, bei denen keine iterierten Integrale betrachtet werden müssen. Dies bedeutet, dass für die Varianzen  $V_l$  der Korrekturterme  $\tilde{P}_l - \tilde{P}_{l-1}$  des Monte Carlo Schätzers  $\hat{Y}$ , die von der starken Konvergenzrate des verwendeten Diskretisierungsverfahrens abhängen, statt einer Konvergenzordnung  $\mathcal{O}(h_l)$  die Ordnung  $\mathcal{O}(h_l^2)$  erreicht werden kann. Diesen verbesserten Multilevel-Schätzer wollen wir nun in den folgenden Abschnitten herleiten.

### 5.2.1 Antithetische Variablen

Um den verbesserten Multilevel-Schätzer einzuführen, wollen wir zunächst die allgemeine Varianzreduktionsmethode der „antithetischen Variablen“ vorstellen. Dazu orientieren wir uns an der Darstellung in [Gla03].

Die Idee der sogenannten „antithetischen Variablen“ beruht darauf, dass wir bei einer Simulation, die auf der Erzeugung von identisch verteilten Zufallszahlen  $Z_1, Z_2, \dots$  beruht, diese bisher erzeugten Zufallszahlen nutzen, um damit weitere identisch verteilte Zufallszahlen, sogenannte „antithetische Gegenstücke“, zu erzeugen. Dies soll auf eine Weise geschehen, dass dadurch die Varianz der zugehörigen Schätzer verringert wird. Für identisch standardnormalverteilte Zufallszahlen  $Z_1, Z_2, \dots$  können wir beispielsweise zusätzlich die Folge  $\bar{Z}_1 = -Z_1, \bar{Z}_2 = -Z_2, \dots$  dieser Zufallszahlen betrachten, welche dann aufgrund der Symmetrie der Standardnormalverteilung dieselbe Verteilung besitzen wie die  $Z_1, Z_2, \dots$ . Die normalverteilten Zufallszahlen  $(Z_i)_i$  können dann z.B. für die Simulation der Wiener-Inkredente bzw. der diskretisierten Wiener-Pfade genutzt werden. Durch die Nutzung der Paare von Zufallszahlen erhoffen wir uns, eine Verringerung der Varianz der simulierten Stichproben zu erreichen.

Im Folgenden wollen wir also eine Folge von Paaren von Zufallszahlen  $(X_1, \bar{X}_1), (X_2, \bar{X}_2), \dots, (X_n, \bar{X}_n)$  betrachten, wobei die  $(X_i)_i$  mithilfe der  $(Z_i)_i$ , die  $(\bar{X}_i)_i$  mithilfe der  $(\bar{Z}_i)_i$  simuliert werden. Dabei sind die Paare untereinander stochastisch unabhängig und identisch verteilt. Ausgehend von den auf diese Weise erzeugten Zufallszahlen können wir dann einen Schätzer der Form

$$\hat{Y}^{\text{AV}} = \frac{1}{n} \sum_{i=1}^n \frac{1}{2} (X_i + \bar{X}_i) \quad (5.12)$$

verwenden, für den „antithetische Variablen“ genutzt werden.

Zu klären ist nun, unter welchen Bedingungen der obige Schätzer tatsächlich eine geringere Varianz als ein Standard Schätzer ohne die Verwendung von „antithetischen Gegenstücken“ besitzt. Dazu wollen wir annehmen, dass die Erzeugung eines Paares  $(X_i, \bar{X}_i)$  denselben Rechenaufwand wie die Erzeugung zweier unabhängiger Zufallszahlen besitzt. Wir ignorieren folglich mögliche Einsparungen

beim Rechenaufwand, die dadurch entstehen könnten, dass zur Erzeugung der  $\bar{Z}_i$  im obigen Fall beispielsweise nur die Vorzeichen umgedreht werden müssen. Dies macht insbesondere dann Sinn, wenn die Erzeugung der  $Z_i$  nur einen kleinen Anteil an der Simulation der  $X_i$  ausmacht. Im Folgenden werden wir deshalb die Varianz des Schätzers  $\hat{Y}^{\text{AV}}$  mit der eines Standard Schätzers  $\hat{Y}^{\text{Std}} := \frac{1}{2n} \sum_{i=1}^{2n} X_i$ , für den dieselbe Anzahl an Simulationen genutzt wird, vergleichen.  
Betrachten wir dazu zunächst den Schätzer  $\hat{Y}^{\text{AV}}$ , so erhalten wir mithilfe der Unabhängigkeit und der identischen Verteilung der Paare  $(X_i, \bar{X}_i)$  sowie der identischen Verteilung der  $X_i$  und der  $\bar{X}_i$

$$\begin{aligned}\mathbb{V}[\hat{Y}^{\text{AV}}] &= \frac{1}{4n} \mathbb{V}[X_1 + \bar{X}_1] = \frac{1}{4n} (\mathbb{V}[X_1] + \mathbb{V}[\bar{X}_1] + 2\text{Cov}[X_1, \bar{X}_1]) \\ &= \frac{1}{2n} (\mathbb{V}[X_1] + \text{Cov}[X_1, \bar{X}_1]).\end{aligned}$$

Die Varianz des Standard Schätzers  $\hat{Y}^{\text{Std}}$  ergibt sich zu  $\mathbb{V}[\hat{Y}^{\text{Std}}] = \frac{1}{2n} \mathbb{V}[X_1]$ . D.h. die Bedingung, damit dieses Verfahren zu einer Varianzreduktion führt, lautet

$$\text{Cov}[X_1, \bar{X}_1] < 0.$$

Dies bedeutet, dass die Methode der „antithetischen Variablen“ im Allgemeinen funktioniert, falls eine negative Korrelation der simulierten Zufallszahlen  $X_i$  und ihrer jeweiligen „antithetischen Gegenstücke“  $\bar{X}_i$  vorliegt.

### 5.2.2 Ein verallgemeinerter Multilevel Monte Carlo Schätzer

Bevor wir zur Konstruktion des „antithetischen“ Multilevel Monte Carlo Schätzers kommen, führen wir noch einen etwas verallgemeinerten Multilevel Monte Carlo Schätzer ein, welchen Giles in [Gil08b] diskutiert.

Bisher sind die Approximationen  $\tilde{P}_l$  und  $\tilde{P}_{l-1}$  für die feinere und gröbere zweier aufeinanderfolgender Diskretisierungsschrittweiten eines Simulationslevels innerhalb des gesamten Multilevel-Schätzers für  $\mathbb{E}[\tilde{P}_L] = \mathbb{E}[\tilde{P}_0] + \sum_{l=1}^L \mathbb{E}[\tilde{P}_l - \tilde{P}_{l-1}]$  auf dieselbe Art und Weise erstellt worden und haben sich nur bzgl. der Diskretisierungsschrittweite unterschieden. Für die verallgemeinerte Variante wollen wir nun die Möglichkeit zulassen, für die feinere und gröbere Schrittweite eines Levels je eine unterschiedliche Approximation  $\tilde{P}_l^f$  bzw.  $\tilde{P}_{l-1}^g$  für den Payoff des zu bewertenden Derivats zu nutzen. Dafür müssen wir voraussetzen, dass die Gleichung

$$\mathbb{E}[\tilde{P}_l^f] = \mathbb{E}[\tilde{P}_l^g] \quad \forall 1 \leq l \leq L \tag{5.13}$$

erfüllt wird, damit die den Gesamtschätzer definierende Gleichung

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

weiterhin Gültigkeit besitzt.

Der zentrale Satz über das Multilevel Monte Carlo Verfahren ist somit auch für diesen leicht verallgemeinerten Schätzer anwendbar. Gleichzeitig besitzen wir jedoch die Möglichkeit Multilevel Monte Carlo Schätzer zu konstruieren, die flexibler sind und somit zu einer geringeren Varianz der Korrekturterme  $\tilde{P}_l^f - \tilde{P}_{l-1}^g$

führen können. Dies führt wiederum zu einem größeren  $\beta$  in Bedingung (iii) des zentralen Satzes 3.4 und somit zu einer möglichen weiteren Verringerung des Rechenaufwands. Dass dieses Verfahren tatsächlich zu enormen Konvergenzverbesserungen – vor allem bei der Bewertung von pfadabhängigen Derivaten – führen kann, zeigt Giles in [Gil08b] numerisch anhand von allgemeinen skalaren SDEs und der Bewertung von Asiatischen, Lookback- und Barriere-Optionen.

### 5.2.3 Der „Antithetische“ Multilevel Monte Carlo Schätzer

Auch bei der Konstruktion eines „antithetischen“ Schätzers wollen wir in Anlehnung an [GS12] das soeben vorgestellte verallgemeinerte Verfahren verwenden und unterschiedliche Approximationen bzgl. der beiden aufeinanderfolgenden Diskretisierungsschrittweiten eines Simulationslevels nutzen.

Die Idee dabei ist es, die Approximation  $\tilde{P}_{l-1}^g$  des zufälligen Payoffs eines zu bewertenden Derivats, welche bzgl. der größeren der je zwei Diskretisierungsschrittweiten eines Simulationslevels  $l$  erstellt wird, auf dieselbe Weise durchzuführen, wie wir es üblicherweise bei der Multilevel Monte Carlo Simulation handhaben. Das bedeutet, dass wir für jede Simulation eines Samples die Wiener-Inkremeante bzgl. der größeren der beiden Schrittweiten, welche hier durch den Vektor  $(Z_i^{l-1})_i$  repräsentiert werden, nutzen, um damit zunächst den Wert  $X_{l-1}^g$  des „underlying assets“, dessen Pfad mithilfe eines Diskretisierungsverfahrens approximiert wurde, in  $T$  und anschließend den simulierten Payoff  $P(X_{l-1}^g) =: \tilde{P}_{l-1}^g$  des davon abhängigen Derivats zu bestimmen.

Die Approximation  $\tilde{P}_l^f$ , welche bzgl. der feineren Diskretisierungsschrittweite erstellt wird, soll nun jedoch auf eine andere Weise bestimmt werden. Es werden also nicht lediglich die simulierten normalverteilten Zufallsvariablen  $(Z_i^l)_i$  genutzt, um damit einen approximierten Pfad des „underlying assets“ zu erzeugen und anschließend den Payoff des Derivats, abhängig vom Wert  $X_l^f$  des „underlying assets“ in  $T$ , zu bestimmen, sondern zusätzlich dazu die „antithetischen Gegenstücke“  $(\bar{Z}_i^l)_i$  der simulierten Zufallsvariablen  $(Z_i^l)_i$  verwendet, um damit einen weiteren approximierten Pfad des „underlying assets“ zu erzeugen. Dies liefert uns zwei aufgrund der identischen Verteilung der  $Z_i$  und  $\bar{Z}_i$  ebenfalls identisch verteilte Zufallsvariablen  $X_l^f$  und  $X_l^a$ , mit deren Hilfe der approximierte Payoff  $\tilde{P}_l^f$  als

$$\tilde{P}_l^f := \frac{1}{2}(P(X_l^f) + P(X_l^a))$$

bestimmt wird.

Wie gefordert stellen wir auf diese Weise sicher, dass die notwendige Bedingung (5.13) wegen der identischen Verteilung der  $X_l^g$ ,  $X_l^f$  und  $X_l^a$  und der daraus resultierenden Gleichungskette

$$\mathbb{E}[\tilde{P}_l^f] = \mathbb{E}\left[\frac{1}{2}(P(X_l^f) + P(X_l^a))\right] = \mathbb{E}[P(X_l^f)] = \mathbb{E}[P(X_l^g)] = \mathbb{E}[\tilde{P}_l^g]$$

für alle  $1 \leq l \leq L$  erfüllt ist.

Für die theoretische Analyse der Konvergenzgeschwindigkeit der Varianz des neuen Schätzers verweisen wir auf eine ausführliche Betrachtung in [GS12].

An dieser Stelle geben wir nur die drei wichtigsten Lemmata aus [GS12], welche uns letztlich die gewünschte Konvergenzordnung zusichern, ohne Beweis an.

**Lemma 5.3**

Existieren für eine Payoff-Funktion  $P \in C^2(\mathbb{R}^d, \mathbb{R})$  Konstanten  $C_1, C_2$ , sodass für alle  $x \in \mathbb{R}^d$

$$\left\| \frac{\partial P}{\partial x} \right\| \leq C_1, \left\| \frac{\partial^2 P}{\partial x^2} \right\| \leq C_2$$

gilt, dann erhalten wir für  $p \geq 2$  die Ungleichung

$$\begin{aligned} \mathbb{E}[(\tilde{P}_l^f - \tilde{P}_{l-1}^g)^p] &= \mathbb{E}\left[\left(\frac{1}{2}(P(X_l^f) + P(X_l^a)) - P(X_{l-1}^g)\right)^p\right] \\ &\leq 2^{p-1} C_1^p \mathbb{E}\left[\left\|\frac{1}{2}(X_l^f + X_l^a) - X_{l-1}^g\right\|^p\right] + 2^{-(p+1)} C_2^p \mathbb{E}\left[\|X_l^f - X_l^a\|^{2p}\right]. \end{aligned}$$

Mit den nächsten beiden Lemmata wollen wir die Summanden in der letzten Zeile des obigen Lemmas weiter abschätzen.

**Lemma 5.4**

Für alle ganzen Zahlen  $p \geq 2$  existiert eine Konstante  $K_p$ , sodass gilt

$$\mathbb{E}\left[\max_{0 \leq n \leq \eta_{l-1}} \|X_{l,n}^f - X_{l,n}^a\|^p\right] \leq K_p h_{l-1}^{p/2}.$$

Dabei haben wir den Index  $n$  mit  $1 \leq n \leq \eta_{l-1}$ , wobei  $\eta_{l-1}$  die Anzahl an Diskretisierungsschritten angibt, die für die Diskretisierung mit der (größeren) Schrittweite  $h_{l-1}$  notwendig sind, ergänzt, um die Werte des „underlying assets“ an den Diskretisierungsstellen  $t_n = nh_{l-1}$  zu bezeichnen. Damit sind obige  $X_l^g, X_l^f$  und  $X_l^a$  in dieser Notation identisch mit den  $X_{l,\eta_{l-1}}^g, X_{l,\eta_{l-1}}^f$  und  $X_{l,\eta_{l-1}}^a$ , die die Werte des „underlying assets“ in  $T = t_{\eta_{l-1}} = \eta_{l-1} h_{l-1}$  angeben. Um einheitliche Bezeichnungen zu gewährleisten, führen wir dabei für einen Indexschritt bzgl. der größeren Schrittweite je  $M$   $M$ -tel Indexschritte bzgl. der feineren Schrittweite durch. Mithilfe dieser Notation können wir auch das nächste Lemma angeben.

**Lemma 5.5**

Für alle  $p \geq 2$  existiert eine Konstante  $\bar{K}_p$ , sodass gilt

$$\mathbb{E}\left[\max_{0 \leq n \leq \eta_{l-1}} \left\| \frac{1}{2}(X_{l,n}^f + X_{l,n}^a) - X_{l-1,n}^g \right\|^p\right] \leq \bar{K}_p h_{l-1}^p.$$

Mit der Wahl von  $p = 2$  in Lemma 5.3,  $p = 4$  in Lemma 5.4 und  $p = 2$  in Lemma 5.5 sowie einer Konstanten  $\tilde{c} > 0$  folgt für die Varianz der Korrekturterme

$$\begin{aligned} \mathbb{V}[\tilde{P}_l^f - \tilde{P}_{l-1}^g] &= \mathbb{V}\left[\frac{1}{2}(P(X_l^f) + P(X_l^a)) - P(X_{l-1}^g)\right] \\ &\leq \mathbb{E}\left[\left(\frac{1}{2}(P(X_l^f) + P(X_l^a)) - P(X_{l-1}^g)\right)^2\right] \\ &\stackrel{\text{Lemma 5.3}}{\leq} 2C_1^2 E\left[\left\|\frac{1}{2}(X_l^f + X_l^a) - X_{l-1}^g\right\|^2\right] + 2^{-3} C_2^2 \mathbb{E}\left[\|X_l^f - X_l^a\|^4\right] \\ &\stackrel{\text{Lemmat 5.5, 5.4}}{\leq} 2C_1^2 \bar{K}_2 h_{l-1}^2 + 2^{-3} C_2^2 K_4 h_{l-1}^2 \leq \tilde{c} h_l^2. \end{aligned}$$

Die Varianz der Korrekturterme besitzt somit eine obere Schranke der Ordnung  $\mathcal{O}(h_l^2)$ . Diese Konvergenzordnung entspricht der Konvergenzordnung, die bei skalaren SDEs mit dem Milstein Diskretisierungsverfahren erreicht werden kann und stellt damit eine enorme Verbesserung der Konvergenzordnung  $\mathcal{O}(h_l)$  des Euler-Maruyama Verfahrens dar.

Voraussetzung dafür ist, dass bei der Simulation der  $X_l^g$ ,  $X_l^f$  und  $X_l^a$  das Milstein Diskretisierungsverfahren genutzt wird, wobei jedoch die Simulation der „Lévy areas“, also der Term  $A_{j_1,j_2,n}$  in der Darstellung (5.11), vernachlässigt wird. Um zu demonstrieren, wie genau die diskretisierten Pfade eines „underlying assets“ aussehen, wählen wir hier wie in [GS12] der Einfachheit halber einen Verfeinigungsfaktor von  $M = 2$ . Des Weiteren bezeichnen wir die Diskretisierungsschrittweite  $h_{l-1}$  hier analog zur Darstellung von Diskretisierungsverfahren in früheren Kapiteln mit  $\Delta t$ . Die Milstein Diskretisierung, bei der wir den „Lévy area“-Term vernachlässigen und stattdessen mit „antithetischen Variablen“ arbeiten, lautet für den Pfad bzgl. der feineren Schrittweite  $h_l$  und für  $1 \leq i \leq d$  sowie  $0 \leq n \leq \eta_{l-1}$  dann

$$\begin{aligned} X_{l,n+\frac{1}{2}}^{f(i)} &= X_{l,n}^{f(i)} + \mu^{(i)}(X_{l,n}^f, t_n) \left( \frac{1}{2} \Delta t \right) + \sum_{j=1}^m \sigma^{(i,j)}(X_{l,n}^f, t_n) \Delta W_{n+\frac{1}{2}}^{f(j)} \\ &\quad + \sum_{j_1,j_2=1}^m h_{i,j_1,j_2}(X_{l,n}^f) \frac{1}{2} (\Delta W_{n+\frac{1}{2}}^{f(j_1)} \Delta W_{n+\frac{1}{2}}^{f(j_2)} - \varrho_{j_1,j_2}(\frac{1}{2} \Delta t)) \end{aligned}$$

bzw.

$$\begin{aligned} X_{l,n+1}^{f(i)} &= X_{l,n+\frac{1}{2}}^{f(i)} + \mu^{(i)}(X_{l,n+\frac{1}{2}}^f, t_{n+\frac{1}{2}}) \left( \frac{1}{2} \Delta t \right) + \sum_{j=1}^m \sigma^{(i,j)}(X_{l,n+\frac{1}{2}}^f, t_{n+\frac{1}{2}}) \Delta W_{n+1}^{f(j)} \\ &\quad + \sum_{j_1,j_2=1}^m h_{i,j_1,j_2}(X_{l,n+\frac{1}{2}}^f) \frac{1}{2} (\Delta W_{n+1}^{f(j_1)} \Delta W_{n+1}^{f(j_2)} - \varrho_{j_1,j_2}(\frac{1}{2} \Delta t)), \end{aligned}$$

wobei  $\Delta W_{n+\frac{1}{2}}^{f(j)}$  bzw.  $\Delta W_{n+1}^{f(j)}$  je zwei Wiener-Inkremente des  $j$ -ten Wiener-Prozesses bzgl. der feineren Schrittweite  $h_l = \frac{1}{2}h_{l-1} = \frac{1}{2}\Delta t$  darstellen.

Zu bemerken ist, dass wir bei der Diskretisierung – wie schon erwähnt – im Vergleich zur größeren Schrittweite hier nur je einen „halben Schritt“ forschreiten.

Für die Wiener-Inkremente des Pfades bzgl. der größeren Schrittweite nutzen wir, wie beim Multilevel Verfahren üblich,

$$\Delta W_{n+1}^{g(j)} = \Delta W_{n+\frac{1}{2}}^{f(j)} + \Delta W_{n+1}^{f(j)}$$

und erhalten die Diskretisierung

$$\begin{aligned} X_{l-1,n+1}^g &= X_{l-1,n}^g + \mu^{(i)}(X_{l-1,n}^g, t_n) \Delta t + \sum_{j=1}^m \sigma^{(i,j)}(X_{l-1,n}^g, t_n) \Delta W_{n+1}^{g(j)} \\ &\quad + \sum_{j_1,j_2=1}^d h_{i,j_1,j_2}(X_{l-1,n}^g) \frac{1}{2} (\Delta W_{n+1}^{g(j_1)} \Delta W_{n+1}^{g(j_2)} - \varrho_{j_1,j_2} \Delta t). \end{aligned}$$

Um die Simulation der „Lévy areas“ zu umgehen, benötigen wir für die Approximation des feineren Levels – wie in diesem Kapitel beschrieben – nun zusätzlich die diskretisierten Pfade des „antithetischen Gegenstücks“. Wie in [GS12] vertauschen wir bei der Diskretisierung dazu je die beiden Wiener-Inkremente  $\Delta W_{n+\frac{1}{2}}^f (j)$  und  $\Delta W_{n+1}^f (j)$ . Auf diese Weise sind die „antithetischen Gegenstücke“ ebenfalls identisch verteilt und die notwendige Bedingung (5.13) wird eingehalten. Wir erhalten dann die Diskretisierung

$$\begin{aligned} X_{l,n+\frac{1}{2}}^{a(i)} = & X_{l,n}^{a(i)} + \mu^{(i)}(X_{l,n}^a, t_n) \left( \frac{1}{2} \Delta t \right) + \sum_{j=1}^m \sigma^{(i,j)}(X_{l,n}^a, t_n) \Delta W_{n+1}^f (j) \\ & + \sum_{j_1, j_2=1}^m h_{i,j_1,j_2}(X_{l,n}^a) \frac{1}{2} (\Delta W_{n+1}^f (j_1) \Delta W_{n+1}^f (j_2) - \varrho_{j_1,j_2}(\frac{1}{2} \Delta t)) \end{aligned}$$

bzw.

$$\begin{aligned} X_{l,n+\frac{1}{2}}^{a(i)} = & X_{l,n+\frac{1}{2}}^{a(i)} + \mu^{(i)}(X_{l,n+\frac{1}{2}}^a, t_{n+\frac{1}{2}}) \left( \frac{1}{2} \Delta t \right) + \sum_{j=1}^m \sigma^{(i,j)}(X_{l,n+\frac{1}{2}}^a, t_{n+\frac{1}{2}}) \Delta W_{n+\frac{1}{2}}^f (j) \\ & + \sum_{j_1, j_2=1}^m h_{i,j_1,j_2}(X_{l,n+\frac{1}{2}}^a) \frac{1}{2} (\Delta W_{n+\frac{1}{2}}^f (j_1) \Delta W_{n+\frac{1}{2}}^f (j_2) - \varrho_{j_1,j_2}(\frac{1}{2} \Delta t)). \end{aligned}$$

### 5.3 Das Milstein Verfahren im Libor Marktmodell

Nach diesem Exkurs über das Milstein Verfahren ohne Simulation der „Lévy areas“ kommen wir noch einmal auf die allgemeine Version (5.11) des Milstein Diskretisierungsverfahrens für höherdimensionale SDEs unter Verwendung eines korrelierten  $m$ -dimensionalen Wiener-Prozesses zurück. Die dabei zugrunde liegende allgemeine SDE (5.2)

$$dX^{(i)}(t) = \mu^{(i)}(X(t), t) dt + \sum_{j=1}^m \sigma^{(i,j)}(X(t), t) dW^{(j)}(t)$$

sowie ihre Milstein Diskretisierung

$$\begin{aligned} X_{n+1}^{(i)} = & X_n^{(i)} + \mu^{(i)}(X_n, t_n) \Delta t + \sum_{j=1}^m \sigma^{(i,j)}(X_n, t_n) \Delta W_n^{(j)} \\ & + \sum_{j_1, j_2=1}^m h_{i,j_1,j_2}(X_n) \left( \frac{1}{2} (\Delta W_n^{(j_1)} \Delta W_n^{(j_2)} - \varrho_{j_1,j_2} \Delta t) + A_{j_1,j_2,n} \right). \end{aligned}$$

geben wir hier noch einmal zur Erinnerung an.

Für das Libor Marktmodell betrachten wir ein  $(N-1)$ -dimensionales SDE-System, welches, wie in Kapitel 4 gezeigt, für alle  $1 \leq i \leq N-1$  durch die Gleichungen

$$dL^{(i)}(t) = L^{(i)}(t) \sigma^{(i)}(t) \left( \left( - \sum_{k=i+1}^{N-1} \frac{\delta L^{(k)}(t) \sigma^{(k)}(t) \varrho_{i,k}}{\delta L^{(k)}(t) + 1} \right) dt + dW^{(i)}(t) \right)$$

definiert ist. Dabei stellt  $(W^{(i)}(t))_t$  die  $i$ -te Komponente eines  $(N-1)$ -dimensionalen korrelierten Wiener-Prozesses  $W$  dar.

Um das Milstein Verfahren in obiger Form anzuwenden, müssen wir diese SDE noch etwas umschreiben und definieren dazu für  $1 \leq i, j \leq N-1$

$$\mu^{(i)}(L(t), t) := L^{(i)}(t)\sigma^{(i)}(t)\left(-\sum_{k=i+1}^{N-1} \frac{\delta L^{(k)}(t)\sigma^{(k)}(t)\varrho_{i,k}}{\delta L^{(k)}(t) + 1}\right)$$

und

$$\sigma^{(i,j)}(L(t), t) := \begin{cases} L^{(i)}(t)\sigma^{(i)}(t), & \text{falls } i = j \\ 0, & \text{sonst.} \end{cases}$$

Auf diese Weise können wir die Milstein Diskretisierung der dem Libor Marktmodell zugrunde liegenden SDEs als

$$\begin{aligned} L_{n+1}^{(i)} = & L_n^{(i)} + \mu^{(i)}(L_n, t_n)\Delta t + \sum_{j=1}^{N-1} \sigma^{(i,j)}(L_n, t_n)\Delta W_n^{(j)} \\ & + \sum_{j_1, j_2=1}^{N-1} h_{i, j_1, j_2}(L_n)\left(\frac{1}{2}(\Delta W_n^{(j_1)}\Delta W_n^{(j_2)} - \varrho_{j_1, j_2}\Delta t) + A_{j_1, j_2, n}\right) \quad (5.14) \end{aligned}$$

angeben, wobei  $\Delta W_n^{(j)}$  wieder je ein Wiener-Inkrement der  $j$ -ten Komponente des  $(N-1)$ -dimensionalen korrelierten Wiener-Prozesses  $W$  und  $L_n = (L_n^{(i)})_{1 \leq i \leq N-1}$  eine Approximation für den Libor-Raten-Vektor  $L(t_n) = (L^{(i)}(t_n))_{1 \leq i \leq N-1}$  darstellt.

Im Folgenden wollen wir nun noch weiter auf den im Vergleich zur Euler-Maruyama Diskretisierung im Milstein Verfahren ergänzten Term

$$\sum_{j_1, j_2=1}^{N-1} h_{i, j_1, j_2}(L_n)\left(\frac{1}{2}(\Delta W_n^{(j_1)}\Delta W_n^{(j_2)} - \varrho_{j_1, j_2}\Delta t) + A_{j_1, j_2, n}\right)$$

eingehen.

Der Ausdruck  $h_{i, j_1, j_2}(L_n)$ , den wir in Gleichung (5.9) definiert hatten, ergibt sich in diesem Modell zu

$$\begin{aligned} h_{i, j_1, j_2}(L_n) &= \sum_{k=1}^{N-1} \sigma^{(k, j_1)}(L_n, t_n) \frac{\partial \sigma^{(i, j_2)}(L_n, t_n)}{\partial L_n^{(k)}} \\ &= \begin{cases} \sigma^{(j_1)}(t_n)L_n^{(j_1)} \frac{\partial (\sigma^{(i)}(t_n)L_n^{(i)})}{\partial L_n^{(j_1)}}, & \text{falls } j_2 = i \\ 0, & \text{sonst} \end{cases} \\ &= \begin{cases} \sigma^{(i)}(t_n)L_n^{(i)}\sigma^{(i)}(t_n), & \text{falls } j_1 = j_2 = i \\ 0, & \text{sonst.} \end{cases} \end{aligned}$$

Wir stellen nun fest, dass insbesondere die Gleichung

$$h_{i,j_1,j_2}(L_n) = h_{i,j_2,j_1}(L_n) \quad \forall 1 \leq j_1, j_2 \leq N - 1$$

erfüllt ist. Da die „Lévy areas“  $A_{j_1,j_2,n}$  nach Definition (5.6) antisymmetrisch sind, was bedeutet, dass  $A_{j_1,j_2,n} = -A_{j_2,j_1,n}$  ist, erhalten wir zusammen mit der Beobachtung, dass ebenfalls nach Definition  $A_{j,j,n} = 0$  für alle  $1 \leq j \leq N - 1$  gilt,

$$h_{i,j_1,j_2}(L_n)A_{j_1,j_2,n} + h_{i,j_2,j_1}(L_n)A_{j_2,j_1,n} = 0 \quad \forall 1 \leq j_1, j_2 \leq N - 1$$

und stellen somit fest, dass die Simulation von „Lévy areas“ im Libor Marktmodell nicht notwendig ist. Wir können das Milstein Verfahren folglich in der Form

$$\begin{aligned} L_{n+1}^{(i)} &= L_n^{(i)} + \mu^{(i)}(L_n, t_n)\Delta t + \sum_{j=1}^{N-1} \sigma^{(i,j)}(L_n, t_n)\Delta W_n^{(j)} \\ &\quad + \sum_{j_1,j_2=1}^{N-1} h_{i,j_1,j_2}(L_n) \left( \frac{1}{2}(\Delta W_n^{(j_1)}\Delta W_n^{(j_2)} - \varrho_{j_1,j_2}\Delta t) \right) \\ &= L_n^{(i)} + (L_n^{(i)}\sigma^{(i)}(t_n)) \left( -\sum_{k=i+1}^{N-1} \frac{\delta L_n^{(k)}\sigma^{(k)}(t_n)\varrho_{i,k}}{\delta L_n^{(k)} + 1} \right) \Delta t + L_n^{(i)}\sigma^{(i)}(t_n)\Delta W_n^{(i)} \\ &\quad + \sigma^{(i)}(t_n)L_n^{(i)}\sigma^{(i)}(t_n) \left( \frac{1}{2}((\Delta W_n^{(i)})^2 - \Delta t) \right) \end{aligned} \tag{5.15}$$

angeben.

Die Varianz der Milstein-Approximation erreicht in diesem Fall also auch ohne eine Simulation von „Lévy areas“ eine Konvergenzordnung  $\mathcal{O}(\Delta t^2)$ , da das Milstein Verfahren nach [KP92] eine starke Konvergenzrate von 1 besitzt. Analog zur Herleitung der Gleichungen (3.5) und (3.6) erhalten wir für eine Schrittweite  $\Delta t = h_l$ , dass  $V_l = \mathbb{V}[\tilde{P}_l - \tilde{P}_{l-1}]$  von der Ordnung  $\mathcal{O}(h_l^2)$  ist.

Wir haben nun also gesehen, wie wir das dem Libor Marktmodell zugrunde liegende SDE-System sowohl mithilfe des Euler-Maruyama Verfahrens als auch mithilfe des Milstein Verfahrens diskretisieren können. Für die Anwendung des Multilevel Monte Carlo Verfahrens fehlen uns jetzt folglich nur noch geeignete Zinsderivate, welche wir bewerten wollen. Diese werden wir im nachfolgenden Kapitel vorstellen.

## 6 Zinsderivate

Nachdem wir die notwendigen theoretischen Grundlagen zur Multilevel Monte Carlo Bewertung im Libor Marktmodell geschaffen haben, werden wir in diesem Kapitel zunächst grundlegende Zinsderivate einführen, um anschließend die (exotischen) Derivate, die CMS Caps und TARNs, vorzustellen, welche wir in dieser Arbeit bewerten wollen. Dazu werden wir uns an der Darstellung in [AP10] orientieren.

### 6.1 Grundlegende Zinsderivate

#### 6.1.1 Swaps und Swaptions

Bei einem Swap einigen sich zwei Parteien darauf, eine Folge von Cash Flows gegeneinander auszutauschen. Ein Plain Vanilla „fixed-for-floating“ Zinsswap – im Folgenden einfach Swap genannt – ist dann ein Tauschgeschäft, bei dem eine Folge von festen Zinszahlungen gegen eine Folge von variablen Zinszahlungen, welche auf einer sogenannten „floating rate“, meistens einem Libor-Zinssatz, beruhen, getauscht wird. Die beiden Zahlungsströme besitzen dabei dieselben Eigenschaften in Bezug auf Währung, Nominal und Fälligkeitsdatum und werden bezüglich einer festgelegten Abfolge von Zahlungszeitpunkten geleistet. Dabei wird die variable Zinsrate typischerweise zu Beginn eines jeden Tenorintervalls beobachtet und festgesetzt und gleichzeitig mit der festen Zinszahlung dann am Ende der Periode ausgezahlt. Ein Plain Vanilla Swap ist also vergleichbar mit einer Folge von Forward Rate Agreements (FRAs).

In der Praxis werden Swaps mit unterschiedlichen Laufzeiten meist zwischen Händlern von Zinsraten und Finanzinstituten gehandelt, um Zinsänderungsrisiken der beiden Parteien anzupassen oder abzutreten. Weiterhin werden Swaps oft von Firmen genutzt, um feste Zinszahlungsverpflichtungen, die meist im Zusammenhang mit der Ausgabe von Bonds auftreten, gegen variable – oder umgekehrt – zu tauschen.

Um einen „fixed-for-floating“ Zinsswap auch formal zu definieren, müssen wir eine Tenorstruktur, d.h. eine aufsteigende Folge von Fälligkeitszeitpunkten

$$T_0 < T_1 < \dots < T_N,$$

festlegen. An dieser Stelle sei bemerkt, dass wir in der weiteren Arbeit wieder von konstanten Tenorintervallängen  $\delta$  ausgehen werden. Die Definitionen aller weiteren Zinsderivate lassen sich aber natürlich auch auf eine Situation mit variablen Tenorintervallängen übertragen.

Bei einem Swap mit fester Zinsrate  $K$  werden die variablen Libor-Raten dann zu Beginn eines jeden Intervalls, also jeweils in  $T_n$ , fixiert und die Zahlungen am Ende der jeweiligen Periode, also in  $T_{n+1}$ , ausgetauscht. In der Praxis wird dabei natürlich je nur die Differenz der beiden Zahlungsströme bezahlt.

Im Folgenden betrachten wir einen Swap, dessen Laufzeit im Tenorzeitpunkt  $T_a$  beginnt und im Tenorzeitpunkt  $T_b$  endet. Dies bedeutet, dass in  $T_a$  erstmalig ein sogenanntes „Fixing“ stattfindet und in  $T_b$  die letzte Zahlung geleistet wird. Natürlich sollte dazu  $0 \leq a < b \leq N$  gelten. Weiterhin wollen wir hier und in der weiteren Arbeit der Einfachheit halber davon ausgehen, dass das Nominal den Wert 1 hat, wenn wir dieses nicht explizit angeben.

Die Zahlung des sogenannten Payers, also der Partei, die die festen Zinsraten zahlt und die flexiblen erhält, zum Zeitpunkt  $T_{n+1}$  mit  $a \leq n \leq b-1$  ergibt sich dann zu

$$\delta(L^{(n)}(T_n) - K).$$

Für den Wert eines solchen Swaps  $V_{\text{Swap}}^{a,b}(t)$  zum Zeitpunkt  $t \leq T_a$  erhalten wir

$$V_{\text{Swap}}^{a,b}(t) = \sum_{n=a}^{b-1} \beta(t) \delta \mathbb{E}[\beta(T_{n+1})^{-1} (L^{(n)}(T_n) - K) | \mathcal{F}_t],$$

wobei  $\beta(t)$  den Wert des Geldmarktkontos zum Zeitpunkt  $t$  bezeichne, welcher zum Diskontieren der Zahlungen verwendet werden kann, und der Erwartungswert bzgl. des risikolosen äquivalenten Martingalmaßes  $\mathbb{P}$  genommen wird.

Im Folgenden bezeichne  $\mathbb{E}^{\mathbb{P}_{T_{n+1}}}$  den Erwartungswert bzgl. des Forwardmartingalmaßes  $\mathbb{P}_{T_{n+1}}$ , welches ein äquivalentes Martingalmaß bzgl. des Numeraire Assets  $(B(t, T_{n+1}))_{0 \leq t \leq T_{n+1}}$  darstellt.

Für den Wert des Swaps erhalten wir mit dieser Bezeichnung

$$\begin{aligned} V_{\text{Swap}}^{a,b}(t) &= \sum_{n=a}^{b-1} \beta(t) \delta \mathbb{E}^{\mathbb{P}_{T_{n+1}}} \left[ \frac{\beta(T_{n+1})}{B(T_{n+1}, T_{n+1})} \frac{B(t, T_{n+1})}{\beta(t)} \beta(T_{n+1})^{-1} (L^{(n)}(T_n) - K) \mid \mathcal{F}_t \right] \\ &= \sum_{n=a}^{b-1} \delta B(t, T_{n+1}) \mathbb{E}^{\mathbb{P}_{T_{n+1}}} [(L^{(n)}(T_n) - K) | \mathcal{F}_t] \\ &= \sum_{n=a}^{b-1} \delta B(t, T_{n+1}) (L^{(n)}(t) - K), \end{aligned}$$

da  $(L^{(n)}(t))_t$  nach Definition ein  $\mathbb{P}_{T_{n+1}}$ -Martingal ist und die Maßwechseldichte  $(Z_t)_t$  durch

$$Z_t := \frac{d\mathbb{P}}{d\mathbb{P}_{T_{n+1}}} \Big|_{\mathcal{F}_t} = \frac{\beta(t)}{B(t, T_{n+1})} B(0, T_{n+1}), \quad 0 \leq n \leq N-1,$$

gegeben ist. Den ersten Gleichungsschritt erhalten wir dann also mithilfe dieser Maßwechseldichte und der „Bayes-Regel für bedingte Erwartungswerte“, welche wir in Anlehnung an [Shr04] im folgenden Lemma angeben.

**Lemma 6.1** (Bayes-Regel für bedingte Erwartungswerte)

Sei  $(Z_t)_t$  ein Radon-Nikodym Dichteprozess, d.h. es sei  $Z := \frac{d\tilde{\mathbb{P}}}{d\mathbb{P}}$  und  $Z_t := \mathbb{E}[Z | \mathcal{F}_t]$ .

Dann gilt für eine  $\mathcal{F}_t$ -messbare Zufallsvariable  $X$  und  $0 \leq s \leq t \leq T$

$$\tilde{\mathbb{E}}[X|\mathcal{F}_s] = \frac{1}{Z_s} \mathbb{E}[XZ_t|\mathcal{F}_s].$$

Setzen wir nun die Definition der Libor-Raten in die oben hergeleitete Formel ein, so erhalten wir eine alternative Darstellung des Wertes eines „fixed-for-floating“ Swaps als

$$\begin{aligned} V_{\text{Swap}}^{a,b}(t) &= \sum_{n=a}^{b-1} \delta B(t, T_{n+1}) \left( \frac{1}{\delta} \left( \frac{B(t, T_n)}{B(t, T_{n+1})} - 1 \right) - K \right) \\ &= \sum_{n=a}^{b-1} \delta B(t, T_{n+1}) \left( \frac{1}{\delta} \frac{B(t, T_n) - B(t, T_{n+1})}{B(t, T_{n+1})} - K \right) \\ &= \sum_{n=a}^{b-1} B(t, T_n) - B(t, T_{n+1}) - K \sum_{n=a}^{b-1} \delta B(t, T_{n+1}) \\ &= \underbrace{\left( \frac{B(t, T_a) - B(t, T_b)}{\sum_{n=a}^{b-1} \delta B(t, T_{n+1})} - K \right)}_{=: R_{\text{Swap}}^{a,b}(t)} \sum_{n=a}^{b-1} \delta B(t, T_{n+1}) \\ &= (R_{\text{Swap}}^{a,b}(t) - K) \sum_{n=a}^{b-1} \delta B(t, T_{n+1}). \end{aligned} \quad (6.1)$$

Die feste Zinsrate  $K$ , die den Swap „fair“ macht, wird dann dementsprechend so gewählt, dass der Wert des Swaps zum (aktuellen) Zeitpunkt  $t$  null ist. Das so gewählte  $K = R_{\text{Swap}}^{a,b}(t)$  nennt man deshalb auch „break-even“ Swap-Rate.

Neben diesen Plain Vanilla Swaps können wir natürlich auch exotische Swaps betrachten. Dazu kann in obiger Darstellung statt der festen Zinsrate auch für dieses sogenannte „leg“ eine variable Zinsrate verwendet werden. Wir machen diese Zinsrate, welche dabei weiterhin gegen eine Libor-Rate getauscht wird, folglich ebenso abhängig von einer Libor-Rate, d.h. wir tauschen hier die Zinsrate  $K_n(L^{(n)}(T_n))$  gegen eine Libor-Rate  $L^{(n)}(T_n)$ . In der Literatur wird  $K_n$  dabei meist als „structured coupon“, also strukturierter Kupon, bezeichnet. Der Wert eines exotischen Swaps ergibt sich dann zu

$$V_{\text{exot. Swap}}^{a,b}(t) = \sum_{n=a}^{b-1} \beta(t) \delta \mathbb{E}[\beta(T_{n+1})^{-1} (L^{(n)}(T_n) - K_n(L^{(n)}(T_n))) | \mathcal{F}_t].$$

Mithilfe der Kenntnis von Swaps können wir nun auch Swaptions definieren. Eine Swaption stellt die Option dar, zum Zeitpunkt  $T_a > T_0$  in einen Swap mit Laufzeitende in  $T_b$  und Festzinssatz  $K$  einzutreten. Die Option wird natürlich genau dann ausgeübt, wenn der Wert des Swaps in  $T_a$  positiv ist, wenn also  $V_{\text{Swap}}^{a,b}(T_a) > 0$  gilt. Die Auszahlung einer solchen Swaption in  $T_a$  ergibt sich mithilfe von (6.1)

dann folglich zu

$$[R_{\text{Swap}}^{a,b}(T_a) - K]^+ \sum_{n=a}^{b-1} \delta B(T_a, T_{n+1}). \quad (6.2)$$

### 6.1.2 Caps und Floors

Ein Zinscaplet ist ein Wertpapier, welches dem Inhaber das Recht einräumt, eine variable Zinsrate, meistens eine Libor-Rate, am Ende eines Tenorintervalls  $[T_n, T_{n+1}]$  gegen eine feste Zinsrate zu tauschen, wobei  $0 \leq n \leq N-1$  gelten sollte. Bei einem Caplet handelt es sich im Prinzip also um eine Call-Option auf eine Libor-Rate. Dazu wird die variable Zinsrate wie bei den oben vorgestellten Swaps zu Beginn des betrachteten Tenorintervalls beobachtet und festgehalten. Die Auszahlung ergibt sich zum Zeitpunkt  $T_{n+1}$  damit zu

$$\delta(L^{(n)}(T_n) - K)^+$$

pro Einheit des Nominals.

Ein Zinsfloorlet ist analog zum Caplet das Recht, eine feste Zinsrate gegen eine variable Zinsrate zu tauschen, also prinzipiell eine Put-Option auf eine Libor-Rate. Die Auszahlung in  $T_{n+1}$  ergibt sich demnach zu

$$\delta(K - L^{(n)}(T_n))^+$$

pro Einheit des Nominals.

Ein Cap bzw. ein Floor ist nun eine rollierende Folge von Caplets bzw. von Floorlets, welche in  $T_a$  startet und in  $T_b$  endet.

Den Wert eines Caps bzw. eines Floors zum Zeitpunkt  $t > 0$  erhalten wir dann als

$$V_{\text{Cap}}^{a,b}(t) = \beta(t) \sum_{n=a}^{b-1} \delta \mathbb{E}[\beta(T_{n+1})^{-1} (L^{(n)}(T_n) - K)^+ | \mathcal{F}_t]$$

bzw.

$$V_{\text{Floor}}^{a,b}(t) = \beta(t) \sum_{n=a}^{b-1} \delta \mathbb{E}[\beta(T_{n+1})^{-1} (K - L^{(n)}(T_n))^+ | \mathcal{F}_t].$$

Führen wir für jeden Summanden analog zum Vorgehen bei der Bewertung von Swaps je einen Maßwechsel zum Forwardmartingalmaß  $\mathbb{P}_{T_b}$  durch, so ergibt sich eine alternative Darstellung als

$$\begin{aligned} V_{\text{Cap}}^{a,b}(t) &= \beta(t) \sum_{n=a}^{b-1} \delta \mathbb{E}^{\mathbb{P}_{T_b}} \left[ \frac{\beta(T_{n+1})}{B(T_{n+1}, T_b)} \frac{B(t, T_b)}{\beta(t)} \beta(T_{n+1})^{-1} (L^{(n)}(T_n) - K)^+ | \mathcal{F}_t \right] \\ &= \sum_{n=a}^{b-1} \delta B(t, T_b) \mathbb{E}^{\mathbb{P}_{T_b}} [B(T_{n+1}, T_b)^{-1} (L^{(n)}(T_n) - K)^+ | \mathcal{F}_t] \end{aligned} \quad (6.3)$$

bzw.

$$V_{\text{Floor}}^{a,b}(t) = \sum_{n=a}^{b-1} \delta B(t, T_b) \mathbb{E}^{\mathbb{P}_{T_b}} [B(T_{n+1}, T_b)^{-1} (K - L^{(n)}(T_n))^+ | \mathcal{F}_t].$$

Alternativ könnten wir selbstverständlich auch hier – wie es in der Literatur meist zu finden ist – statt bei allen Summanden zum selben Maß  $\mathbb{P}_{T_b}$  zu wechseln, für jeden einzelnen Summanden einen Maßwechsel zum jeweiligen Forwardmartingalmaß durchführen. Dies würde dafür sorgen, dass der Bondpreis  $B(T_{n+1}, T_b)$  im Erwartungswert für alle  $n$  verschwindet. Da wir in dieser Arbeit aber nicht daran interessiert sind, analytische Formeln für die Bewertung von Zinsderivaten herzuleiten, was bei komplexeren Derivaten meist auch nicht möglich ist, sondern die Derivate mithilfe von Simulationen bewerten wollen, ist es sinnvoll, die Maßwechsel je zum selben Forwardmartingalmaß durchzuführen. Dadurch müssen wir Wiener-Prozesse bzw. Inkremente von Wiener-Prozessen nur für ein Wahrscheinlichkeitsmaß simulieren, was den Simulationsaufwand deutlich verringert. Auf diese Weise müssen wir den Bondpreis im Erwartungswert allerdings mithilfe der simulierten Libor-Raten-Pfade ausdrücken. Wir werden in der Simulation also den Ausdruck

$$\begin{aligned} B(T_{n+1}, T_b) &= B(T_{n+1}, T_{n+1}) \prod_{j=n+1}^{b-1} \frac{B(T_{n+1}, T_{j+1})}{B(T_{n+1}, T_j)} \\ &= \prod_{j=n+1}^{b-1} \frac{1}{\delta L^{(j)}(T_{n+1}) + 1}, \end{aligned} \tag{6.4}$$

wobei wir im letzten Schritt die Definition (4.1) der Libor-Raten genutzt haben, als Darstellung für die Bonds verwenden.

## 6.2 CMS Caps und Floors

Da der Markt an Plain Vanilla Swaps sehr groß ist, werden Marktwerte von zugehörigen Swap-Raten oft als eine Art Marktvariable verwendet. Diese werden dann genutzt, um Payoffs weiterer Derivate zu definieren.

Eine „Constant-Maturity Swap“-Rate, kurz CMS-Rate, ist dann als „break-even“ Swap-Rate auf einen Plain Vanilla Swap mit fester Laufzeit, z.B. 10 oder 30 Jahre, definiert. Davon ausgehend können wir nun CMS Caps bzw. CMS Floors definieren. Bei einem CMS Cap wird dann beispielsweise eine Option auf den Tausch von solchen CMS-Raten gegen feste Zinsraten gewährt. Dies bedeutet, dass der Unterschied eines Plain Vanilla Caps und eines CMS Caps darin liegt, dass bei einem CMS Cap für das „floating leg“, also die variable Komponente des Caps, statt der Libor-Rate  $L^{(n)}(T_n)$  die Swap-Rate für einen Swap mit langer Laufzeit, sagen wir  $m$  Jahre, verwendet wird. Ein Grund, warum solche CMS Produkte in großer Zahl gehandelt werden, ist, dass mithilfe von CMS Produkten auf den langfristigen Verlauf der Zinsstrukturkurve statt nur auf Zinsraten für kurze Laufzeiten – wie die Libor-Raten bei Plain Vanilla Caps – gesetzt werden kann und trotzdem nur eine relativ kurze Laufzeit des Produktes vorliegt. Dies ist beispielsweise der

Fall, wenn  $a$  und  $b$  dicht beieinander liegen und ein großer Wert für  $m$  gewählt wird.

Der Wert eines CMS Caps mit Beginn in  $T_a$  und Laufzeitende in  $T_b$  ergibt sich dann folglich zu

$$V_{\text{CMSCap}}^{a,b}(t) = \beta(t) \sum_{n=a}^{b-1} \delta \mathbb{E}[\beta(T_{n+1})^{-1} (R_{\text{Swap}}^{n,n+m}(T_n) - K)^+ | \mathcal{F}_t]$$

bzw. mit analoger Herleitung wie bei den Caps zu

$$V_{\text{CMSCap}}^{a,b}(t) = \sum_{n=a}^{b-1} \delta B(t, T_b) \mathbb{E}^{\mathbb{P}_{T_b}}[B(T_{n+1}, T_b)^{-1} (R_{\text{Swap}}^{n,n+m}(T_n) - K)^+ | \mathcal{F}_t]. \quad (6.5)$$

Analog dazu ergibt sich der Wert eines CMS Floors zu

$$V_{\text{CMSFloor}}^{a,b}(t) = \beta(t) \sum_{n=a}^{b-1} \delta \mathbb{E}[\beta(T_{n+1})^{-1} (K - R_{\text{Swap}}^{n,n+m}(T_n))^+ | \mathcal{F}_t]$$

bzw.

$$V_{\text{CMSFloor}}^{a,b}(t) = \sum_{n=a}^{b-1} \delta B(t, T_b) \mathbb{E}^{\mathbb{P}_{T_b}}[B(T_{n+1}, T_b)^{-1} (K - R_{\text{Swap}}^{n,n+m}(T_n))^+ | \mathcal{F}_t].$$

Für die Bewertung über Monte Carlo Simulationen, bei denen wir Libor-Raten-Pfade simulieren, können wir die Swap-Rate  $R_{\text{Swap}}^{n,n+m}(T_n)$  mithilfe der Darstellung (6.4) der Bondpreise umschreiben zu

$$\begin{aligned} R_{\text{Swap}}^{n,n+m}(T_n) &= \frac{B(T_n, T_n) - B(T_n, T_{n+m})}{\sum_{j=n}^{n+m-1} \delta B(T_n, T_{j+1})} \\ &= \frac{1 - \prod_{j=n}^{n+m-1} \frac{1}{\delta L^{(j)}(T_n) + 1}}{\sum_{j=n}^{n+m-1} \delta \prod_{i=n}^j \frac{1}{\delta L^{(i)}(T_n) + 1}} \end{aligned} \quad (6.6)$$

und erhalten somit eine Schreibweise der Swap-Raten, die nur noch von den Libor-Raten abhängt, sodass die Swap-Raten dann direkt über die simulierten Libor-Raten bestimmt werden können.

Alternativ wäre es auch möglich, die Swap-Raten direkt zu simulieren. Dazu müsste man dann allerdings entweder ein neues Modell kalibrieren, um die Parameter für ein sogenanntes Swap Marktmodell zu bestimmen, oder die Dynamiken der Swap-Raten in einem Libor Marktmodell herleiten. Dabei muss man aber feststellen, dass auch in diesen Dynamiken die Libor-Raten explizit vorkommen, weshalb diese also in jedem Fall simuliert werden müssen (siehe [BM06], S.227f und S.385). Aus diesem Grund ist es sinnvoll, obige Darstellung (6.6) der Swap-Raten zu nutzen.

### 6.3 TARNs

Bei „Target(ed) Redemption Notes“ (TARNs), welche teilweise auch unter der Bezeichnung „Targeted Accrual Redemption Notes“ zu finden sind, erhält der Investor für die Investition eines Nominals zu festgelegten (Tenor-) Zeitpunkten  $(T_n)_n$  gewisse strukturierte Kuponzahlungen  $(K_n)_n$ . Die gesamten Kuponzahlungen werden gleichzeitig gespeichert und aufsummiert. Sobald die Summe der einzelnen Kuponzahlungen einen vorher festgelegten (Jahres-)Zielkupon  $R$  erreicht bzw. überschreitet, wird der Vertrag vorzeitig beendet und das Nominal wird vor dem eigentlichen Laufzeitende  $T_b$  zurückgezahlt. Darüber hinaus gibt es verschiedene Gestaltungsmodalitäten. Wir werden hier die häufig vorzufindende Variante betrachten, bei der, falls die Schranke bis zum Laufzeitende  $T_b$  nicht erreicht wird, der Zielkupon am Ende der Laufzeit erzwungen wird, indem die noch fehlende Zinszahlung durchgeführt wird. Dies bedeutet, dass Investoren eine sichere, vorher festgelegte Kuponzahlung erhalten, welche spätestens zum Laufzeitende vollständig getätigt wird. In der Literatur nennt man diese Eigenschaft „lifetime floor“ (vgl. [AP10]). In Bezug auf die letzte Kuponzahlung bei einem vorzeitigen Erreichen der Schranke legen wir fest, dass nur der Differenzkupon bis zum Erreichen des Zielkupons gezahlt wird und nicht der komplette, den Zielkupon in der Summe überschreitende Kupon für diesen Zeitpunkt. Dieses Merkmal wird in der Literatur meist „cap at trigger“ genannt.

Der Einfachheit halber nehmen wir an, dass das Nominal weiterhin 1 betrage. Für ein Investment der Höhe 1 in  $T_0$  erhalten wir dann eine Auszahlung eines TARNs zum Zeitpunkt  $T_n$  als

$$\mathbf{1}_{\{\sum_{i=1}^{n-1} K_i < R\}} (\delta \min\{K_n, R - \sum_{i=1}^{n-1} K_i\} + \mathbf{1}_{\{R \leq \sum_{i=1}^n K_i\}}), \text{ für } 1 \leq n \leq b-1$$

bzw.

$$\mathbf{1}_{\{\sum_{i=1}^{b-1} K_i < R\}} (\delta(R - \sum_{i=1}^{b-1} K_i) + 1), \text{ für } n = b.$$

Typischerweise wird den Investoren bei einem TARN in der ersten Periode ein vorher festgelegter (oft recht hoher) Kupon gezahlt, welcher den TARN für Investoren sehr attraktiv macht. Die obige Darstellung der Auszahlungen des TARNs ändert dies jedoch nicht, da nur die erste Auszahlung auf einen besonderen Kupon  $K_1$  abgeändert wird. Bei der Bewertung können wir dies unbeachtet lassen, da auch im Fall ohne eine besondere Kuponzahlung in  $T_1$  der Kupon  $K_1$  in  $T_0$  festgelegt wird und somit ohnehin  $\mathcal{F}_{T_0}$ -messbar ist.

Mit den obigen Auszahlungen des TARN-Produktes erhalten wir dessen Wert als

$$\begin{aligned}
 V_{\text{TARN}}^b(t) = & -1 + \beta(t) \mathbb{E} \left[ \left( \sum_{n=0}^{b-2} \beta(T_{n+1})^{-1} \mathbf{1}_{\{\sum_{i=1}^n K_i < R\}} \right. \right. \\
 & (\delta \min\{K_{n+1}, R - \sum_{i=1}^n K_i\} + \mathbf{1}_{\{R \leq \sum_{i=1}^{n+1} K_i\}})) \\
 & \left. \left. + \beta(T_b)^{-1} \mathbf{1}_{\{\sum_{i=1}^{b-1} K_i < R\}} (\delta(R - \sum_{i=1}^{b-1} K_i) + 1) \mid \mathcal{F}_t \right) \right].
 \end{aligned}$$

Auch für diesen Erwartungswert führen wir wieder einen Maßwechsel zum Forwardmartingalmaß  $\mathbb{P}_{T_b}$  durch und erhalten

$$\begin{aligned}
 V_{\text{TARN}}^b(t) = & -1 + B(t, T_b) \mathbb{E}^{\mathbb{P}_{T_b}} \left[ \left( \sum_{n=0}^{b-2} B(T_{n+1}, T_b)^{-1} \mathbf{1}_{\{\sum_{i=1}^n K_i < R\}} \right. \right. \\
 & (\delta \min\{K_{n+1}, R - \sum_{i=1}^n K_i\} + \mathbf{1}_{\{R \leq \sum_{i=1}^{n+1} K_i\}})) \\
 & \left. \left. + \mathbf{1}_{\{\sum_{i=1}^{b-1} K_i < R\}} (\delta(R - \sum_{i=1}^{b-1} K_i) + 1) \mid \mathcal{F}_t \right) \right].
 \end{aligned}$$

Im Regelfall behalten die Emittenten diese komplexen Finanzinstrumente nicht in ihren Büchern (siehe [AP10]) und reinvestieren die Nominalzahlung zu Libor-Zinssätzen. Dies bedeutet, dass ein TARN so im Prinzip einen exotischen Swap darstellt, welcher vorzeitig beendet wird, falls die Summe der Kuponzahlungen die festgelegte Grenze erreicht. Für einen Investor erhalten wir zum Zeitpunkt  $T_n$  auf diese Weise eine Auszahlung von

$$\mathbf{1}_{\{\sum_{i=1}^{n-1} K_i < R\}} \delta(\min\{K_n, R - \sum_{i=1}^{n-1} K_i\} - L^{(n-1)}(T_{n-1})), \quad \text{für } 1 \leq n \leq b-1$$

bzw.

$$\mathbf{1}_{\{\sum_{i=1}^{b-1} K_i < R\}} \delta(R - \sum_{i=1}^{b-1} K_i - L^{(b-1)}(T_{b-1})), \quad \text{für } n = b.$$

Der Wert des TARN-Produktes zum Zeitpunkt  $t$  ergibt sich in diesem Fall zu

$$\begin{aligned}
 V_{\text{TARN}}^{\text{Swap}, b}(t) = & \beta(t) \mathbb{E} \left[ \left( \sum_{n=0}^{b-2} \beta(T_{n+1})^{-1} \mathbf{1}_{\{\sum_{i=1}^n K_i < R\}} \right. \right. \\
 & \delta(\min\{K_{n+1}, R - \sum_{i=1}^n K_i\} - L^{(n)}(T_n))) \\
 & \left. \left. + \beta(T_b)^{-1} \mathbf{1}_{\{\sum_{i=1}^{b-1} K_i < R\}} \delta((R - \sum_{i=1}^{b-1} K_i) - L^{(b-1)}(T_{b-1})) \mid \mathcal{F}_t \right) \right]
 \end{aligned}$$

bzw. nach einem Maßwechsel und mithilfe der Darstellung (6.4) der Bonds zu

$$\begin{aligned}
 V_{\text{TARN}_{\text{Swap}}}^b(t) &= B(t, T_b) \mathbb{E}^{\mathbb{P}_{T_b}} \left[ \left( \sum_{n=0}^{b-2} B(T_{n+1}, T_b)^{-1} \mathbf{1}_{\{\sum_{i=1}^n K_i < R\}} \right. \right. \\
 &\quad \delta(\min\{K_{n+1}, R - \sum_{i=1}^n K_i\} - L^{(n)}(T_n))) \\
 &\quad \left. \left. + \mathbf{1}_{\{\sum_{i=1}^{b-1} K_i < R\}} \delta((R - \sum_{i=1}^{b-1} K_i) - L^{(b-1)}(T_{b-1})) \mid \mathcal{F}_t \right] \right] \\
 &= B(t, T_b) \mathbb{E}^{\mathbb{P}_{T_b}} \left[ \left( \sum_{n=0}^{b-2} \left( \prod_{j=n+1}^{b-1} L^{(j)}(T_{n+1}) \delta + 1 \right) \mathbf{1}_{\{\sum_{i=1}^n K_i < R\}} \right. \right. \\
 &\quad \delta(\min\{K_{n+1}, R - \sum_{i=1}^n K_i\} - L^{(n)}(T_n))) \\
 &\quad \left. \left. + \mathbf{1}_{\{\sum_{i=1}^{b-1} K_i < R\}} \delta((R - \sum_{i=1}^{b-1} K_i) - L^{(b-1)}(T_{b-1})) \mid \mathcal{F}_t \right] \right].
 \end{aligned}$$

### Bemerkung 6.2

Mit Hinblick auf die Anwendung des zentralen Satzes 3.4 sollte hier bemerkt werden, dass dieser in der dort angegeben Form für ein TARN-Produkt wegen der unstetigen Abbruchbedingungen zumindest in der Version als „Note“ nicht mehr angewendet werden kann. Für die allgemeine Betrachtung solcher unstetigen Payoff-Funktionen verweisen wir auf eine Untersuchung in [GHM09]. In der Version als „Swap“ ist die Payoff-Funktion eines TARNs weiterhin stetig, weshalb wir im Folgenden immer von dieser Version ausgehen wollen, wenn wir von TARNs sprechen. Aus diesem Grund werden wir in Zukunft auf die Kennzeichnung, dass es sich um die „Swap-Variante“ handelt verzichten und die Bezeichnung  $V_{\text{TARN}}^b$  im Folgenden für den Wert eines TARNs in der „Swap-Variante“ nutzen.

Zuletzt wollen wir noch auf die strukturierten Kupons  $(K_n)_n$  innerhalb der TARN-Produkte eingehen. Für diese Kupons können wir prinzipiell beliebige strukturierte Zahlungsströme verwenden. In der „klassischen“ Variante wird hier ein „inverse floating coupon“ verwendet. Dieser besitzt für  $0 \leq n \leq b-1$  in  $T_{n+1}$  die Auszahlung

$$K_{n+1} = (K - gL^{(n)}(T_n))^+,$$

wobei  $K$  ein vorher festgelegter Strike-Zinssatz sei und  $g$  einen sogenannten „gearing“-Faktor, also einen Hebel, der den Libor-Zinssatz vervielfachen kann, darstelle. In der Praxis wird dabei oft ein Faktor  $g = 2$  genutzt. Weiterhin kann man einen „inverse floater“, wie dieser Kupon auch genannt wird, in einer „capped“ oder „floored“ Variante vorfinden. Im ersten Fall sorgt dies dafür, dass ein maximaler Kupon nicht überschritten wird, im anderen Fall dafür, dass ein minimaler Kupon nicht unterschritten wird.

---

## 7 Bewertung von Zinsderivaten im Libor Marktmodell

Nachdem wir im vorherigen Kapitel die notwendigen Grundlagen über Zinsderivate geschaffen haben, wollen wir nun untersuchen, wie sich diese Zinsderivate mithilfe des Multilevel Monte Carlo Verfahrens im Libor Marktmodell bewerten lassen. Dazu wollen wir im kommenden Abschnitt zunächst untersuchen, inwieviel wir den Satz 3.4 in dieser speziellen Situation verwenden können.

### 7.1 Anwendung des zentralen Satzes

Im zentralen Satz 3.4 über die Funktionsfähigkeit des Multilevel Monte Carlo Verfahrens wird als erstes eine schwache Konvergenzrate  $\alpha \geq \frac{1}{2}$  gefordert. Aus der Literatur (siehe z.B. [KP92]) ist bekannt, dass sowohl das Euler-Maruyama Verfahren als auch das Milstein Verfahren eine schwache Konvergenzrate von 1 besitzt, wenn die Drift- und Volatilitätsterme der zugrunde liegenden SDE sowie die Payoff-Funktion  $f$  beispielsweise die Bedingungen aus Lemma (3.1) erfüllen. Dass die Koeffizientenfunktionen

$$L^{(i)}(t)\sigma^{(i)}(t)\left(-\sum_{k=i+1}^{N-1} \frac{\delta L^{(k)}(t)\sigma^{(k)}(t)\varrho_{i,k}}{\delta L^{(k)}(t)+1}\right) \text{ und } L^{(i)}(t)\sigma^{(i)}(t)$$

der dem Libor Marktmodell zugrunde liegenden SDE (4.3) Lipschitz-stetig sind, haben wir schon in Bemerkung 4.2 und den Anmerkungen zum Existenz- und Eindeutigkeitssatz einer Lösung einer SDE (Satz 4.3) gesehen. Dass die beiden Koeffizientenfunktionen darüber hinaus ausreichend glatt sind, kann ebenfalls leicht gezeigt werden, wenn wir zusätzlich zur Beschränktheit der deterministischen Volatilitäten  $\sigma^{(i)}(t)$  deren Differenzierbarkeit voraussetzen, was sich bei der im Abschnitt 8.1 vorgestellten Wahl dieser Volatilitätsfunktionen ergeben wird.

Problematischer wird es mit der Glattheit der Payoff-Funktionen der Zinsderivate. Betrachten wir dazu beispielsweise die einfachste Payoff-Funktion der betrachteten Zinsderivate, nämlich die Payoff-Funktion eines Caplets

$$\delta(L^{(n)}(T_n) - K)^+$$

für ein beliebiges  $1 \leq n \leq N-1$ , so sehen wir, dass diese schon nicht differenzierbar in  $L^{(n)}(T_n) = K$  ist.

Im Folgenden wollen wir aber dennoch davon ausgehen, dass alle hier vorgestellten Payoff-Funktionen die Glattheitsbedingungen des Lemmas 3.1 erfüllen. Dies ist damit zu rechtfertigen, dass wir die stetigen Payoff-Funktionen an der bzw. den entsprechenden Stellen, an denen sie nicht differenzierbar sind, „glätten“ können, ohne dabei den Wert des Derivats entscheidend zu verändern. Liegt nämlich die geglättete Funktion ausreichend nah an der ursprünglichen Funktion, so ist der Unterschied, welcher sich bei der Bewertung mithilfe von Monte Carlo Methoden ergibt, nur minimal klein, sodass er bei der in dieser Arbeit geforderten Fehlertoleranz keine Auswirkungen hat.

Damit können wir also davon ausgehen, dass wir die erste Voraussetzung (i) des

zentralen Satzes mit  $\alpha = 1$  erfüllen können.

Die Voraussetzungen (ii) und (iv) zu erfüllen, ist bei der Konstruktion des Multilevel Monte Carlo Schätzers, wie wir es in dieser Arbeit durchgeführt haben, ebenfalls unkompliziert (siehe [Gil08a]) bzw. folgt deren Gültigkeit sogar unmittelbar aus unseren Definitionen der Schätzer  $(\hat{Y}_l)_{0 \leq l \leq L}$  und des Rechenaufwands.

Kommen wir nun zur dritten Voraussetzung (iii) des Satzes, welche fordert, dass

$$\mathbb{V}[\hat{Y}_l] \leq c_2 N_l^{-1} h_l^\beta$$

für Konstanten  $c_2, \beta > 0$  gilt. In der Ungleichung (3.6) aus Abschnitt 3.2 haben wir gesehen, dass bei der Verwendung einer Lipschitz-stetigen Payoff-Funktion und des Euler-Maruyama Diskretisierungsverfahrens die Varianz  $V_l$  von der Ordnung  $\mathcal{O}(h_l)$  ist, was bedeutet, dass der Parameter  $\beta$  den Wert 1 erreicht.

Unter Verwendung des Milstein Diskretisierungsverfahrens erhalten wir eine starke Konvergenzrate von 1, was dazu führt, dass die Varianz  $V_l$  die Ordnung  $\mathcal{O}(h_l^2)$  besitzt, wie wir es am Ende des Kapitels 5.3 beschrieben haben. Wir können in diesem Fall den Parameter  $\beta$  also sogar als 2 wählen, was eine deutliche Verbesserung des Verfahrens mit sich bringt.

Betrachten wir für ein beliebiges  $1 \leq n \leq N-1$  nun noch einmal die Payoff-Funktion eines Caplets

$$\delta(L^{(n)}(T_n) - K)^+.$$

Wir können sehen, dass diese Funktion tatsächlich Lipschitz-stetig ist, denn es gilt

$$\begin{aligned} |\delta(x - K)^+ - \delta(y - K)^+| \\ &= \delta|(x - K)^+ - (y - K)^+| \\ &= \delta[\mathbf{1}_{\{y < K < x\}}(x - K) + \mathbf{1}_{\{x < K < y\}}(y - K) + \mathbf{1}_{\{x, y \geq K\}}|x - y|] \\ &\leq \delta[\mathbf{1}_{\{y < K < x\}}(x - y) + \mathbf{1}_{\{x < K < y\}}(y - x) + \mathbf{1}_{\{x, y \geq K\}}|x - y|] \\ &= \delta|x - y|\mathbf{1}_{\{x > K \vee y > K\}} \\ &\leq \delta|x - y|. \end{aligned}$$

Für ein Caplet im Libor Marktmodell können somit alle Kriterien des zentralen Satzes erfüllt werden, was dazu führt, dass unter Verwendung des Euler-Maruyama Diskretisierungsverfahrens bzw. des Milstein Diskretisierungsverfahrens theoretisch eine Konvergenzordnung  $\mathcal{O}(\epsilon^{-2}(\ln(\epsilon)^2))$  bzw.  $\mathcal{O}(\epsilon^{-2})$  erreicht werden kann. Dies verbessert den Rechenaufwand im Vergleich zum Standard Monte Carlo Verfahren, welches die Konvergenzordnung  $\mathcal{O}(\epsilon^{-3})$  besitzt, deutlich.

Dieses Ergebnis motiviert uns, das Multilevel Verfahren auch für die weiteren zu bewertenden Zinsderivate zu nutzen. Auf eine theoretische Untersuchung der Konvergenzrate der Varianzen  $V_l$  der jeweiligen Korrekturterme im Multilevel Monte Carlo Schätzer wollen wir an dieser Stelle allerdings verzichten und im nächsten Kapitel – ähnlich wie in [Gil08a] – versuchen, einen numerischen „Beweis“ zu

erbringen, dass die Varianzen  $V_l$  für das Euler-Maruyama Verfahren – wie behauptet – von der Ordnung  $\mathcal{O}(h_l)$  und für das Milstein Verfahren von der Ordnung  $\mathcal{O}(h_l^2)$  sind bzw. die Konvergenzordnungen zumindest annähernd erreicht werden können.

## 7.2 Vorstellung der produktunabhängigen Algorithmen

### 7.2.1 Allgemeines Setting und Parameterübergabe

In den folgenden Abschnitten kommen wir nun zur praktischen Umsetzung der bisher theoretisch diskutierten Verfahren.

Für die Simulation der Libor-Raten-Pfade, auf der die Produktbewertung via Monte Carlo Verfahren im Libor Marktmodell basiert, sind zunächst diverse Parameter festzulegen.

Dazu klären wir zuerst, welche Parameter wir im Allgemeinen, also produktunabhängig, für das (Multilevel) Monte Carlo Verfahren vorgeben müssen. In unserem Fall benötigen wir für jeden Level  $L$  die (feinere) Schrittweite  $h_L$  für die Diskretisierung eines Tenorintervalls, den Multilevel-Verfeinerungsfaktor  $M$  sowie die Anzahl  $N_L$  der für den  $L$ -ten Level zu simulierenden Libor-Raten-Pfade.

Um verschiedene Zinsderivate bewerten zu können, müssen wir die Simulation der Libor-Raten-Pfade relativ allgemein halten. Wir simulieren die notwendigen Pfade deshalb immer nur für ein Tenorintervall und rufen die Funktion dann – wie wir später sehen werden – gegebenenfalls mehrmals auf, um komplettete Pfade zu simulieren. Das Zeitintervall  $[T_\tau, T_{\tau+1}]$ , für welches die diskreten Libor-Raten-Pfade simuliert werden, kann deshalb in Form des Tenorindex des  $\tau$  seines Anfangszeitpunktes und der Tenorintervalllänge  $\delta$  festgelegt und dem Algorithmus übergeben werden.

Ferner wollen wir die Menge der Indizes der zu simulierenden Libor-Raten flexibel lassen, d.h. im Simulationsalgorithmus wird zunächst offen gelassen, für welche Libor-Raten simulierte Pfade benötigt werden. Für die Festlegung dieser Indexmenge wird dem Algorithmus dann sowohl der Index  $a^\tau \geq a$  der ersten Libor-Rate, für die ein Pfad für das jeweilige Tenorintervall zu simulieren ist, als auch der Index  $b^\tau$  mit  $a^\tau < b^\tau \leq N-1$  der letzten zu simulierenden Libor-Rate  $L^{(b^\tau)}$  übergeben. Da wir die Libor-Raten-Dynamiken per Rückwärtsinduktion erhalten, beginnen wir auch die Simulation mit der  $b^\tau$ -ten Libor-Rate und gehen iterativ von hinten nach vorne vor, bis wir zuletzt einen Pfad der  $a^\tau$ -ten Libor-Rate  $L^{(a^\tau)}$  für das vorgegebene Zeitintervall simuliert haben. Zu bemerken ist, dass dabei sichergestellt werden muss, dass  $\tau + 1 \leq a^\tau$  gewählt wird, da ein Pfad der  $a^\tau$ -ten Libor-Rate natürlich nur bis  $T_{a^\tau}$  simuliert werden kann.

Weiterhin müssen wir einen Vektor  $L_\tau := L^{(i)}(T_\tau)_{a^\tau \leq i \leq b^\tau}$  der Anfangswerte der zu simulierenden Libor-Raten übergeben. Benötigen wir beispielsweise Libor-Raten-Pfade für das Tenorintervall  $[T_1, T_2]$ , so müssen wir zuerst ausgehend von den „Spot“ Libor-Raten, also den aktuellen und somit gegebenen Libor-Raten in  $T_0$ , die Pfade für das Intervall  $[T_0, T_1]$  simulieren, um anschließend den Anfangswert

für das nächste Intervall als Endwert dieses Intervalls übergeben zu können.

Darüber hinaus gehen wir davon aus, dass die Gesamtzahl  $N-1$  der Komponenten des korrelierten Wiener-Prozesses und dessen Korrelationsmatrix  $\Sigma$  zur Verfügung stehen. Zu bemerken ist hier, dass  $b^\tau$  und  $N-1$  nicht unbedingt übereinstimmen müssen. Dies liegt daran, dass wir zwar alle vorhandenen Libor-Raten und damit alle vorhandenen Komponenten des treibenden Wiener-Prozesses benötigen, um, wie in Abschnitt 4.2 gesehen, die Cholesky-Matrix  $A$  zu bestimmen, welche bei der Simulation die Korrelation zwischen den Wiener-Komponenten erzeugt, wir für die Bewertung allerdings immer zum Forwardmartingalmaß  $\mathbb{P}_{T_{b^\tau+1}}$  übergehen können. Dieses legen wir dann „künstlich“ als „terminal measure“ fest. Dies ist möglich, da wir für die Bewertung theoretisch zu jedem beliebigen äquivalenten Forwardmartingalmaß übergehen können, wir so aber gerade das wählen, für welches Pfade für die wenigsten Libor-Raten simuliert werden müssen, gleichzeitig aber die Libor-Raten-Dynamiken für alle notwendigen Libor-Raten bzgl. dieses Maßes darstellbar sind.

Im Folgenden werden wir die Anzahl der Diskretisierungsschritte pro Tenorintervall  $\frac{\delta}{h_L} = M^L$  mit  $\eta$  bezeichnen und der Einfachheit halber auf den Index  $L$ , welcher den Simulationslevel angibt, verzichten. Mithilfe der Definition  $t_{n/M}^\tau := t_{(\eta\tau+n)/M}$  werden dann zuletzt noch die an den entsprechenden Stellen ausgewerteten deterministischen Volatilitätsfunktionen  $\sigma^{(i)}(t_{n/M}^\tau)$  für alle  $a^\tau \leq i \leq b^\tau$  und alle  $0 \leq n \leq \eta - 1$ , also für alle Diskretisierungszeitpunkte  $t_0^\tau = T_\tau \leq t_{1/M}^\tau = T_\tau + h_L \leq \dots \leq t_{(\eta-1)/M}^\tau = T_{\tau+1} - h_L$ , benötigt und in der Form einer  $((b^\tau - a^\tau + 1) \times \eta)$ -dimensionalen Matrix  $\sigma$  an den Algorithmus übergeben. In jeder Zeile  $i \in \{1, \dots, b^\tau - a^\tau + 1\}$  dieser Matrix stehen dabei jeweils die Volatilitäten  $\sigma^{(i)}(t_{n/M}^\tau)$  für  $0 \leq n \leq \eta - 1$ . Die Schreibweise  $t_{n/M}^\tau$  mit Brüchen in den Indizes wählen wir, da wir für einen Indexschritt bei der Pfadsimulation bzgl. der größeren Schrittweite je  $M$  ( $= 4$ ) Diskretisierungsschritte der Pfadsimulation bzgl. der feineren Schrittweite zusammenfassen wollen.

Die oben beschriebene abschnittsweise Simulation der Libor-Raten-Pfade soll in Abbildung 2 verdeutlicht werden.

Bevor wir zum Simulationsalgorithmus kommen, wollen wir noch bemerken, dass wir bei der Bezeichnung der approximierten Libor-Raten-Pfade in diesem Abschnitt der Einfachheit halber auf eine Kennzeichnung, dass es sich um einen Teilpfad für das Intervall  $[T_\tau, T_{\tau+1}]$  handelt, verzichten wollen. Wir werden diese Pfade folglich beispielsweise einfach mit  $L_{(n+1)/M}$  bezeichnen, wobei eigentlich  $L_{(\eta\tau+(n+1))/M}$  gemeint ist. Dies ist unproblematisch, da die Libor-Raten-Pfade mithilfe des Algorithmus jeweils nur für je ein Intervall simuliert werden.

### 7.2.2 Simulation der Libor-Raten-Pfade (Euler-Maruyama Diskretisierung)

Mithilfe der oben eingeführten Bezeichnungen sowie der Kenntnis darüber, welche Parameter übergeben werden müssen, können wir den Algorithmus zur Simulation

## 7.2 Vorstellung der produktunabhängigen Algorithmen

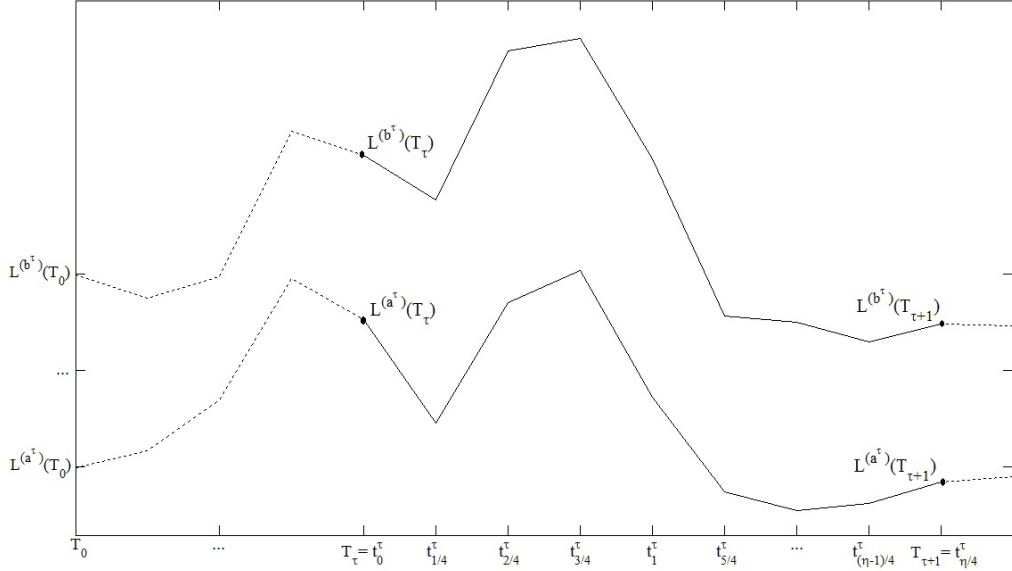


Abbildung 2: Intervallweise Simulation der benötigten Libor-Raten-Pfade ( $M=4$ )

der Libor-Raten nun angeben. In dieser Version nutzen wir das Euler-Maruyama Verfahren zur Diskretisierung der Libor-Raten-Dynamiken. Mithilfe der Bezeichnung  $\Delta t := h_L = \frac{\delta}{M^L}$  für die feinere Diskretisierungsschrittweite erfolgt die Simulation der diskretisierten Libor-Raten-Pfade dann wie folgt

**Algorithmus 7.1** (Multilevel-Simulation der Libor-Raten mithilfe des Euler-Maruyama Verfahrens)

1. Setze  $\eta = \frac{\delta}{h_L}$  (Anzahl Diskretisierungsintervalle (pro Tenorintervall))
2. Erzeuge einen Pfad eines Wiener-Prozesses (feinere Schrittweite) bzw. dessen Inkremente:
  - (a) Erzeuge  $(N - 1) \times \eta$   $\mathcal{N}(0, \Delta t)$ -verteilte Zufallszahlen

$$\Delta \widetilde{W}_{n/M}^{f(i)} = \sqrt{\Delta t} Z_{i,n}, 1 \leq i \leq N - 1, 1 \leq n \leq \eta,$$

wobei  $Z_{i,n}$  je eine bzgl. der Standardnormalverteilung (unabhängig) gezogene Zufallszahl darstelle

- (b) Multipliziere die Matrix  $A$  aus der Cholesky-Zerlegung der Korrelationsmatrix  $\Sigma$  mit der Matrix dieser Zufallszahlen, um die Wiener-Prozesse zu korrelieren:

$$\begin{pmatrix} \Delta W_{1/M}^{f(1)} & \Delta W_{2/M}^{f(1)} & \dots & \Delta W_{\eta/M}^{f(1)} \\ \Delta W_{1/M}^{f(2)} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ \Delta W_{1/M}^{f(N-1)} & \dots & \dots & \Delta W_{\eta/M}^{f(N-1)} \end{pmatrix} :=$$

$$A \cdot \begin{pmatrix} \Delta \widetilde{W}_{1/M}^{f(1)} & \Delta \widetilde{W}_{2/M}^{f(1)} & \dots & \Delta \widetilde{W}_{\eta/M}^{f(1)} \\ \Delta \widetilde{W}_{1/M}^{f(2)} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ \Delta \widetilde{W}_{1/M}^{f(N-1)} & \dots & \dots & \Delta \widetilde{W}_{\eta/M}^{f(N-1)} \end{pmatrix}$$

- (c) Schränke die Matrix der erzeugten Wiener-Prozesse auf die benötigten Einträge ein:

$$\Delta W^f := \begin{pmatrix} \Delta W_{1/M}^{f(a^\tau)} & \Delta W_{2/M}^{f(a^\tau)} & \dots & \Delta W_{\eta/M}^{f(a^\tau)} \\ \Delta W_{1/M}^{f(a^\tau+1)} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ \Delta W_{1/M}^{f(b^\tau)} & \dots & \dots & \Delta W_{\eta/M}^{f(b^\tau)} \end{pmatrix}$$

3. Für alle Diskretisierungsstellen  $n$ , d.h.für alle  $n = 0, 1, \dots, \eta - 1$

- (a) Führe den nächsten Zeitschritt zur Erstellung des approximierten Pfades der Libor-Raten  $L_{(n+1)/M}^f \approx L^f(t_{(n+1)/M}^\tau)$  bzgl. der feineren Schrittweite durch:

Rückwärtsinduktion, von der  $b^\tau$ -ten bis zur  $a^\tau$ -ten Libor-Rate:

Für alle  $i = b^\tau, b^\tau - 1, \dots, a^\tau$

- i. Bestimme den approximierten Driftterm  $\mu_{n/M}^{f(i)} \approx \mu^{f(i)}(t_{n/M}^\tau)$ :

$$\mu_{n/M}^{f(i)} = \sum_{k=i+1}^{b^\tau} \frac{\delta L_{n/M}^{f(k)} \sigma^{(k)}(t_{n/M}^\tau) \varrho_{i,k}}{\delta L_{n/M}^{f(k)} + 1}$$

- ii. Erstelle den nächsten Wert des diskreten Pfades:

$$L_{(n+1)/M}^{f(i)} = L_{n/M}^{f(i)} + L_{n/M}^{f(i)} \sigma^{(i)}(t_{n/M}^\tau) (-\mu_{n/M}^{f(i)} \Delta t + \Delta W_{(n+1)/M}^{f(i)})$$

- (b) Falls nötig  $((n+1)/M \in \mathbb{N})$ , führe den nächsten Zeitschritt zur Erstellung des approximierten Pfades der Libor-Raten  $L_{(n+1)/M}^g \approx L^g(t_{(n+1)/M}^\tau)$  bzgl. der größeren Schrittweite durch:

Falls  $(n+1)/M \in \mathbb{N}$

- i. Definiere  $j := (n+1)/M$   
 ii. Fasse je  $M$  Wiener Inkremente bzgl. der feineren Schrittweite zusammen, um die Inkremente bzgl. der größeren Schrittweite zu erhalten:

$$\Delta W_j^{g(i)} = \sum_{m=1}^M \Delta W_{(j-1)+m/M}^{f(i)}$$

- iii. Führe den nächsten Zeitschritt zur Erstellung des approximierten Pfades der Libor-Raten  $L_j^g$  bzgl. der größeren Schrittweite durch:

Rückwärtsinduktion, von der  $b^\tau$ -ten bis zur  $a^\tau$ -ten Libor-Rate:

Für alle  $i = b^\tau, b^\tau - 1, \dots, a^\tau$

A. Bestimme den approximierten Driftterm:

$$\mu_{j-1}^{g(i)} = \sum_{k=i+1}^{b^\tau} \frac{\delta L_{j-1}^{g(k)} \sigma^{(k)}(t_{j-1}^\tau) \varrho_{i,k}}{\delta L_{j-1}^{g(k)} + 1}$$

B. Erstelle den nächsten Wert des diskreten Pfades:

$$L_j^{g(i)} = L_{j-1}^{g(i)} + L_{j-1}^{g(i)} \sigma^{(i)}(t_{j-1}^\tau) (-\mu_{j-1}^{g(i)} \Delta t + \Delta W_j^{g(i)})$$

### Bemerkung 7.2

Bei einer Multilevel Monte Carlo Simulation wird für den 0-ten Level eine gewöhnliche Monte Carlo Simulation durchgeführt. Wir können dazu obigen Algorithmus verwenden, indem wir den Abschnitt 3.(b) auslassen und somit nur Libor-Raten-Pfade bzgl. der „feineren“ Schrittweite  $h_0 = \frac{\delta}{M^0} = \delta$  erzeugen.

### 7.2.3 Der Multilevel Monte Carlo Algorithmus zur Preisbestimmung von Derivaten im Libor Marktmodell

Nachdem wir gesehen haben, wie wir die diskretisierten Libor-Raten-Pfade für je zwei aufeinanderfolgende Diskretisierungsschrittweiten (Kennzeichnung:  $f$  (fein) und  $g$  (grob)) eines Simulationslevels erzeugen können, wollen wir diese im Folgenden nutzen, um Zinsderivate mithilfe von Multilevel Monte Carlo Simulationen zu bewerten. Im Algorithmus verwenden wir dazu aber zunächst weiter kein konkretes Derivat, sondern wollen diesen erst in einer allgemeinen Version vorstellen.

Zunächst ist zu klären, welche Parameter dem Algorithmus, der im Gesamtbewertungskonstrukt als Erstes aufgerufen wird, zu übergeben sind. Zum einen benötigen wir dazu die Parameter, welche das Libor Marktmodell bestimmen, also die Anzahl an Tenorzeitpunkten  $N$ , die Tenorintervalllänge  $\delta$  und die Volatilitätsfunktionen  $(\sigma^{(i)})_i$  sowie die Korrelationsmatrix  $\Sigma$  der Wiener-Prozesse. Zum anderen werden die jeweiligen produktspezifischen Parameter, auf die wir im nachfolgenden Abschnitt näher eingehen werden, sowie der Multilevel-Verfeinerungsfaktor  $M$  und eine initiale Anzahl  $N_{\text{Start}}$  an pro Level zu simulierenden Pfaden übergeben. Darüber hinaus benötigen wir einen zum Diskontieren der Auszahlung des Derivats notwendigen Bondpreis  $B(0, T_i)$  für ein  $i \leq N$ , je nachdem, zum welchem Zeitpunkt wir den Terminpreis der Auszahlungen bestimmen wollen. Letztlich müssen noch die Startwerte  $L_0 := L(T_0)$  der Libor-Raten zum aktuellen Zeitpunkt und eine vorher festzulegende Fehlertoleranz  $\epsilon > 0$  übergeben werden.

Mithilfe dieser Parameter werden dann für jeden Level zunächst  $N_{\text{Start}}$  Libor-Raten-Pfade simuliert, womit im Anschluss die vorliegende Stichprobenvarianz dieser Samples bestimmt wird. Mithilfe dieser Varianz wird dann die optimale Anzahl  $(N_l^*)_{0 \leq l \leq L}$  an für die jeweiligen Level zu simulierenden Pfaden bestimmt. Der Multilevel Monte Carlo Algorithmus, welchen wir zur Bestimmung eines Preises für ein beliebiges Zinsderivat nutzen, lautet dann

**Algorithmus 7.3** ((Multilevel) Monte Carlo Algorithmus im Libor Marktmodell)

1. Den Level  $L = 0$  müssen wir in einem Extrafall behandeln:
  - (a) Setze  $L = 0$ ,  $h_0 = \delta$  und  $N_0 = N_{\text{Start}}$
  - (b) Simuliere mithilfe von Algorithmus 7.1  $N_0$  diskrete Libor-Raten-Pfade für alle benötigten Libor-Raten und bestimme die zugehörigen Preise  $P_0^{(j)}$ ,  $1 \leq j \leq N_0$ , des Derivats mittels der produktsspezifischen Bewertungsalgorithmen, welche im Anschluss angegeben werden, als diskontierte Terminpreise der simulierten Payoffs
  - (c) Bestimme den Mittelwert der simulierten Preise als Approximation für den Erwartungswert der diskontierten Auszahlungen und somit als Approximation für den Preis des Derivats. Bestimme auch die Stichprobenvarianz der simulierten Preise:

$$E_0 = \frac{1}{N_0} \sum_{j=1}^{N_0} P_0^{(j)}, \quad V_0 = \frac{1}{N_0 - 1} \sum_{j=1}^{N_0} (P_0^{(j)} - E_0)^2$$

- (d) Mithilfe dieser Varianz kann die optimale Anzahl (vgl. (3.23)) an zu simulierenden Pfaden für diesen Level bestimmt werden:

$$N_0^* = \lceil 2\epsilon^{-2} \sqrt{V_0 h_0} \sqrt{V_0/h_0} \rceil = \lceil 2\epsilon^{-2} V_0 \rceil$$

- (e) Falls nötig ( $N_0^* > N_0$ ), simuliere weitere Pfade sowie die zugehörigen Preise und aktualisiere den Mittelwert und die Stichprobenvarianz:

Falls  $N_0^* > N_0$

- i. Simuliere  $N_0^* - N_0$  weitere diskrete Libor-Raten-Pfade und bestimme die zugehörigen Preise  $P_0^{(j)}$ ,  $N_0 + 1 \leq j \leq N_0^*$
- ii. Aktualisiere den Mittelwert und die Stichprobenvarianz:

$$E_0 = \frac{1}{N_0^*} \sum_{j=1}^{N_0^*} P_0^{(j)}, \quad V_0 = \frac{1}{N_0^* - 1} \sum_{j=1}^{N_0^*} (P_0^{(j)} - E_0)^2$$

- iii. Setze  $N_0 = N_0^*$

2. Setze die Variable, in der der aktuelle Bias gespeichert wird, so, dass die Abbruchbedingung ( $bias < \frac{1}{\sqrt{2}}(M - 1)\epsilon$ ) für die ersten beiden Level noch nicht erfüllt wird:

$$bias = \frac{1}{\sqrt{2}}(M - 1)\epsilon + 1$$

Dazu wollen wir daran erinnern, dass der Bias erst ab dem Level 2 abgeschätzt werden kann, weshalb wir ihn hier zunächst „künstlich“ festlegen.

3. Für die nächsten Level durchlaufe eine Schleife solange, bis die gewünschte Fehlertoleranz erreicht ist. Für die Level 1 und 2 soll allerdings auf jeden Fall simuliert werden:

Solange  $L < 2$  oder  $bias \geq \frac{1}{\sqrt{2}}(M - 1)\epsilon$

- (a) Setze  $L=L+1$
- (b) Setze  $h_L = \delta M^{-L}$  und  $N_L = N_{\text{Start}}$
- (c) Simuliere mithilfe von Algorithmus 7.1  $N_L$  Paare von diskreten Libor-Raten-Pfaden bzgl. der beiden Schrittweiten  $h_L$  (für den „feineren Pfad“) und  $h_{L-1}$  (für den „größeren Pfad“) für alle benötigten Libor-Raten. Mittels der produktspezifischen Bewertungsalgorithmen bestimme anschließend für jedes Paar von simulierten Pfaden die zugehörigen Preise  $P_L^{f(j)}$  und  $P_{L-1}^{g(j)}$ ,  $1 \leq j \leq N_L$ , des Derivats als diskontierte Terminpreise der simulierten Payoffs
- (d) Bestimme je die Differenz  $D_L^{(j)}$  der beiden Preise, die sich aus den Paaren der Libor-Raten-Pfade eines Simulationslevels ergeben, sowie den Erwartungswert und die Stichprobenvarianz dieser Differenzen:

$$D_L^{(j)} = P_L^{f(j)} - P_{L-1}^{g(j)}, 1 \leq j \leq N_L$$

$$E_L = \frac{1}{N_L} \sum_{j=1}^{N_L} D_L^{(j)}$$

$$V_L = \frac{1}{N_L - 1} \sum_{j=1}^{N_L} (D_L^{(j)} - E_L)^2$$

- (e) Mithilfe dieser Varianz kann die optimale Anzahl (vgl. (3.23)) an simulierten Pfaden für alle bisherigen Level bestimmt werden:

Für alle  $l = 0, 1, \dots, L$

- i. Setze  $N_l^* = \lceil 2\epsilon^{-2}\sqrt{V_l h_l} (\sum_{i=0}^L \sqrt{V_i/h_i}) \rceil$
- ii. Falls nötig ( $N_l^* > N_l$ ), simuliere weitere Paare von Pfaden, bestimme die zugehörigen Preise und aktualisiere den Mittelwert und die Stichprobenvarianz:

Falls  $N_l^* > N_l$

- A. Simuliere  $N_l^* - N_l$  weitere Paare diskreter Libor-Raten-Pfade und bestimme die zugehörigen Preise  $P_l^{(j)}$  für  $l = 0$  bzw. Preisdifferenzen  $D_l^{(j)} = P_l^{f(j)} - P_{l-1}^{g(j)}$  für  $l > 0$  und alle  $N_l + 1 \leq j \leq N_l^*$

- B. Aktualisiere den Mittelwert und die Stichprobenvarianz:

$$E_l = \frac{1}{N_l^*} \sum_{j=1}^{N_l^*} P_l^{(j)}, V_l = \frac{1}{N_l^* - 1} \sum_{j=1}^{N_l^*} (P_l^{(j)} - E_l)^2 \text{ für } l = 0$$

bzw.

$$E_l = \frac{1}{N_l^*} \sum_{j=1}^{N_l^*} D_l^{(j)}, V_l = \frac{1}{N_l^* - 1} \sum_{j=1}^{N_l^*} (D_l^{(j)} - E_l)^2 \text{ für } l > 0$$

- C. Setze  $N_l = N_l^*$

(f) Führe gegebenenfalls die Biasschätzung durch:

Falls  $L \geq 2$

$$\text{Setze } bias = \max\{|E_L|, M^{-1}|E_{L-1}|\}$$

4. Setze obige Approximationen für den Preis bzw. die Preisdifferenzen zusammen, um den MLMC-Schätzwert für den Preis  $P$  des Derivats zu erhalten:

$$P = E_0 + \sum_{l=1}^L E_l$$

### 7.3 Vorstellung der produktabhängigen Algorithmen

Nachdem wir die allgemeinen Multilevel Monte Carlo Algorithmen zur Bewertung von Zinsderivaten im Libor Marktmodell angegeben haben, wollen wir im nächsten Abschnitt konkrete Zinsderivate betrachten und aufzeigen, wie die obigen beiden Algorithmen 7.1 und 7.3 für die Bewertung dieser Derivate genutzt werden können. Dazu geben wir die fehlenden Algorithmen an, die zur Bewertung der im Kapitel 6 vorgestellten CMS Caps und TARNs notwendig sind.

#### 7.3.1 Bewertung von CMS Caps

Wie wir im Abschnitt 6.2 gezeigt haben, ergibt sich der Wert eines CMS Caps zum Zeitpunkt  $t$  als Summe der diskontierten erwarteten Auszahlungen zu

$$V_{\text{CMSCap}}^{a,b}(t) = \sum_{j=a}^{b-1} \delta B(t, T_b) \mathbb{E}^{\mathbb{P}_{T_b}} [B(T_{j+1}, T_b)^{-1} (R_{\text{Swap}}^{j,j+m}(T_j) - K)^+ | \mathcal{F}_t].$$

Um die Swap-Rate  $(R_{\text{Swap}}^{j,j+m}(t))_t$  nicht ebenfalls simulieren zu müssen, haben wir gesehen (vgl. (6.6)), dass wir diese durch den Ausdruck

$$\begin{aligned} R_{\text{Swap}}^{j,j+m}(T_j) &= \frac{1 - \prod_{k=j}^{j+m-1} \frac{1}{\delta L^{(k)}(T_j) + 1}}{\sum_{k=j}^{j+m-1} \delta \prod_{i=j}^k \frac{1}{\delta L^{(i)}(T_j) + 1}} \\ &= \frac{1 - \prod_{k=j}^{j+m-1} (\delta L^{(k)}(T_j) + 1)^{-1}}{\sum_{k=j}^{j+m-1} \delta \prod_{i=j}^k (\delta L^{(i)}(T_j) + 1)^{-1}} \end{aligned}$$

ersetzen können. Im Folgenden wollen wir damit nun den Wert eines CMS Caps im Zeitpunkt  $t = T_0 = 0$  bestimmen.

Im Gesamtbewertungsprozess wird der nachfolgende Algorithmus dann für jeden Level  $L$  vom Algorithmus 7.3 aufgerufen. Dabei werden für den 0-ten Level  $N_0$  Preise bzw. für die Level  $L > 0$   $N_L$  Paare von Preisen bzgl. der beiden Diskretisierungsschrittweiten zurückgegeben. Die produktspezifischen Parameter  $a, b, m$  und  $K$  sowie die oben definierten Parameter  $\delta, N, \Sigma, M, N_L, h_L, L_0$  und  $B(0, T_b)$  sind dazu an den Algorithmus zu übergeben. Der Parameter  $a$  ist dabei der Index des Tenorzeitpunktes  $T_a$ , in dem der CMS Cap startet, d.h. des Zeitpunktes, in dem die Swap-Rate erstmalig gefixt wird. Der Parameter  $b$  bezeichnet den Index des Tenorzeitpunktes  $T_b$ , in dem der Cap endet, d.h. des Endzeitpunktes

des Intervalls  $[T_{b-1}, T_b]$ , in dem letztmalig ein Tausch der festen Zinsrate  $K$  und der Swap-Rate stattfindet. Der Parameter  $m$  bezeichnet wieder die Laufzeit der „Constant-Maturity Swaps“. Zum Diskontieren der Auszahlung nutzen wir den Bond  $B(0, T_b)$ .

Nach einer Bemerkung dazu, welche Libor-Raten zu welchen Zeitpunkten für die Bewertung benötigt werden, können wir den Algorithmus damit angeben.

**Bemerkung 7.4**

Für die Bewertung eines CMS Caps werden die Libor-Raten  $L^{(i)}(T_j)$  zu den Zeitpunkten  $T_j$  mit  $a \leq j \leq b-1$  und  $j \leq i \leq j+m-1$  benötigt, d.h., dass insgesamt die diskreten Libor-Raten-Pfade der Libor-Raten  $L^{(i)}$  mit  $i \in \{a, \dots, b-1+m-1\}$  startend aus  $T_0$  je bis zum Zeitpunkt  $\min\{T_i, T_{b-1}\}$  simuliert werden müssen.

**Algorithmus 7.5** (Bewertung von CMS Caps)

1. Setze  $\eta = \frac{\delta}{h_L} = M^L$  (Anzahl Diskretisierungsintervalle (pro Tenorintervall))
2. Bestimme zunächst die deterministischen Volatilitäten für die erforderlichen Libor-Raten und für die vom Level  $L$  abhängigen Zeitpunkte, an denen die Volatilität ausgewertet werden muss:

Für alle  $i = a, a+1, \dots, b-1+m-1$ :

Bestimme die  $\sigma^{(i)}(t_{n/M})$  für alle  $0 \leq n \leq \min\{\eta i - 1, \eta(b-1) - 1\}$ , indem die deterministischen Volatilitätsfunktionen an den entsprechenden Stellen ausgewertet werden. Die  $(t_{n/M})_n$  sind dabei wie oben durch  $t_0 = T_0 \leq t_{1/M} = T_0 + h_L \leq \dots \leq t_{(\min\{\eta i - 1, \eta(b-1) - 1\})/M} = T_{\min\{i, b-1\}} - h_L$  definiert.

3. Erzeuge mithilfe von Algorithmus 7.1  $N_L$  Paare von diskreten Libor-Raten-Pfaden für die erforderlichen Libor-Raten:

- (a) Für die Intervalle  $[T_0, T_1]$  bis  $[T_{a-1}, T_a]$  werden je Pfade für alle oben genannten Libor-Raten benötigt, d.h.

Für alle  $i = a, a+1, \dots, b-1+m-1$ :

- i. Erzeuge  $L_{n/M}^{f(i)}$  für alle  $1 \leq n \leq \eta a$  als Approximation für die  $L^{(i)}(T_0 + h_L), L^{(i)}(T_0 + 2h_L), \dots, L^{(i)}(T_a)$
- ii. Erzeuge  $L_j^{g(i)}$  für alle  $1 \leq j \leq \eta a/M$  als Approximation für die  $L^{(i)}(T_0 + Mh_L), L^{(i)}(T_0 + 2(Mh_L)), \dots, L^{(i)}(T_a)$

- (b) Für die nachfolgenden Tenorintervalle  $[T_a, T_{a+1}]$  bis  $[T_{b-2}, T_{b-1}]$  fällt je eine zu simulierende Libor-Rate weg, d.h.

Für alle  $k = a, a+1, \dots, b-2$ :

Für alle  $i = k+1, k+2, \dots, b-1+m-1$ :

- i. Erzeuge  $L_{(\eta k+n)/M}^{f(i)}$  für alle  $1 \leq n \leq \eta$  als Approximation für die  $L^{(i)}(T_k + h_L), L^{(i)}(T_k + 2h_L), \dots, L^{(i)}(T_{k+1})$

- ii. Erzeuge  $L_{\eta k/M+n}^g$  für alle  $1 \leq n \leq \eta/M$  als Approximation für die  

$$L^{(i)}(T_k + Mh_L), L^{(i)}(T_k + 2(Mh_L)), \dots, L^{(i)}(T_{k+1})$$
- 4. Bestimme für jedes der  $N_L$  Paare von diskreten Libor-Raten-Pfaden die zu den je zwei Pfaden gehörigen Auszahlungen des CMS Caps:

Für alle Auszahlungszeitpunkte  $T_{j+1}$  mit  $a \leq j \leq b-1$ , d.h. für alle  $j = a, a+1, \dots, b-1$ :

- (a) Bestimme die jeweiligen (approximierten) Bondpreise mithilfe der oben erzeugten diskreten Libor-Raten-Pfade bzgl. der feineren bzw. größeren Schrittweite als:

$$B^f(T_{j+1}, T_b) = \prod_{k=j+1}^{b-1} (\delta L_{\eta(j+1)/M}^{(k)} + 1)^{-1}$$

bzw.

$$B^g(T_{j+1}, T_b) = \prod_{k=j+1}^{b-1} (\delta L_{\eta(j+1)/M}^{(k)} + 1)^{-1}$$

- (b) Bestimme die jeweiligen (approximierten) Swap-Raten bzgl. der feineren bzw. größeren Schrittweite als:

$$R_{\text{Swap}}^{j,j+m} f(T_j) = \frac{1 - \prod_{k=j}^{j+m-1} (\delta L_{\eta j/M}^{(k)} + 1)^{-1}}{\sum_{k_1=j}^{j+m-1} \delta \prod_{k_2=j}^{k_1} (\delta L_{\eta j/M}^{(k_2)} + 1)^{-1}}$$

bzw.

$$R_{\text{Swap}}^{j,j+m} g(T_j) = \frac{1 - \prod_{k=j}^{j+m-1} (\delta L_{\eta j/M}^{(k)} + 1)^{-1}}{\sum_{k_1=j}^{j+m-1} \delta \prod_{k_2=j}^{k_1} (\delta L_{\eta j/M}^{(k_2)} + 1)^{-1}}$$

- 5. Bestimme die jeweiligen zugehörigen Preise des CMS Caps:

$$V_{\text{CMSCap}}^{a,b} f(T_0) = \sum_{j=a}^{b-1} B(0, T_b) \delta B^f(T_{j+1}, T_b)^{-1} \max\{R_{\text{Swap}}^{j,j+m} f(T_j) - K, 0\}$$

bzw.

$$V_{\text{CMSCap}}^{a,b} g(T_0) = \sum_{j=a}^{b-1} B(0, T_b) \delta B^g(T_{j+1}, T_b)^{-1} \max\{R_{\text{Swap}}^{j,j+m} g(T_j) - K, 0\}$$

### Bemerkung 7.6

Wie bereits in Bemerkung 7.2 erklärt, wird beim Multilevel Monte Carlo Verfahren für das 0-te Level eine gewöhnliche Monte Carlo Simulation durchgeführt. Daher müssen wir auch hier auf eine Standard Monte Carlo Version dieses Algorithmus hinweisen. Wir können dazu obigen Algorithmus verwenden, indem wir überall statt je eines Paares wiederum nur den Wert bzgl. der „feineren“ Schrittweite  $h_0 = \delta$  bestimmen und somit am Ende anstatt von  $N_L$  Paaren nur  $N_L$  einzelne Preise für das Derivat erhalten und diese übergeben.

Wir haben nun alle Algorithmen, die für eine Bewertung von CMS Caps mittels des Multilevel Monte Carlo Verfahrens benötigt werden, angegeben und erläutert. Das Zusammenspiel der einzelnen in diesem Kapitel aufgeführten Algorithmen soll in Abbildung 3 noch einmal verdeutlicht werden.

### 7.3.2 Bewertung von TARNs

Für eine Bewertung des zweiten in dieser Arbeit betrachteten exotischen Zinsderivats müssen wir obigen Algorithmus 7.5 durch eine Version zur Bewertung von TARNs ersetzen, welche wir in diesem Abschnitt angeben.

Wie wir im Abschnitt 6.3 gesehen haben, ergibt sich der Wert eines TARNs zum Zeitpunkt  $t$  als Summe der erwarteten diskontierten Auszahlungen zu

$$\begin{aligned} V_{\text{TARN}}^b(t) = & B(t, T_b) \mathbb{E}^{\mathbb{P}_{T_b}} \left[ \left( \sum_{j=0}^{b-2} \underbrace{\left( \prod_{i=j+1}^{b-1} L^{(i)}(T_{j+1}) \delta + 1 \right)}_{=B(T_{j+1}, T_b)^{-1}} \mathbf{1}_{\{\sum_{k=1}^j K_k < R\}} \right. \right. \\ & \left. \left. \delta(\min\{K_{j+1}, R - \sum_{k=1}^j K_k\} - L^{(j)}(T_j)) \right. \right. \\ & \left. \left. + \mathbf{1}_{\{\sum_{k=1}^{b-1} K_k < R\}} \delta((R - \sum_{k=1}^{b-1} K_k) - L^{(b-1)}(T_{b-1})) | \mathcal{F}_t \right] . \right. \end{aligned}$$

Dabei werden für die strukturierten Kuponzahlungen  $(K_k)_{1 \leq k \leq b-1}$  wieder „inverse floating coupons“ der Form  $K_k = (K - gL^{(k-1)}(T_{k-1}))^+$  verwendet.  $K$  bezeichnet somit wieder die feste Zinszahlung des „inverse floaters“,  $g$  dessen Hebel und  $R$  den festgelegten Zielkupon.

Im Folgenden wollen wir nun diese Formel nutzen, um den Wert eines TARNs zum Zeitpunkt  $t = T_0$  zu bestimmen.

Dazu wird der nachfolgende Algorithmus statt des Algorithmus 7.5 dann analog zur Darstellung in Abbildung 3 für jeden Level  $L$  vom Algorithmus 7.3 aufgerufen, wodurch für den 0-ten Level  $N_0$  Preise, für die Level  $L > 0$   $N_L$  Paare von Preisen bzgl. der beiden unterschiedlichen Diskretisierungsschrittweiten zurückgegeben werden.

Dem Algorithmus zur Bewertung eines TARNs werden dabei sowohl die produkt-spezifischen Parameter  $b$ ,  $R$ ,  $K$  und  $g$  als auch wieder die Parameter  $\delta$ ,  $N$ ,  $\Sigma$ ,  $M$ ,  $N_L$ ,  $h_L$ ,  $L_0$  und  $B(0, T_b)$  übergeben. Für die Bestimmung der Zinszahlung im ersten Tenorintervall  $[T_0, T_1]$  muss außerdem die aktuelle Zinsrate für dieses Intervall übergeben werden, d.h. wir benötigen im Prinzip die bekannte, also  $\mathcal{F}_{T_0}$ -messbare, „Libor-Rate“  $L^{(0)}(T_0)$ . Zum Diskontieren der Auszahlung nutzen wir abermals den Bond  $B(0, T_b)$ .

Damit können wir den Algorithmus nach einer erneuten Bemerkung dazu, welche Libor-Raten zu welchen Zeitpunkten für die Bewertung benötigt werden, nun ebenfalls angeben.

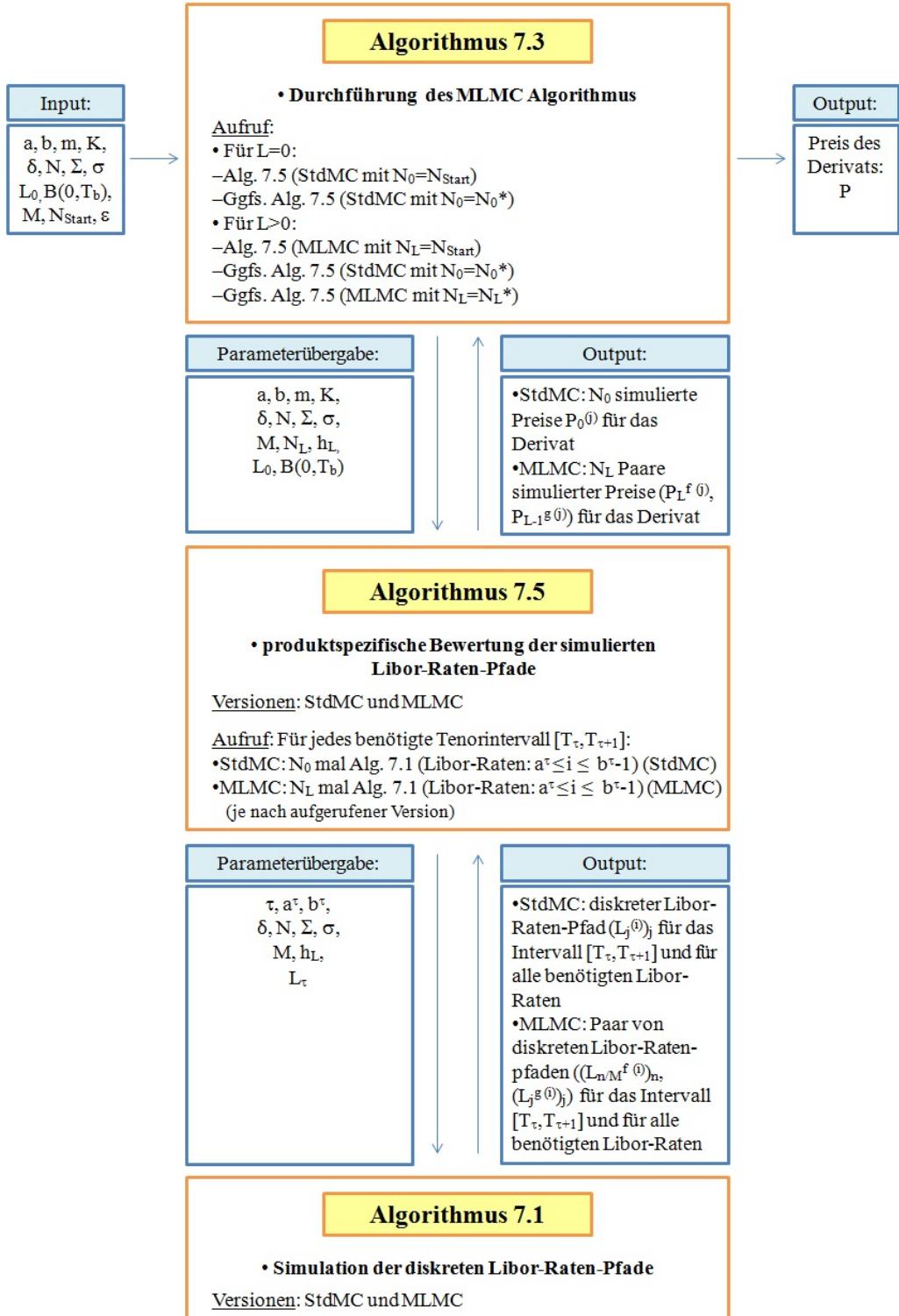


Abbildung 3: Übersicht über die Interaktion der vorgestellten Algorithmen am Beispiel der Bewertung eines CMS Caps (StdMC: Standard Monte Carlo, MLMC: Multilevel Monte Carlo)

**Bemerkung 7.7**

Für die Bewertung eines TARNs werden die Libor-Raten  $L^{(i)}(T_j)$  zu den Zeitpunkten  $T_j$  mit  $1 \leq j \leq b-1$  und  $j \leq i \leq b-1$  benötigt, d.h., dass insgesamt die diskreten Pfade der Libor-Raten  $L^{(i)}$  für alle  $1 \leq i \leq b-1$  aus  $L^{(i)}(T_0)$  startend jeweils bis zum Zeitpunkt  $T_i$ , des Beginns der jeweiligen Zinsperiode der  $i$ -ten Libor-Rate, simuliert werden müssen.

**Algorithmus 7.8** (Bewertung von TARNs)

1. Setze  $\eta = \frac{\delta}{h_L} = M^L$  (Anzahl Diskretisierungsintervalle (pro Tenorintervall))
2. Bestimme zunächst die deterministischen Volatilitäten für die erforderlichen Libor-Raten und für die vom Level  $L$  abhängigen Zeitpunkte, an denen die Volatilität ausgewertet werden muss:

Für alle  $i = 1, 2, \dots, b-1$ :

Bestimme wie oben die  $\sigma^{(i)}(t_{n/M})$  für alle  $0 \leq n \leq \eta i - 1$ .

3. Erzeuge mithilfe von Algorithmus 7.1  $N_L$  Paare von diskreten Libor-Raten-Pfaden für die erforderlichen Libor-Raten:

Für jedes Tenorintervall, startend mit  $[T_0, T_1]$  und endend mit  $[T_{b-2}, T_{b-1}]$ , fällt je eine zu simulierende Libor-Rate weg, d.h.

Für alle  $k = 0, 1, \dots, b-2$ :

Für alle  $i = k+1, k+2, \dots, b-1$ :

- (a) Erzeuge  $L_{(\eta k+n)/M}^{(i)}$  für alle  $1 \leq n \leq \eta$  als Approximation für die  $L^{(i)}(T_k + h_L), L^{(i)}(T_k + 2h_L), \dots, L^{(i)}(T_{k+1})$
- (b) Erzeuge  $L_{\eta k/M+n}^{(i)}$  für alle  $1 \leq n \leq \eta/M$  als Approximation für die  $L^{(i)}(T_k + Mh_L), L^{(i)}(T_k + 2(Mh_L)), \dots, L^{(i)}(T_{k+1})$

4. Bestimme für jedes der  $N_L$  Paare von diskreten Libor-Raten-Pfaden die zu den je zwei Pfaden gehörigen Terminpreise der Auszahlungen des TARNs zum Termin  $T_b$ :

- Bestimme zunächst die Terminpreise der Auszahlungen bzgl. der Pfade mit der feineren Schrittweite:

- (a) Für alle Auszahlungszeitpunkte  $T_{j+1}$  mit  $0 \leq j \leq b-2$ , d.h. für alle  $j = 0, 1, \dots, b-2$ :
  - i. Initialisiere den Terminpreis der Auszahlung in  $T_{j+1}$  und setze dazu  $P_{T_b}^f(T_{j+1}) = 0$
  - ii. Überprüfe, ob der Zielkupon schon erreicht wurde, und bestimme gegebenenfalls die Auszahlung in  $T_{j+1}$ :

Falls  $\sum_{k=1}^j K_k^f < R$ :

- A. Bestimme den jeweiligen (approximierten) Bondpreis mithilfe der oben erzeugten diskreten Libor-Raten-Pfade als:

$$B^f(T_{j+1}, T_b) = \prod_{i=j+1}^{b-1} (L_{\eta(j+1)/M}^{(i)} \delta + 1)^{-1}$$

- B. Bestimme die jeweilige Kuponzahlung  $K_{j+1}$ :

$$K_{j+1}^f = \max\{K - g L_{\eta j/M}^{(j)}, 0\}$$

- C. Bestimme den jeweiligen Terminpreis  $P_{T_b}^f(T_{j+1})$  der Auszahlung des TARNs in  $T_{j+1}$  zum Termin  $T_b$ :

$$\begin{aligned} P_{T_b}^f(T_{j+1}) &= (B^f(T_{j+1}, T_b))^{-1} \\ &\quad \delta(\min\{K_{j+1}^f, R - \sum_{k=1}^j K_k^f\} - L_{\eta j/M}^{(j)}) \end{aligned}$$

- (b) Den letzten Auszahlungszeitpunkt  $T_b$  müssen wir aufgrund der möglichen Aufstockung der Zinszahlung zum Erreichen des Zielkupons gesondert betrachten:

- i. Initialisiere den Terminpreis der Auszahlung in  $T_b$  und setze dazu  $P_{T_b}^f(T_b) = 0$
- ii. Überprüfe, ob der Zielkupon schon erreicht wurde, und bestimme gegebenenfalls die Auszahlung in  $T_b$ :

Falls  $\sum_{k=1}^{b-1} K_k^f < R$ :

- A. Bestimme den Terminpreis  $P_{T_b}^f(T_b)$  der Auszahlung des TARNs in  $T_b$  zum Termin  $T_b$ :

$$P_{T_b}^f(T_b) = \delta((R - \sum_{k=1}^{b-1} K_k^f) - L_{\eta(b-1)/M}^{(b-1)})$$

- Für den Pfad mit der größeren Schrittweite führe obige Anweisungen analog durch und erhalte damit alle Terminpreise  $P_{T_b}^g(T_{j+1})$  für alle  $0 \leq j \leq b-1$

5. Bestimme die jeweiligen zugehörigen Preise des TARNs:

$$V_{\text{TARN}}^b(T_0) = B(0, T_b) \sum_{j=0}^{b-1} P_{T_b}^f(T_{j+1})$$

bzw.

$$V_{\text{TARN}}^b(T_0) = B(0, T_b) \sum_{j=0}^{b-1} P_{T_b}^g(T_{j+1})$$

**Bemerkung 7.9**

Analog zum Bewertungsalgorithmus 7.5 für CMS Caps wollen wir auch hier noch einmal darauf hinweisen, dass für die Standard Monte Carlo Version ( $L = 0$ ) des Algorithmus statt  $N_L$  Paaren je nur  $N_L$  einzelne Preise des Derivats bestimmt und übergeben werden müssen, d.h., dass überall nur die Werte bzgl. der „feineren“ Schrittweite  $h_0 = \delta$  bestimmt werden müssen.

## 7.4 Simulation der Libor-Raten-Pfade (Milstein Diskretisierung)

Bevor wir zu den numerischen Ergebnissen und dem Vergleich des Multilevel Monte Carlo Verfahrens mit der klassischen Monte Carlo Methode kommen, wollen wir noch auf den Algorithmus zur Simulation der diskreten Libor-Raten-Pfade eingehen, falls wir das Milstein Verfahren zur Diskretisierung verwenden, um auch diese Version in den Vergleich der Verfahren aufnehmen zu können.

Für die Umsetzung der Milstein Diskretisierung betrachten wir noch einmal den Algorithmus 7.1 zur (Multilevel Monte Carlo) Simulation der Libor-Raten-Pfade unter Benutzung des Euler-Maruyama Verfahrens. Offensichtlich müssen wir dort lediglich das Diskretisierungsverfahren anpassen, d.h. wir tauschen im obigen Algorithmus den Teil 3.(a).ii

$$L_{(n+1)/M}^{f(i)} = L_{n/M}^{f(i)} + L_{n/M}^{f(i)} \sigma^{(i)}(t_{n/M}^\tau) (-\mu_{n/M}^{f(i)} \Delta t + \Delta W_{(n+1)/M}^{f(i)})$$

aus und ersetzen diesen durch die entsprechende Milstein-Variante

$$\begin{aligned} L_{(n+1)/M}^{f(i)} &= L_{n/M}^{f(i)} + L_{n/M}^{f(i)} \sigma^{(i)}(t_{n/M}^\tau) (-\mu_{n/M}^{f(i)} \Delta t + \Delta W_{(n+1)/M}^{f(i)} \\ &\quad + \frac{1}{2} \sigma^{(i)}(t_{n/M}^\tau) ((\Delta W_{(n+1)/M}^{f(i)})^2 - \Delta t)). \end{aligned}$$

Entsprechend tauschen wir analog dazu für die gröbere Schrittweite den Teil 3.(b).iii.B des Algorithmus

$$L_j^{g(i)} = L_{j-1}^{g(i)} + L_{j-1}^{g(i)} \sigma^{(i)}(t_{j-1}^\tau) (-\mu_{j-1}^{g(i)} \Delta t + \Delta W_j^{g(i)})$$

aus und ersetzen ihn durch die Milstein-Variante

$$\begin{aligned} L_j^{g(i)} &= L_{j-1}^{g(i)} + L_{j-1}^{g(i)} \sigma^{(i)}(t_{j-1}^\tau) (-\mu_{j-1}^{g(i)} \Delta t + \Delta W_j^{g(i)} \\ &\quad + \frac{1}{2} \sigma^{(i)}(t_{j-1}^\tau) ((\Delta W_j^{g(i)})^2 - \Delta t)). \end{aligned}$$

## 8 Numerische Ergebnisse

### 8.1 Marktkalibrierung

Bevor wir in diesem Kapitel die numerischen Ergebnisse der Simulationen vorstellen, wollen wir kurz darauf eingehen, woher wir die modellspezifischen Parameter erhalten, welche wir den Algorithmen übergeben müssen.

Betrachten wir nochmals die dem Libor Marktmodell zugrunde liegende SDE (4.3)

$$dL^{(i)}(t) = L^{(i)}(t)\sigma^{(i)}(t)((-\sum_{k=i+1}^{N-1} \frac{\delta L^{(k)}(t)\sigma^{(k)}(t)\varrho_{i,k}}{\delta L^{(k)}(t) + 1})dt + dW^{(i)}(t)),$$

so sehen wir, dass dazu die deterministischen Volatilitäten  $(\sigma^{(i)}(t))_t$  sowie die Korrelationen  $\varrho_{i,k}$  der Einträge des  $(N-1)$ -dimensionalen Wiener-Prozesses  $W$  für alle  $1 \leq i, k \leq N-1$  zu spezifizieren sind. Um diese Parameter zu bestimmen, wird eine Kalibrierung des Modells an Marktdaten durchgeführt. Dies bedeutet, dass die freien Modellparameter so bestimmt werden sollen, dass die zu beobachtenden Marktvoltalitäten möglichst gut mit den Modellvoltalitäten übereinstimmen. „Möglichst gut“ bedeutet hier, dass der MSE der Markt- und Modellvoltalitäten minimiert werden soll. Im Folgenden werden wir die Vorgehensweise einer Kalibrierung grob vorstellen, indem wir uns an [Sch11] und [San13] orientieren. Für eine detaillierte Darstellung sowie die numerische Bestimmung der Parameter, welche anschließend zur Berechnung der Preise der Zinsderivate verwendet werden, verweisen wir auf die Arbeit [San13] von Santen.

Es gibt viele verschiedene Ansätze zur Kalibrierung von Libor Marktmodellen. Wir werden uns hier auf eine Kalibrierung nach Schoenmakers und Coffey (vgl. [SC00]) beschränken, wobei das Modell simultan an Caps und Swaptions kalibriert wird.

Da wir in dieser Arbeit ein Libor Marktmodell mit deterministischen Voltalitäten vorausgesetzt haben, kann das Modell den sogenannten „Smile-Effekt“ nicht darstellen. Dieser bezeichnet den Effekt, dass die implizite Voltalität von Marktpreisen, die vom Strike-Preis der betrachteten Derivate abhängt, für „at-the-money“ Derivate niedriger ist als die von „in-“ bzw. „out-of-the-money“ Derivaten. Dies führt zu einer Art „Smile“, wenn man die implizite Voltalität der Marktpreise in Abhängigkeit von den Strike-Preisen der Derivate darstellt. Wir wollen das Modell deshalb an „at-the-money“ Caps und Swaptions kalibrieren.

Bezeichnet  $T_a$  wieder den Zeitpunkt des ersten Fixings eines Swaps und  $T_b$  den Zeitpunkt der letzten Zinszahlung sowie  $R_{a,b}^{\text{Swap}}$  wie oben die Swap-Rate eines Swaps, welcher in  $T_a$  beginnt und in  $T_b$  endet, so können wir die am Markt quotierte Voltalität einer Swaption mithilfe einer Approximation von Rebonato (siehe [BM06]) angeben als

$$(\sigma_{a,b}^{\text{SM}})^2 \approx \sum_{i,j=a}^{b-1} \frac{w_i(0)w_j(0)L^{(i)}(0)L^{(j)}(0)\varrho_{i,j}}{R_{a,b}^{\text{Swap}}(0)} \frac{1}{T_a} \int_0^{T_a} \sigma^{(i)}(t)\sigma^{(j)}(t)dt, \quad (8.1)$$

wobei die Gewichte  $w_i$  durch

$$w_i(t) = \frac{\delta \prod_{k=a}^i (1 + \delta L^{(k)}(t))^{-1}}{\sum_{n=a}^{b-1} \delta \prod_{k=a}^n (1 + \delta L^{(k)}(t))^{-1}}$$

gegeben sind.

Um das Modell zugleich an Caplets zu kalibrieren, können wir die am Markt quotierte Volatilität eines Caplets für das Tenorintervall  $[T_i, T_{i+1}]$  mit  $1 \leq i \leq N-1$  angeben als

$$(\sigma_i^{\text{CM}})^2 = \frac{1}{T_i} \int_0^{T_i} \sigma^{(i)}(t)^2 dt. \quad (8.2)$$

Da am Markt meist nur Cap-Volatilitäten quotiert sind, weshalb wir oben von einer Kalibrierung an Caps gesprochen haben, müssen wir die Caplet-Volatilitäten zunächst aus den Cap-Volatilitäten ableiten. Dies kann beispielsweise mithilfe eines sogenannten „Cap-to-Caplet-Strippings“ (siehe [San13]) durchgeführt werden.

Im nächsten Schritt geben wir uns nun funktionale Formen für die Volatilitätsfunktionen  $\sigma^{(i)}(t)$  und für die Einträge  $(\varrho_{i,j})_{i,j}$  der Korrelationsmatrix  $\Sigma$  vor. Für die Volatilitäten verwenden wir wie in [Sch11] oder [San13] dazu die Funktion

$$\sigma^{(i)}(t) = \varphi_i g_{\alpha_1, \alpha_2, \alpha_3, \alpha_4}(T_i - t) := \varphi_i [(\alpha_1(T_i - t) + \alpha_4)\exp(-\alpha_2(T_i - t)) + \alpha_3] \quad (8.3)$$

mit Koeffizienten  $(\varphi_i)_i$  und Konstanten  $\alpha_1$  bis  $\alpha_4$ , welche den Bedingungen  $\alpha_1 + \alpha_3 > 0$ ,  $\alpha_2 > 0$  und  $\alpha_3 > 0$  genügen. Bei der Wahl der funktionalen Darstellung der Korrelationsmatrix müssen wir beachten, dass die Einträge  $(\varrho_{i,j})_{i,j}$  so gewählt werden, dass wir eine positiv semidefinite Matrix erhalten und alle Einträge nichtnegativ sind. Wie in [San13] verwenden hierzu die Funktion

$$\begin{aligned} \varrho_{i,j}(\gamma, \varrho_\infty) &= \exp\left(-\frac{|T_i - T_j|}{(N-1)-1}(-\ln(\varrho_\infty)\right. \\ &\quad \left. + \gamma \frac{T_i^2 + T_j^2 + T_i T_j - 3(N-1)T_i - 3(N-1)T_j + 3T_i + 3T_j + 2(N-1)^2 - (N-1) - 4}{((N-1)-2)((N-1)-3)})\right) \end{aligned} \quad (8.4)$$

mit  $0 < \varrho_\infty \leq 1$  und  $0 \leq \gamma \leq -\ln(\varrho_\infty)$ .

Setzen wir die Volatilitätsfunktion dann in die Darstellung (8.2) der Caplet-Marktvolatilität ein, so erhalten wir

$$(\sigma_i^{\text{CM}})^2 = \frac{\varphi_i^2}{T_i} \int_0^{T_i} (g_{\alpha_1, \alpha_2, \alpha_3, \alpha_4}(T_i - t))^2 dt. \quad (8.5)$$

Der entscheidende Schritt beim Ansatz von Schoenmakers und Coffey ist nun die Caplet-Volatilitäten in die Formel für die Swaption-Volatilität (8.1) einzusetzen. Auf diese Weise erhalten wir

$$\sigma_{a,b}(\alpha_1, \alpha_2, \alpha_3, \alpha_4, \gamma, \varrho_\infty) = \sqrt{\sum_{i,j=a}^{b-1} \frac{w_i(0)w_j(0)L^{(i)}(0)L^{(j)}(0)}{R_{a,b}^{\text{Swap}}(0)} \varrho_{i,j}(\gamma, \varrho_\infty) \sigma_i^{\text{CM}} \sigma_j^{\text{CM}} \theta_{i,j,a}^{\alpha_1, \alpha_2, \alpha_3, \alpha_4}}, \quad (8.6)$$

wobei  $\theta_{i,j,a}^{\alpha_1, \alpha_2, \alpha_3, \alpha_4}$  durch

$$\begin{aligned} \theta_{i,j,a}^{\alpha_1, \alpha_2, \alpha_3, \alpha_4} &:= \frac{1}{T_a} \frac{1}{\sigma_i^{\text{CM}} \sigma_j^{\text{CM}}} \int_0^{T_a} \sigma^{(i)}(t) \sigma^{(j)}(t) dt \\ &= \frac{1}{T_a} \frac{\varphi_i \varphi_j}{\sigma_i^{\text{CM}} \sigma_j^{\text{CM}}} \int_0^{T_a} g_{\alpha_1, \alpha_2, \alpha_3, \alpha_4}(T_i - t) g_{\alpha_1, \alpha_2, \alpha_3, \alpha_4}(T_j - t) dt \\ &\stackrel{(8.5)}{=} \frac{\sqrt{T_i} \sqrt{T_j}}{T_a} \frac{\int_0^{T_a} g_{\alpha_1, \alpha_2, \alpha_3, \alpha_4}(T_i - t) g_{\alpha_1, \alpha_2, \alpha_3, \alpha_4}(T_j - t) dt}{\sqrt{\int_0^{T_i} (g_{\alpha_1, \alpha_2, \alpha_3, \alpha_4}(T_i - t))^2 dt} \sqrt{\int_0^{T_j} (g_{\alpha_1, \alpha_2, \alpha_3, \alpha_4}(T_j - t))^2 dt}} \end{aligned}$$

definiert ist und aufgrund der gewählten Volatilitätsstruktur (8.3) analytisch bestimmt werden kann.

Das Minimierungsproblem ergibt sich dann, indem wir die mittlere quadratische relative Abweichung aller vorhandenen Swaption-Modellvolatilitäten (8.6) von den Swaption-Marktvolatilitäten minimieren. Das bedeutet, wir erhalten das Minimierungsproblem

$$\min_{\alpha_1, \alpha_2, \alpha_3, \alpha_4, \gamma, \varrho_\infty} \sqrt{\frac{1}{|I|} \sum_{(a,b) \in I} \frac{\sigma_{a,b}^{\text{SM}} - \sigma_{a,b}(\alpha_1, \alpha_2, \alpha_3, \alpha_4, \gamma, \varrho_\infty)}{\sigma_{a,b}^{\text{SM}}}}, \quad (8.7)$$

wobei  $|I|$  die Anzahl der Elemente in der Menge

$$I := \{(a, b) : \sigma_{a,b}^{\text{SM}} \text{ ist am Markt beobachtbar}\},$$

also der Menge aller Indizes, für die die Swaption-Volatilitäten am Markt quotiert sind, bezeichnet.

Nachdem das Minimierungsproblem gelöst ist, können wir zuletzt die Koeffizienten  $\varphi_i$  mithilfe von Gleichung (8.5) bestimmen und erhalten ein an Cap- und Swaption-Volatilitäten kalibriertes Modell.

### Bemerkung 8.1

Für die Bewertung der Zinsderivate werden wir – wie in [San13] – ein Modell mit Tenorintervalllänge  $\delta=0.5$  (Jahre) verwenden. Insgesamt wird eine Laufzeit von 20 Jahren, also  $N = 40$  Tenorintervallen, betrachtet. Das Modell wird deshalb an  $N-1 = 39$  Caplet-Volatilitäten ( $\sigma_i^{\text{CM}}$ )<sub>i</sub>, welche aus den gegebenen Cap-Volatilitäten ( $\sigma_{1,b}^{\text{CapM}}$ )<sub>b</sub> abgeleitet werden können, und alle vorhandenen Swaption-Volatilitäten ( $\sigma_{a,b}^{\text{SM}}$ )<sub>a,b</sub>, deren Laufzeit  $[T_a, T_b]$  mit Ausübungszeitpunkt  $T_a > T_0$

der Option spätestens in  $T_N = T_{40}$  endet, kalibriert. Die verwendeten Marktdaten werden im Anhang angegeben.

Für die numerische Bewertung der Zinsderivate mithilfe der in Kapitel 7 vorgestellten Algorithmen verwenden wir die in der Arbeit [San13] bestimmten Parameter  $\alpha_1 = -0.6790$ ,  $\alpha_2 = 0.3725$ ,  $\alpha_3 = 2.0594$  und  $\alpha_4 = 0.3261$  sowie  $\gamma = 0.7896$  und  $\varrho_\infty = 0.1154$ . Für die Koeffizienten  $(\varphi_i)_{1 \leq i \leq 39}$  geben wir die übernommenen Werte in der nachfolgenden Tabelle an.

$i$	$\varphi_i$	$i$	$\varphi_i$	$i$	$\varphi_i$
1	0.4921	14	0.2101	27	0.1270
2	0.3647	15	0.2073	28	0.1224
3	0.3703	16	0.1879	29	0.1190
4	0.4332	17	0.1722	30	0.1097
5	0.4626	18	0.1634	31	0.1096
6	0.4038	19	0.1617	32	0.1112
7	0.3494	20	0.1626	33	0.1144
8	0.3521	21	0.1648	34	0.1196
9	0.3461	22	0.1627	35	0.1268
10	0.3097	23	0.1587	36	0.1363
11	0.2655	24	0.1473	37	0.1483
12	0.2260	25	0.1396	38	0.1632
13	0.2071	26	0.1328	39	0.1816

Tabelle 1: Der Marktkalibrierung in [San13] entnommene Werte für die Koeffizienten  $(\varphi_i)_{1 \leq i \leq 39}$  der Volatilitätsfunktionen  $\sigma^{(i)}(t)$ ,  $1 \leq i \leq 39$

Für die Anwendung in den Simulationsalgorithmen, welche wir in Kapitel 7 vorgestellt haben, müssen wir die auf diese Art und Weise erhaltenen Volatilitätsfunktionen  $\sigma^{(i)}(t)$  dann dementsprechend nur noch für die jeweils benötigten Libor-Raten an den jeweils benötigten diskreten Zeitpunkten auswerten und an die Algorithmen übergeben.

## 8.2 Bewertung von Caplets

In den kommenden beiden Abschnitten wollen wir die durch eine Kalibrierung erhaltenen Modell-Parameter nutzen, um zunächst zu Demonstrationszwecken einfache Caplets und anschließend CMS Caps und TARNs zu bewerten.

Der Grund für eine Betrachtung von Caplets liegt darin, dass wir die numerisch bestimmten Preise mit Preisen, die mit Black's Formel für Caplets analytisch berechnet werden, vergleichen können und auf diese Weise exemplarisch gezeigt werden kann, dass die vorgestellten Methoden tatsächlich korrekte Ergebnisse liefern.

Für die Preisberechnung der Caplets verwenden wir den Algorithmus 7.1 in der Milstein-Variante, den Multilevel-Algorithmus 7.3 und eine für die Capletbewertung angepasste Version der Algorithmen 7.5 bzw. 7.8, die wir hier nicht explizit angeben wollen. Beispielhaft werden wir den Preis eines Caplets für das Tenor-

intervall  $[T_2, T_3] = [1, 1.5]$  mit Strike-Zinssatz  $K = 0.39\%$  bestimmen. Für die Bewertung werden weiterhin der Libor-Raten Startwert  $L^{(2)}(T_0) = 0.4203\%$  (siehe Tabelle 2) sowie der Bondpreis  $B(0, T_3) = 0.9946\text{ €}$  benötigt. Die aktuellen Libor-Raten sowie die Bondpreise können wir dabei über die am Markt vorliegenden Swap-Raten bestimmen und stammen somit aus demselben Satz an Marktdaten, der für die Kalibrierung genutzt wurde. Für weitere Details der Bestimmung der Libor-Raten Startwerte sowie der Bondpreise sei erneut auf die Arbeit [San13] verwiesen.

Um etwas realistischere Größenordnungen zu betrachten, wählen wir für die nachfolgende Preisbestimmung ein Nominal (Nom) in Höhe von 1 Million Euro. Der durch Multilevel Monte Carlo Simulationen bestimmte Preis eines solchen Caplets ergibt sich dann zu

$$V_{\text{Caplet}}^{2,3}(T_0) = \text{Nom} \cdot \delta B(T_0, T_3) \mathbb{E}^{\mathbb{P}_{T_3}}[(L^{(2)}(T_2) - K)^+ | \mathcal{F}_{T_0}] \approx 670.7671\text{ €},$$

wobei eine initiale Anzahl an zu simulierenden Pfaden  $N_{\text{Start}} = 10^4$  und eine Fehlerschranke  $\epsilon = 2 \cdot 10^{-6}$  vorgegeben wurde.

Bemerkt sei dabei noch, dass sich die Fehlerschranke hier – wie auch im Folgenden – auf den Wert des Produkts mit einem Nominal von 1 Euro bezieht. Dies bedeutet, dass der MSE des Simulationspreises eines Produktes mit Nominal von 1 Euro die Fehlerschranke  $\epsilon^2$  einhalten soll.

Wenn – wie in dieser Arbeit – von einem Libor Marktmodell mit deterministischer Volatilität ausgegangen wird, ist es möglich, Preise für Caplets auch analytisch zu bestimmen. Um hier beispielhaft zu zeigen, dass die in dieser Arbeit vorgestellten Monte Carlo Methoden korrekte Werte liefern, berechnen wir für obiges Caplet zum Vergleich den exakten Preis. Dies können wir mithilfe von Black's Formel zur Bewertung von Caplets, welche wir nachfolgend herleiten, durchführen.

In Kapitel 6 haben wir gesehen, dass bei einem Caplet zum (Tenor-)Zeitpunkt  $T_{n+1}$  eine Auszahlung in Höhe von  $\delta(L_n(T_n) - K)^+$  pro Einheit des Nominals geleistet wird. Betrachten wir dann den unter  $\mathcal{F}_t$  bedingten Erwartungswert dieser Auszahlung bzgl. des zur  $n$ -ten Libor-Rate gehörigen Forwardmartingalmaßes  $\mathbb{P}_{T_{n+1}}$ , so erhalten wir für  $t \leq T_n$

$$\begin{aligned} & \mathbb{E}^{\mathbb{P}_{T_{n+1}}}[\delta(L^{(n)}(T_n) - K)^+ | \mathcal{F}_t] \\ &= \mathbb{E}^{\mathbb{P}_{T_{n+1}}}[\delta L^{(n)}(T_n) \mathbf{1}_{\{L^{(n)}(T_n) > K\}} | \mathcal{F}_t] - K \delta \mathbb{P}_{T_{n+1}}[L^{(n)}(T_n) > K | \mathcal{F}_t] \\ &= L^{(n)}(t) \mathbb{E}^{\mathbb{P}_{T_{n+1}}}[\delta \frac{L^{(n)}(T_n)}{L^{(n)}(t)} \mathbf{1}_{\{L^{(n)}(T_n) > K\}} | \mathcal{F}_t] - K \delta \mathbb{P}_{T_{n+1}}[L^{(n)}(T_n) > K | \mathcal{F}_t] \\ &= L^{(n)}(t) \delta \mathbb{Q}_{T_{n+1}}[L^{(n)}(T_n) > K | \mathcal{F}_t] - K \delta \mathbb{P}_{T_{n+1}}[L^{(n)}(T_n) > K | \mathcal{F}_t], \end{aligned}$$

wobei das äquivalente Wahrscheinlichkeitsmaß  $\mathbb{Q}_{T_{n+1}}$  durch die Maßwechseldichte

$$\frac{d\mathbb{Q}_{T_{n+1}}}{d\mathbb{P}_{T_{n+1}}} \Big| \mathcal{F}_t = \frac{L^{(n)}(t)}{L^{(n)}(0)} \quad \forall 0 \leq t \leq T_n$$

gegeben ist.

Den Preis des Caplets in  $t$  erhalten wir dann durch Diskontieren mit dem  $T_{n+1}$ -Bond als

$$\begin{aligned} V_{\text{Caplet}}^{n,n+1}(t) &= B(t, T_{n+1}) L^{(n)}(t) \delta \mathbb{Q}_{T_{n+1}}[L^{(n)}(T_n) > K | \mathcal{F}_t] \\ &\quad - B(t, T_{n+1}) K \delta \mathbb{P}_{T_{n+1}}[L^{(n)}(T_n) > K | \mathcal{F}_t]. \end{aligned}$$

Da  $(L^{(n)}(t))_t$  nach Definition des Forwardmartingalmaßes ein  $\mathbb{P}_{T_{n+1}}$ -Martingal ist, können die Dynamiken von  $(L^{(n)}(t))_t$  in der Form

$$dL^{(n)}(t) = L^{(n)}(t) \sigma^{(n)}(t) dW^{(n)}(t) \quad \forall 0 \leq t \leq T_n,$$

wobei  $W^{(n)}$  die  $n$ -te Komponente eines Wiener-Prozesses bzgl.  $\mathbb{P}_{T_{n+1}}$  darstellt, angegeben werden.

Auf diese Weise gelangen wir zu derselben Problemstellung wie bei der Bewertung einer Call-Option in einem Black-Scholes Modell mit konstanter Zinsrate  $r=0$  und einer deterministischen Volatilitätsfunktion. Dies bedeutet, dass wir den Capletpreis mithilfe der bekannten Black's Formel

$$V_{\text{Caplet}}^{n,n+1}(t) = B(t, T_{n+1}) \delta [L^{(n)}(t) \Phi(h_1(L^{(n)}(t), t)) - K \Phi(h_2(L^{(n)}(t), t))] \quad (8.8)$$

mit

$$h_1(L^{(n)}(t), t) = (\ln(\frac{L^{(n)}(t)}{K}) + \frac{1}{2} \int_t^{T_n} \sigma^{(n)2}(s) ds) \left( \sqrt{\int_t^{T_n} \sigma^{(n)2}(s) ds} \right)^{-1}$$

und

$$h_2(L^{(n)}(t), t) = (\ln(\frac{L^{(n)}(t)}{K}) - \frac{1}{2} \int_t^{T_n} \sigma^{(n)2}(s) ds) \left( \sqrt{\int_t^{T_n} \sigma^{(n)2}(s) ds} \right)^{-1}$$

bestimmen können.

Für das Caplet mit einem Nominal von 1 Million Euro ergibt sich mithilfe von Black's Formel dann ein Preis in Höhe von  $V_{\text{Caplet}}(T_0) \approx 671.4936 \text{ €}$ , was mit dem obigen Preis bis auf einen Fehler in Höhe von  $7.265 \cdot 10^{-5} \%$  des Nominals übereinstimmt.

Wir sehen also, dass die Simulation tatsächlich sehr nah an den exakten Preis herankommen kann, je nachdem wie groß die Fehlertoleranz  $\epsilon$  gewählt wird.

### 8.3 Bewertung von CMS Caps und TARNs

In diesem Abschnitt wollen wir die Simulationspreise für einen CMS Cap und ein TARN Produkt angeben. Im Unterschied zum Caplet gibt es hier keine einfachen analytischen Bewertungsformeln. Genau aus diesem Grund müssen überhaupt numerische Näherungsverfahren für die Bewertung der Derivate herangezogen werden. Für die Bewertung der Produkte werden wir die Parameter möglichst praxisnah wählen, um annähernd realistische Ergebnisse zu erhalten.

Zunächst bewerten wir einen CMS Cap, dessen Laufzeit in  $T_a = 1$  beginnt und in  $T_b = 5$  endet. Dies bedeutet, dass die Indizes  $a$  und  $b$  bei einer Tenorintervalllänge von  $\delta = 0.5$  die Werte 2 bzw. 10 annehmen. Für die Laufzeit der „Constant-Maturity Swaps“ wählen wir  $m = 20$ , also ein Laufzeit der Swaps von 10 Jahren. Als Strike-Zinssatz wählen wir die 10-Jahres Swap-Rate  $K = 1.56\%$ . Die benötigten Startwerte der Libor-Raten  $L^{(a)}(T_0) = L^{(2)}(T_0)$  bis  $L^{(b-1+m-1)}(T_0) = L^{(28)}(T_0)$ , welche wie alle Marktdaten aus dem der Arbeit [San13] zugrunde liegenden Datensatz abgeleitet sind, entnehmen wir der nachfolgenden Tabelle. Der Bondpreis des benötigten  $T_{10}$ -Bonds beträgt  $B(T_0, T_{10}) = 0.9610\text{ €}$ .

$i$	$L^{(i)}(T_0)$	$i$	$L^{(i)}(T_0)$	$i$	$L^{(i)}(T_0)$
0	0.3220	14	2.3651	28	3.2827
1	0.3460	15	2.5363	29	3.3826
2	0.4203	16	2.5991	30	2.6301
3	0.5124	17	2.7607	31	2.6710
4	0.6367	18	2.7988	32	2.7122
5	0.7318	19	2.9505	33	2.7540
6	0.9724	20	2.9272	34	2.7962
7	1.1103	21	3.0647	35	2.8390
8	1.3726	22	3.0327	36	2.8823
9	1.5404	23	3.1577	37	2.9262
10	1.7631	24	2.8987	38	2.9706
11	1.9440	25	2.9925	39	3.0157
12	2.0893	26	3.0877		
13	2.2681	27	3.1844		

Tabelle 2: Aus dem zugrunde liegenden Datensatz vom 18.04.2013 abgeleitete Libor-Raten  $L^{(i)}(T_0)$  (in Prozent), welche der Arbeit [San13] entnommen wurden

Mit der Wahl  $N_{\text{Start}} = 500$  sowie einer Fehlertoleranz von  $\epsilon = 2 \cdot 10^{-4}$  und einem Nominal in Höhe von 1 Million Euro erhalten wir mithilfe von Multilevel Monte Carlo Simulationen für den CMS Cap dann einen Preis in Höhe von ungefähr  $V_{\text{CMS Cap}}^{2,10} \approx 28808.0398\text{ €}$ .

Die Wahl der initialen Anzahl an zu simulierenden Pfaden sowie der Fehlertoleranz richtet sich hier danach, dass zum einen der zeitliche Rechenaufwand nicht zu groß wird, damit das Verfahren auch für die Anwendung in der Praxis interessant bleibt, und zum anderen der Vorteil der Multilevel Verfahren in Bezug auf den verringerten Rechenaufwand bei vorgegebener Genauigkeit bzw. Fehlertoleranz zum Vorschein kommt. Wird  $N_{\text{Start}}$  in Bezug auf  $\epsilon$  zu groß gewählt, so werden bei jedem der höheren Level im Multilevel Verfahren jeweils immer  $N_{\text{Start}}$  Simulationen durchgeführt, obwohl die optimale Anzahl an Simulationen deutlich geringer sein kann. Dies lässt den Rechen- sowie Zeitaufwand des Multilevel Verfahrens im Vergleich zum Standard Monte Carlo Verfahren, bei dem auch mit einer kleineren Schrittweite der höheren Level viele Simulationen durchgeführt werden müssen, um die geforderte Genauigkeit zu erreichen, zu groß erscheinen. Um für diese

Zwecke passende Werte zu finden, sind verschiedene Parameter-Kombinationen getestet worden, wobei sich die hier angegeben Werte als geeignet herausgestellt haben.

Analog zu den Kriterien bei der Parameterwahl zur Bewertung der CMS Caps werden auch die Parameter zur Bewertung eines TARNs gesetzt. Wir wählen dazu ebenfalls eine Fehlertoleranz  $\epsilon = 2 \cdot 10^{-4}$ . Die initiale Anzahl an zu simulierenden Pfaden wird hier als  $N_{\text{Start}} = 1000$  gewählt.

Für die nachfolgende Bewertung wählen wir einen TARN mit Laufzeit von 5 Jahren, d.h. es werden  $b = 10$  sowie  $T_b = 5$  gesetzt. Die benötigten Libor-Raten Startwerte  $L^{(0)}(T_0)$  bis  $L^{(9)}(T_0)$  können wieder obiger Tabelle 2 entnommen werden. Der benötigte Bondpreis  $B(T_0, T_{10}) = 0.9610 \text{ €}$  wurde ebenso schon bei der Bewertung des CMS Caps verwendet. Als Zielkupon wählen wir eine Rendite von  $R = 10 \%$  und den Strike-Zinssatz des strukturierten Kupons als  $K = 2 \%$  sowie einen „gearing-Faktor“  $g = 2$ .

Bei der Wahl dieser Parameter und einem erneuten Nominal in Höhe von 1 Million Euro ergibt sich der Wert eines TARNs zu  $V_{\text{TARN}}^{10} \approx 13257.3437 \text{ €}$ .

## 8.4 Vergleich der numerischen Verfahren

Im Folgenden wollen wir nun den erforderlichen Rechenaufwand des Multilevel Monte Carlo (MLMC) Verfahrens und des Standard Monte Carlo (StdMC) Verfahrens vergleichen. Die Multilevel Monte Carlo Verfahren werden wir dabei zum einen unter Verwendung des Euler-Maruyama Diskretisierungsverfahrens, zum anderen unter Verwendung des Milstein Diskretisierungsverfahrens betrachten.

Für die Bestimmung der Standard Monte Carlo Preise nutzen wir den Schätzer

$$\frac{1}{\tilde{N}_L} \sum_{j=1}^{\tilde{N}_L} \tilde{P}_L^{(j)},$$

wobei  $\tilde{P}_L^{(j)}$  für alle  $j$  einen simulierten Payoff darstellt. Zu klären ist dabei, für welchen Level  $L$ , d.h. bezüglich welcher konstanten Diskretisierungsschrittweite  $h_L$ , die Pfade simuliert werden sollen, mit denen dann die Preise der Derivate bestimmt werden können.

Um den Bias abschätzen und damit dieselbe Konvergenzbedingung wie beim Multilevel Monte Carlo Verfahren nutzen zu können, müssen wir allerdings auch hier Simulationen für verschiedene Diskretisierungslevel durchführen. Analog zum Multilevel Verfahren beginnen wir folglich mit den Pfadsimulationen des Levels 0 und erhöhen den Diskretisierungslevel  $L$  solange, bis der Bias ausreichend klein ist, was durch den Konvergenztest

$$\max\{|E_L|, M^{-1}|E_{L-1}|\} < \frac{1}{\sqrt{2}}(M-1)\epsilon \quad (\text{vgl. (3.22)})$$

überprüft werden kann. Zu bemerken ist allerdings, dass für die Preisschätzung selbst nur die bzgl. des feinsten Levels  $L$  simulierten Pfade genutzt werden und wir

die Simulationen für die vorherigen Level durchführen müssen, damit der Bias des Schätzers approximiert und durch die Nutzung desselben Konvergenzkriteriums Vergleichbarkeit der Verfahren hergestellt werden kann.

Des Weiteren muss auch für den Standard Monte Carlo Schätzer die Varianz ausreichend klein gehalten werden. Dazu müssen wir für jeden Level  $l$  je

$$\tilde{N}_l = \lceil 2\epsilon^{-2}\mathbb{V}[\tilde{P}_l] \rceil \quad (8.9)$$

Pfade simulieren, denn auf diese Weise besitzt der Monte Carlo Schätzer die Varianz

$$\mathbb{V}[\hat{Y}] = \mathbb{V}\left[\frac{1}{\tilde{N}_L} \sum_{j=1}^{\tilde{N}_L} \tilde{P}_L^{(j)}\right] = \frac{1}{\tilde{N}_L} \mathbb{V}[\tilde{P}_L] \stackrel{(8.9)}{\leq} \frac{1}{2}\epsilon^2,$$

was wir für die Einhaltung der vorgegebenen Fehlerschranke benötigen. Bei der Simulation wird die unbekannte Varianz  $\mathbb{V}[\tilde{P}_L]$  dazu natürlich wieder mithilfe der Stichprobenvarianz der simulierten Samples approximiert.

Im Folgenden wollen wir nun die Konvergenzraten der verschiedenen Verfahren zum einen anhand der Stichprobenvarianzen  $V_l$  der Samplepreise bzw. der Multilevel-Preisdifferenzen, zum anderen anhand des Durchschnitts der Samplepreise bzw. der Multilevel-Preisdifferenzen vergleichen und numerisch zeigen, dass die theoretisch hergeleiteten Konvergenzraten erreicht werden können. Ferner werden wir den numerischen sowie zeitlichen Rechenaufwand der drei verschiedenen Verfahren vergleichen.

Beginnen wollen wir mit der Konvergenz der Verfahren. Um die Konvergenzraten besser ableiten zu können, haben wir bei der Simulation die Durchführung der Algorithmen bis zum Level 4 erzwungen und die Abbruchbedingung zu diesem Zweck nicht beachtet. In der Abbildung 4 werden die erhaltenen Ergebnisse einmal für den CMS CAP und einmal für den TARN dargestellt.

In den jeweils linken Plots der Abbildung ist – wie in [Gil08a] – der Logarithmus zur Basis  $M$  der Stichprobenvarianz  $V_l$  der Samplepreise bzw. Preisdifferenzen gegen den Level  $l$  aufgetragen.

Doch wie lassen sich die dargestellten Ergebnisse interpretieren? In Gleichung (3.6) aus Kapitel 3.2 haben wir gesehen, dass die Varianz  $V_l$  mit der Ordnung  $\mathcal{O}(h_l)$  konvergiert, falls beim Multilevel Verfahren eine starke Konvergenzrate 1/2 des Diskretisierungsverfahrens vorliegt. Dies ist, wie wir gesehen haben, unter Verwendung des Euler-Maruyama Verfahrens der Fall. Die  $\mathcal{O}(h_l)$ -Konvergenz ist nun äquivalent dazu, dass die Gleichung

$$\begin{aligned} \log_M(V_l) &\approx \log_M(ch_l) = \log_M(cM^{-l}T) \\ &= \log_M(M^{-l}) + \log_M(cT) = -l + \log_M(cT) \end{aligned} \quad (8.10)$$

erfüllt ist. Mit der Äquivalenz von Gleichung (8.10) zur Gleichung

$$\frac{\partial \log_M(V_l)}{\partial l} = -1$$

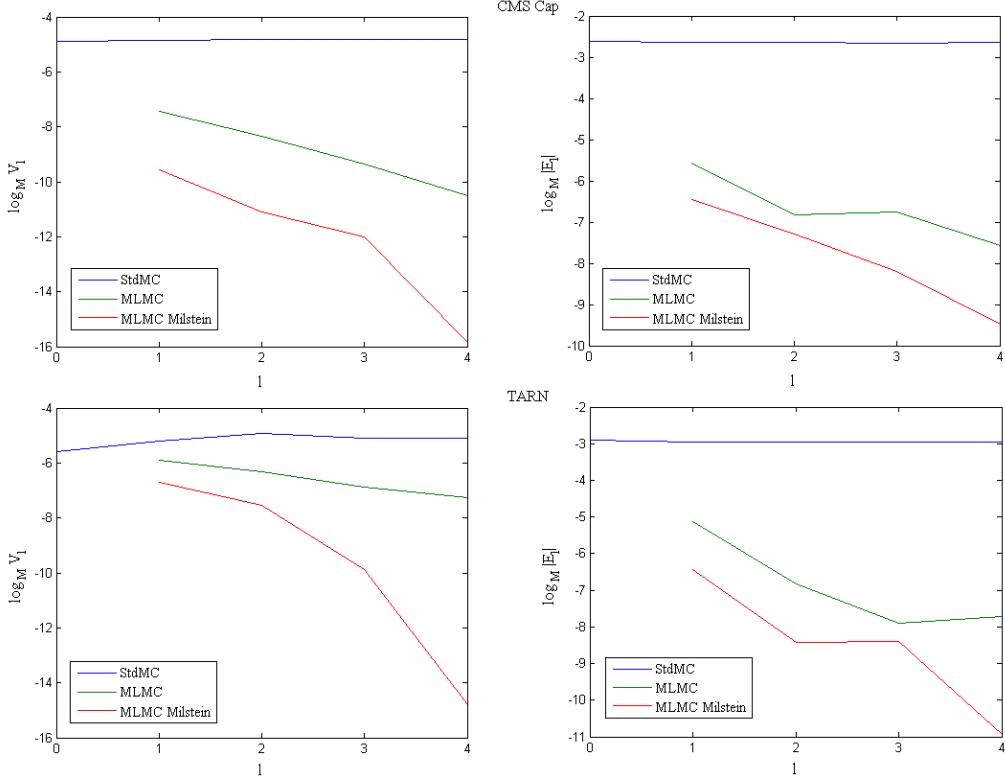


Abbildung 4: Stichprobenvarianzen und durchschnittliche Preisdifferenzen für die verschiedenen Level bei der Bewertung von CMS Caps (oben) und TARNs (unten)

sollten wir in den linken der beiden Plots beim Multilevel Verfahren mit einer Euler-Maruyama Diskretisierung also eine ungefähre Steigung von  $-1$  erwarten. Dies führt dazu, dass wir für eine leichtere Beobachtung der Konvergenzrate in der Abbildung 4 gerade den Logarithmus zur Basis  $M$  verwenden.

Das Milstein Diskretisierungsverfahren besitzt eine starke Konvergenzrate von  $1$ , was theoretisch zu einer Konvergenzordnung  $\mathcal{O}(h_l^2)$  der Varianz  $V_l$  führt. Dies bedeutet, dass in den beiden jeweils linken Plots für das Multilevel Verfahren mit Milstein Diskretisierung eine ungefähre Steigung von  $-2$  erkennbar sein sollte.

Bei einem Blick auf die beiden Plots lässt sich dieses Verhalten tatsächlich zumindest erahnen. Klar wird hingegen, dass die Varianz  $V_l$  der Korrekturterme für das Multilevel Verfahren unter Benutzung einer Milstein Diskretisierung deutlich schneller sinkt als die des Multilevel Verfahrens unter Verwendung einer Euler-Maruyama Diskretisierung.

Neben der Beobachtung der Konvergenzraten der verwendeten Diskretisierungsverfahren können wir auch den varianzreduzierenden Effekt des Multilevel Verfahrens erkennen. Dieser kommt dadurch zustande, dass die Stichprobenvarianz der Preisdifferenzen, welche den ursprünglich bestimmten Preis immer weiter korrigieren, im Vergleich zur konstant bleibenden Varianz des Standard Monte Carlo

Schätzers für höhere Level immer kleiner wird. Dass die Steigungen in beiden Fällen nicht ganz die theoretisch vorhergesagten Werte erreichen, liegt zum einen an der Abweichung, die generell bei numerischen Verfahren vorzufinden ist, wodurch immer Ausreißer oder Ähnliches zustande kommen können, zum anderen daran, dass die Konvergenzrate theoretisch auch nur etwas über Grenzwerte aussagt. In der Theorie wird nur eine Aussage für sehr große Level  $l$  getroffen. Darüber hinaus kann auch die Konstante in der Definition der Konvergenzrate zu einer leicht veränderten Rate bzw. Steigung führen.

In den jeweils rechten Plots der Abbildung 4 ist für das Standard Monte Carlo Verfahren der Logarithmus zur Basis  $M$  des Durchschnitts aller simulierten Samplepreise des jeweiligen Levels gegen den Level  $l$  aufgetragen. Für die beiden Multilevel Monte Carlo Verfahren sehen wir den Logarithmus zur Basis  $M$  des Durchschnitts aller simulierten Korrekturterme bzw. Preisdifferenzen in Abhängigkeit des jeweiligen Levels  $l > 0$ . In diesen Plots impliziert eine approximative Steigung von  $-1$  die Konvergenzordnung  $\mathcal{O}(h_l)$  des Erwartungswertes  $\mathbb{E}[\tilde{P}_l - \tilde{P}_{l-1}]$ . Diese lässt sich durchaus erkennen, wenn wir uns die jeweiligen Geraden ein wenig um Ausreißer bereinigt vorstellen.

Nachdem wir nun die behauptete Konvergenz der Monte Carlo Verfahren numerisch bekräftigt haben, wollen wir nun zum wichtigsten Gütekriterium der Verfahren kommen. Das vorrangige Ziel der Arbeit besteht schließlich darin, den Rechenaufwand des klassischen Monte Carlo Verfahrens zu verringern.

Bevor wir die zugehörigen Ergebnisse betrachten, wollen wir aber noch darauf eingehen, wie genau sich der numerische Rechenaufwand für die Bewertung von CMS Caps und TARNs zusammensetzt.

Zu bemerken ist zunächst, dass bei der Bestimmung des Rechenaufwands die Anzahl der für die Preisberechnung erforderlichen Libor-Raten keine Rolle spielt. Die Simulation je eines Vektors von Wiener-Inkrementen wird als eine Einheit Aufwand gewertet. Alternativ könnte man jede dieser Einheiten selbstverständlich auch mit der Länge des jeweiligen Vektors, also der Anzahl der Libor-Raten, welche simuliert werden müssen, multiplizieren. Dies würde die Zahlen jedoch nur weiter vergrößern und die Vergleichbarkeit der Verfahren untereinander nicht beeinflussen.

Der Rechenaufwand für das wie oben definierte Standard Monte Carlo Verfahren ergibt sich für einen CMS Cap damit zu

$$C_{\text{StdMC}}^{\text{CMS Cap}} = \sum_{l=0}^L (b-1) \tilde{N}_l M^l,$$

da für die je  $b-1$  Tenorintervalle  $[0, \delta], \dots, [(b-2)\delta, (b-1)\delta]$  jeweils  $\tilde{N}_l$  diskrete Libor-Raten-Pfade mit  $M^l$  Diskretisierungsstellen simuliert werden müssen. Für das Multilevel Verfahren wird in analoger Weise ein Rechenaufwand in der

Höhe

$$C_{\text{MLMC}}^{\text{CMS Cap}} = (b - 1)N_0^* + \sum_{l=1}^L (b - 1)N_l^*(M^l + M^{l-1})$$

benötigt, welcher nach Definition des numerischen Rechenaufwands für die Milstein-Variante ebenso zu berechnen ist. Die Summe  $M^l + M^{l-1}$  kommt dabei dadurch zustande, dass die Simulation des Pfades bzgl. der feineren Schrittweite für das l-te Level  $M^l$  Diskretisierungsschritte pro Tenorintervall erfordert und für die Simulation des Pfades bzgl. der größeren Schrittweite  $M^{l-1}$  Diskretisierungsschritte pro Tenorintervall benötigt werden.

In Abbildung 5 können wir sehen, dass bei der Bewertung beider Produkte der numerische sowie auch der zeitliche Rechenaufwand des Multilevel Monte Carlo Verfahrens im Vergleich zum Standard Monte Carlo Verfahren tatsächlich deutlich verringert wird. Ebenso ist zu sehen, dass auch der Einsatz des Milstein Diskretisierungsverfahrens statt des Euler-Maruyama Diskretisierungsverfahrens den Rechenaufwand noch einmal deutlich verringern kann, wie es die theoretischen Ergebnisse suggeriert haben.

Bei der Bewertung des CMS Caps wird der numerische Rechenaufwand bei der vorliegenden Simulation durch den Einsatz des Multilevel Verfahrens zusammen mit der Milstein Diskretisierung im Vergleich zum Standard Monte Carlo Verfahren ungefähr um den Faktor 16 verringert. Der zeitliche Aufwand hat sich von ca. 3000 Sekunden auf ca. 440 Sekunden, also immerhin fast um den Faktor 7, verringert. Bei der Bewertung des TARNs ergibt sich eine Verringerung des numerischen Rechenaufwands in etwa um den Faktor 37 sowie des zeitlichen Aufwands hier ebenfalls ungefähr um den Faktor 7.

Nachdem wir die zentralen numerischen Ergebnisse nun angegeben und erläutert haben, wollen wir noch auf einige Details der Ergebnisse eingehen.

Besonders auffällig bei der Erstellung der Ergebnisse ist gewesen, dass die Vorteile des Multilevel Verfahrens erst oder besonders bei der Simulation der höheren Level zum Vorschein kommen, was vor allem bei der Betrachtung der Plots, die den Rechenaufwand der Verfahren angeben, zum Vorschein kommt (vgl. Abbildung 5). Dies bedeutet im Umkehrschluss auch, dass das Multilevel Verfahren bei niedriger geforderter Genauigkeit, welche nur eine geringere Anzahl an zu simulierenden Leveln erfordert, im Vergleich zum Standard Monte Carlo Verfahren keine Vorteile bringt oder sogar einen höheren Zeitaufwand benötigt. Je mehr Level jedoch simuliert werden müssen, desto vorteilhafter wird auch der Einsatz des Multilevel Verfahrens. Dies liegt daran, dass beim Multilevel Verfahren für die niedrigen Level vergleichsweise sehr viele Pfade simuliert werden müssen, für die höheren Level jedoch stetig deutlich weniger Samplepfade benötigt werden, welche mit hohem Zeitaufwand simuliert werden müssen. So werden bei der Bewertung des TARNs für den 0-ten Level in obiger Simulation beispielsweise noch  $N_0^* = 179625$  Samplepfade benötigt, für die nachfolgenden Level sind dann jedoch nur noch  $N_1^* = 11115$ ,  $N_2^* = 1755$  und letztlich sogar nur  $N_3^* = 180$  Pfade zu simulieren gewesen.

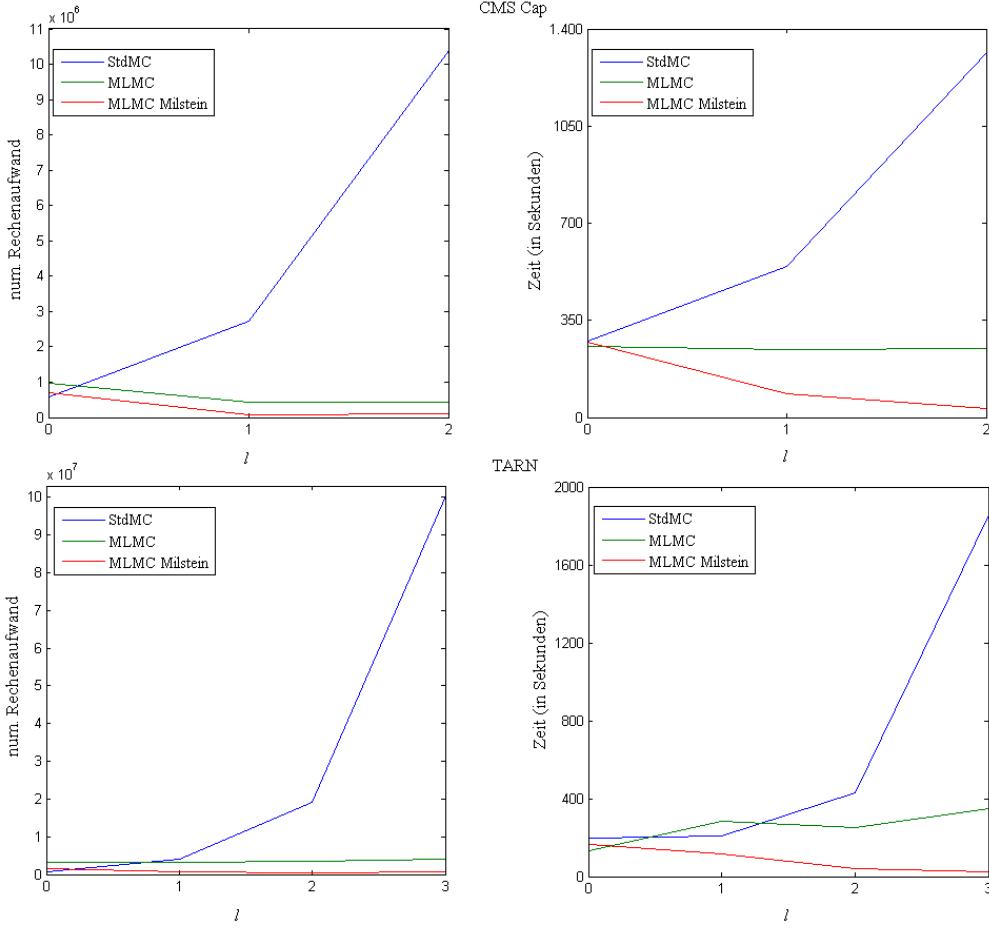


Abbildung 5: Numerischer sowie zeitlicher Rechenaufwand der Monte Carlo Verfahren bei der Bewertung von CMS Caps (oben) und TARNs (unten) für die verschiedenen Level

Eine weitere Auffälligkeit, die bei der Erstellung der numerischen Ergebnisse zum Vorschein gekommen ist, ist, dass das Multilevel Verfahren einen höheren Aufwand benötigen kann als das Standard Monte Carlo Verfahren, falls die Payoff-Funktion schon von sich aus eine sehr geringe Varianz aufweist. Dies ist beispielsweise der Fall, wenn bei einem TARN ein geringer Zielkupon  $R$  geboten wird, welcher bei der Simulation der Libor-Raten-Pfade folglich mit sehr hoher Wahrscheinlichkeit erreicht wird, was zur vorzeitigen Beendigung des TARNs führt. Dies liefert dann in sehr vielen Fällen einen relativ ähnlichen Samplepreis, was zu einer geringen Stichprobenvarianz führt. Wird im obigen Anwendungsfall der Zielkupon zum Beispiel als  $R = 5\%$  gewählt, so können wir sehen, dass der zeitliche Rechenaufwand des Multilevel Verfahrens bei gleicher Genauigkeit  $\epsilon$  tatsächlich höher ist als der des Standard Monte Carlo Verfahrens. Erhöhen wir zusätzlich die Fehlerschranke  $\epsilon$  leicht auf  $\epsilon = 5 \cdot 10^{-4}$  und belassen alle anderen Parameter bis auf den Zielkupon wie oben bei der Bewertung des TARNS, so wird für das Standard Monte Carlo Verfahren in etwa eine Zeit von 95 Sekunden, für das Multilevel Monte Carlo Ver-

fahren unter Benutzung des Euler-Maruyama Diskretisierungsverfahrens jedoch eine Zeit von ungefähr 140 Sekunden benötigt. Dabei wird jeweils bis einschließlich zum zweiten Level simuliert.

Abschließend können wir jedoch festhalten, dass sich die im Theorie-Teil prognostizierten Verbesserungen im Vergleich zum Standard Monte Carlo Verfahren in vielen Fällen durchaus auch in der Anwendung bestätigen lassen, sofern ein gewisser Grad an Genauigkeit erforderlich ist, was in der Praxis sicher meist der Fall ist, wenn Zinsderivate bewertet werden sollen.

## 9 Fazit und Ausblick

Wie wir in dieser Arbeit, vor allem im vorangehenden Kapitel, gesehen haben, lässt sich der zur Bewertung von Zinsderivaten im Libor Marktmodell notwendige Rechenaufwand durch den Einsatz eines Multilevel Monte Carlo Verfahrens im Vergleich zur Standard Monte Carlo Methode deutlich verringern. Darüber hinaus haben wir gesehen, dass der Einsatz eines Diskretisierungsverfahrens höherer Ordnung, nämlich des Milstein Verfahrens, den Rechenaufwand – im Vergleich zum Euler-Maruyama Diskretisierungsverfahren – bei gleicher Genauigkeit noch weiter verringert. Dabei sichert uns der zentrale Satz dieser Arbeit theoretisch zu, dass das Multilevel Verfahren im Allgemeinen anwendbar ist und die gewünschte Verringerung des Rechenaufwands erzielt werden kann. Bei der Darstellung der numerischen Ergebnisse haben wir sogar gesehen, dass auch in Fällen, in denen nicht alle Voraussetzungen des Satzes erfüllt sind, deutliche Einsparungen beim Rechenaufwand erreicht werden können, wenn das Multilevel Verfahren eingesetzt wird.

Ausgehend von dieser Grundlage entstehen viele weitere interessante Ansatzpunkte, welche weiter untersucht werden könnten.

Ein mögliches Thema für weitere Untersuchungen ist beispielsweise die Analyse, inwieweit sich die Ergebnisse auf Derivate mit Payoff-Funktionen, welche Unstetigkeitsstellen besitzen, übertragen lassen. Im Libor Marktmodell wären dabei z.B. TARNs in der Version als „Notes“ oder auch sogenannte „Autocallables“, welche eine Art von exotischen Barriere-Optionen auf Bonds darstellen, interessante Derivate.

Eine andere äußerst interessante Art von Derivaten, welche wir bislang nicht betrachtet haben, sind sicher auch pfadabhängige Derivate wie z.B. Bermuda Swaptions oder Asiatische und Amerikanische Optionen auf Libor-/Euribor-Zinssätze. In [GDR13] haben Giles et al. beispielsweise die Verwendung des Milstein Diskretisierungsverfahrens u.a. für pfadabhängige Asiatische oder Lookback-Optionen modellunabhängig im Kontext von Multilevel Monte Carlo Verfahren untersucht. Dabei ist gezeigt worden, dass durch die Verwendung des Milstein Verfahrens zusammen mit der Verwendung einer sogenannten „Brownschen Interpolation“ ähnliche Ergebnisse wie bei der Betrachtung von Lipschitz-stetigen Payoff-Funktion erzielt werden können.

Ein weiteres, interessantes zu untersuchendes Themengebiet ist die Betrachtung von Derivaten in anderen Modellen, in denen bei der Simulation die Verwendung der sogenannten „Lévy areas“ notwendig ist. In dieser Arbeit haben wir vorgestellt, wie wir in solchen Fällen auch ohne eine aufwändige Simulation dieser „Lévy areas“ mithilfe der „antithetischen Variablen“ einen präzisen Multilevel Monte Carlo Schätzer erhalten können. Dies könnte dann z.B. in einem Libor Marktmodell mit stochastischer Volatilität angewendet werden.

Ferner ist es sicherlich lohnend, weitere Varianzreduktionsmethoden zusammen mit dem Multilevel Monte Carlo Verfahren zu testen. Dabei ist zu untersuchen, ob auch bei Verwendung des Multilevel Verfahrens ähnliche (zusätzliche) Verbes-

---

serungen, die sich im Vergleich der Standard Monte Carlo Methode mit varianzreduzierten Monte Carlo Methoden herausstellen, zu erreichen sind.

Außerdem könnte, wie es häufig in der Literatur (siehe z.B. [Fri07]) zu finden ist, bei der Marktkalibrierung eine Rangreduzierung des zugrunde liegenden Wiener-Prozesses durchgeführt und gleichzeitig die Parameter im Volatilitätsterm der dem Libor Marktmodell zugrunde liegenden SDE dementsprechend angepasst werden. Dies ließe sich zusammen mit den vorgestellten Verfahren umsetzen und könnte den Rechenaufwand, der für die Erzeugung der Wiener-Prozesse notwendig ist, noch zusätzlich verringern.

## A Anhang

### A.1 Marktdaten

An dieser Stelle wollen wir zuletzt die für die Kalibrierung erforderlichen Marktdaten, welche aus einem Datensatz vom 18.04.2013 stammen und aus der Arbeit [San13] übernommen werden, in den nachfolgenden Tabellen angegeben. Es handelt sich dabei um die für die Bestimmung der aktuellen Bondpreise und der aktuellen Libor-Raten  $L(T_0)$  notwendigen Swap-Raten, sowie die für die Bestimmung der Parameter in den Korrelations- und Volatilitätsfunktionen des Modells notwendigen (impliziten) Cap- bzw. Caplet- und Swaption-Volatilitäten.

#### Bemerkung A.1

Bei den vorliegenden Daten handelt es sich um einen Datensatz, der auf Euribor-Zinssätzen basiert. In der Theorie macht es jedoch keinen Unterschied, ob wir mit Libor- oder Euribor-Zinssätzen arbeiten. Üblicherweise wird deshalb immer von Libor-Zinssätzen gesprochen, weshalb wir diese Bezeichnung auch in dieser Arbeit verwendet haben.

$b$	$R_{\text{Swap}}^{0,b}(T_0)$	$b$	$R_{\text{Swap}}^{0,b}(T_0)$
1	0.3220	14	1.1290
2	0.3340	16	1.2845
3	0.3627	18	1.4280
4	0.4000	20	1.5595
6	0.4942	22	1.6760
8	0.6294	24	1.7795
10	0.7915	30	2.0108
12	0.9630	40	2.1735

Tabelle 3: Swap-Raten  $R_{\text{Swap}}^{0,b}(T_0)$  (in Prozent) vom 18.04.2013

$b$	$\sigma_{1,b}^{\text{CapM}}$	$b$	$\sigma_{1,b}^{\text{CapM}}$
2	108.70	16	49.40
4	81.44	18	45.15
6	81.89	20	41.70
8	74.58	24	37.45
10	68.54	30	32.75
12	61.70	40	29.70
14	54.40		

Tabelle 4: (Implizite) „at-the-money“ Cap-Volatilitäten  $\sigma_{1,b}^{\text{CapM}}$  (in Prozent) vom 18.04.2013 für Caps mit Laufzeit über das Intervall  $[T_1, T_b]$

$i$	$\sigma_i^{\text{CM}}$	$i$	$\sigma_i^{\text{CM}}$	$i$	$\sigma_i^{\text{CM}}$
1	108.70	14	35.15	27	22.67
2	75.58	15	34.79	28	21.95
3	72.81	16	31.66	29	21.45
4	81.62	17	29.16	30	19.85
5	84.22	18	27.80	31	19.92
6	71.56	19	27.67	32	20.28
7	60.66	20	27.98	33	20.95
8	60.19	21	28.52	34	21.97
9	58.51	22	28.31	35	23.38
10	51.99	23	27.77	36	25.20
11	44.39	24	25.91	37	27.51
12	37.73	25	24.68	38	30.36
13	34.59	26	23.59	39	33.88

Tabelle 5: (Implizite) „at-the-money“ Caplet-Volatilitäten  $\sigma_i^{\text{CM}}$  (in Prozent) vom 18.04.2013

$a \setminus j$	2	4	6	8	10	12	14	16	18	20
1	94.66	85.90	76.30	65.80	56.60	48.90	43.60	39.80	37.10	35.10
2	75.82	75.60	68.80	59.90	52.75	46.95	42.90	39.90	37.80	36.00
4	77.27	66.10	58.00	50.05	46.25	42.15	39.25	37.40	35.30	34.05
6	63.68	55.05	48.55	44.00	40.85	37.85	36.20	33.90	32.60	31.65
8	50.92	45.40	40.35	37.60	35.50	33.45	31.75	30.75	29.95	29.35
10	41.44	37.80	35.20	33.25	31.70	30.40	29.45	28.70	28.15	27.75
12	35.95	33.09	31.29	29.90	28.91	27.98	27.37	26.90	26.54	26.39
14	30.92	28.75	27.65	26.80	26.30	25.75	25.40	25.20	25.10	25.10
16	28.73	27.08	26.19	25.53	25.51	24.81	24.55	24.56	24.82	24.64
18	26.34	25.34	24.62	24.28	24.63	23.77	23.81	23.92	24.54	24.17
20	24.07	23.55	23.20	23.00	23.70	23.00	23.15	23.35	24.30	23.75

Tabelle 6: (Implizite) „at-the-money“ Swaption-Volatilitäten  $\sigma_{a,a+j}^{\text{SM}}$  (in Prozent) vom 18.04.2013 für Swaptions mit Fälligkeit der Option in  $a$  halben Jahren ( $\delta = 0.5$ ) und Laufzeit der Swaps von  $j$  halben Jahren

## Literatur

- [Als07] Gerold Alsmeyer. Wahrscheinlichkeitstheorie. *Skripten zur Mathematischen Statistik*, Nr. 30, 5. Auflage, 2007.
- [AP10] Leif B. G. Andersen, Vladimir V. Piterbarg. Interest Rate Modelling Volume I: Foundations and Vanilla Models. *Atlantic Financial Press*, 2010.
- [BM06] Damiano Brigo, Fabio Mercurio. Interest Rate Models - Theory and Practice. With Smile, Inflation and Credit. *Springer Finance*, 2nd Edition, 2006.
- [Fri07] Christian Fries. Mathematical Finance. Theory, Modeling, Implementation. *John Wiley & Sons*, 2007.
- [GDR13] Michael B. Giles, Kristian Debrabant, Andreas Rößler. Numerical analysis of multilevel Monte Carlo path simulation using the Milstein discretisation. <http://arxiv.org/pdf/1302.4676.pdf>, 2013.
- [GHM09] Michael B. Giles, Desmond J. Higham, Xuerong Mao. Analysing Multilevel Monte Carlo for Options with Non-globally Lipschitz Payoff. *Finance and Stochastics*, 13 (3), pp. 403-413, 2009.
- [Gil08a] Michael B. Giles. Multilevel Monte Carlo Path Simulation. *Operations Research*, Vol. 56, No. 3, May-June 2008, pp. 607-617, 2008.
- [Gil08b] Michael B. Giles. Improved multilevel Monte Carlo convergence using the Milstein scheme. In: A. Keller, S. Heinrich, H. Niederreiter (eds.) *Monte Carlo and Quasi-Monte Carlo Methods 2006*, pp. 343-358, Springer, 2008.
- [Gla03] Paul Glasserman. Monte Carlo Methods in Financial Engineering. *Springer*, Applications of Mathematics ; 53, 2003.
- [GS12] Michael B. Giles, Lukasz Szpruch. Antithetic multilevel Monte Carlo estimation for multi-dimensional SDEs without Lévy area simulation. <http://arxiv.org/pdf/1202.6283.pdf>, 2012.
- [Has12] Katharina Hasow. Modellierung der Volatilitätsstruktur in LIBOR-Markt-Modellen. *Masterarbeit*, Mathematisches Institut für Statistik, WWU Münster, 2012.
- [KP92] Peter E. Kloeden, Eckhard Platen. Numerical Solution of Stochastic Differential Equations. *Springer*, Applications of Mathematics; 23, 1992.
- [KS97] Ioannis Karatzas, Steven E. Shreve. Brownian Motion and Stochastic Calculus. *Springer*, Graduate Texts in Mathematics; 113, 2nd edition, 1997.
- [Mül02] Thomas Müller-Gronbach. Strong approximation of systems of stochastic differential equations. *Habilitation thesis*, TU Darmstadt, 2002.
- [Pit06] Vladimir V. Piterbarg. TARNs: Models, Valuation, Risk Sensitivities. In: Paul Wilmott. *The best of Wilmott 2*, chapter 14, John Wiley & Sons, 2006.

- [San13] Erik Santen. Kalibrierung eines Libor-Markt-Modells. *Masterarbeit*, Mathematisches Institut für Statistik, WWU Münster, 2013.
- [SC00] John Schoenmakers, Brian Coffey. Stable implied calibration of a multi-factor LIBOR model via a semi-parametric correlation structure. *Preprint*, Weierstraß-Institut für Angewandte Analysis und Stochastik im Forschungsverbund Berlin e.V.; 611, 2000.
- [Sch11] Daniel Schlotmann. Interpolation im Rahmen des LIBOR-Markt-Modells Eine Analyse ausgewählter Ansätze. *Diplomarbeit*, Mathematisches Institut für Statistik, WWU Münster, 2011.
- [Shr04] Steven E. Shreve. Stochastic Calculus for Finance II. Continuous-Time Models. *Springer Finance*, 2004.
- [Wik01] Magnus Wiktorsson. Joint characteristic function and simultaneous simulation of iterated Itô integrals for multiple independent Brownian motions. *The Annals of Applied Probability*, Vol. 11, No. 2, 470-487, 2001.