

Multilevel Monte-Carlo Simulationsverfahren mit Anwendung auf Multi-Asset Optionen

Bachelorarbeit

14. März 2011

eingereicht bei

Prof. Dr. Thomas Gerstner
Fachbereich Informatik und Mathematik
Institut für Mathematik
Goethe Universität
Frankfurt am Main

vorgelegt von

Marco Noll
Geb. am: 03.10.1986
Matrikelnummer: 3803372
Studienrichtung: Mathematik

Frankfurt am Main, den 14. März 2011

Inhaltsverzeichnis

Abbildungsverzeichnis	II
Ehrenwörtliche Erklärung	III
1 Einleitung	1
2 Grundlagen	3
2.1 Stochastische Numerik	3
2.2 Grundlagen der Optionsbewertung	4
2.3 Multi-Asset Optionen	5
3 Multilevel Monte-Carlo Simulationsverfahren	7
3.1 Monte-Carlo Methode	7
3.2 Multilevel Monte-Carlo Methode	8
3.3 Komplexitätstheorem	13
3.4 Erweiterungen	21
3.4.1 Optimales M	21
3.4.2 Fehlerschätzung	24
3.4.3 Richardson Extrapolation	25
4 Implementierung	26
5 Numerische Ergebnisse	29
5.1 Europäische Option	29
5.2 Asiatische Option	31
5.3 Multi-Asset Optionen	33
5.3.1 Geometrischer Basket	33
5.3.2 Arithmetischer Basket	34
5.4 Heston stochastic volatility Model	35
6 Schlussbetrachtungen und Ausblick	38
A Anhang: Programmcodes	40
Literaturverzeichnis	45

Abbildungsverzeichnis

1	Optimales M (geometrisches Mittel)	22
2	Optimales M (arithmetisches Mittel)	23
3	Europäische Option	30
4	Asiatische Option	32
5	Geometrischer Basket	33
6	Arithmetischer Basket	35
7	Europäische Option mit Heston Model	37

Ehrenwörtliche Erklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbständig abgefaßt und keine anderen Hilfsmittel als die angegebenen benutzt habe. Ich erkläre ferner, dass diejenigen Stellen der Arbeit, die anderen Werken wörtlich oder dem Sinn nach entnommen sind, in jedem einzelnen Fall unter Angabe der Quellen kenntlich gemacht sind.

Frankfurt am Main, den 14. März 2011

1 Einleitung

Auf den internationalen Finanzmärkten werden viele verschiedene Optionen gehandelt, für die keine geschlossenen Bewertungsformeln existieren. Um den Wert dieser Optionen zu ermitteln werden daher numerische Verfahren eingesetzt. Dabei treten neue Probleme auf. Der approximierte Optionswert soll möglichst genau und die Rechenkosten niedrig sein. Ein weitverbreitetes Verfahren ist die Monte Carlo (MC) Simulation. Dabei entspricht der approximierte Optionswert dem arithmetischen Mittel aus mehreren unabhängigen Simulationen des Optionswertes. Die Genauigkeit wird erhöht indem die Anzahl der Simulationen erhöht wird. Allerdings steigen damit auch die Rechenkosten.

Wie kann aber die Genauigkeit verbessert werden, ohne die Rechenkosten zu erhöhen?

Indem Multi-Grid Methoden verwendet werden. Das Prinzip der Multi-Grid Methode wird bei der Multilevel Monte-Carlo Methode (MLMC) angewandt. Dabei wird der Aktienkurs mit unterschiedlichen Schrittweiten simuliert und daraus der Optionswert ermittelt. Somit wird, bei bedeutend geringerem Rechenaufwand, die Genauigkeit des feinsten Diskretisierungslevels erreicht. Um einen Vergleich der MLMC Methode mit der Monte-Carlo Methode zu ziehen, wird bei beiden Verfahren die gleiche Genauigkeit verlangt, welche sich aus dem mean square error (MSE) ermittelt und die Rechenkosten verglichen. Es wird gezeigt, dass im einfachsten Fall bei Verwendung einer Lipschitzstetigen Auszahlungsfunktion und des Euler-Maruyama Verfahrens zur Diskretisierung des Aktienkurses, die Rechenkosten von $O(\epsilon^{-3})$ auf $O(\epsilon^{-2}(\log \epsilon)^2)$ bei einem root mean square error (RMSE) von $O(\epsilon)$ sinken. Das Ziel dieser Arbeit ist es schließlich, Multi-Asset Optionen mit beiden Verfahren numerisch zu bewerten und zu testen, ob die MLMC Methode den Rechenaufwand gegenüber der Monte-Carlo Methode reduziert.

Bei der Auswahl dieses Themas empfand ich neben den vielfältigen Möglichkeiten der MLMC Methode auch die Aktualität dieses Themas besonders interessant. Der erste Artikel speziell zu diesem Thema wurde erst 2008 veröffentlicht und aktuell wird nach weiteren Möglichkeiten geforscht die Rechenkosten, aufbauend auf diesem Verfahren, zu reduzieren. So kann die MLMC Methode z.B. mit der Quasi-Monte-Carlo Methode kombiniert werden um weitere Rechenkosten einzusparen. In Vorlesungen zur Finanznumerik und im speziellen zum Monte-Carlo Verfahren wurde klar, wie wichtig die Effizienz von Algorithmen ist und welche Bedeutung diesen in der Finanzmathematik zukommt. In Zukunft wird es spannend sein zu sehen, ob diese Methode auch in der Praxis der Optionsbewertung Anwendung findet und ob sich durch die vielfältige Einsatzmöglich-

keiten von Monte-Carlo Methoden weitere Anwendungsgebiete ergeben.

Anfangs werden in Kapitel 2 die wichtigsten Grundlagen zur numerischen Simulation von Aktienkursen und zur Optionsbewertung gegeben. In Kapitel 3 wird das eigentliche Thema dieser Arbeit behandelt. Zuerst wird die benötigte Anzahl der Simulationen und die erforderliche Schrittweite abgeschätzt, um einen MSE der Größe $O(\epsilon^2)$ für die Monte-Carlo Methode zu erhalten. Anschließend wird erklärt wie das Multilevel Monte Carlo Verfahren funktioniert und wie Rechenkosten gegenüber der MC Methode eingespart werden können. Dabei wird stets eine lipschitzstetige Auszahlungsfunktion vorausgesetzt und das Euler-Maruyama Verfahren zur Diskretisierung des Aktienkurses verwendet. Danach wird im Komplexitätstheorem gezeigt, wie sich der Rechenaufwand bei Verwendung eines beliebigen Diskretisierungsverfahrens und beliebiger Auszahlungsfunktion verhält. Die Schrittweite zu der der Aktienkurs simuliert wird, ist durch einen Parameter M festgelegt. Die Wahl dieses Parameters beeinflusst die Rechenkosten, sodass sich in Kapitel 3.4.1 die Frage nach dem optimalen M stellt. Später, bei der Implementierung, ist eine Abbruchbedingung nötig, diese wird in 3.4.2 durch Abschätzung des erwarteten Fehlers aufgestellt. Anschließend wird gezeigt, wie diese Bedingung durch die Richardson Extrapolation Methode verbessert werden kann. In Kapitel 4 ist eine Möglichkeit angegeben, den Multilevel Monte-Carlo Algorithmus zu implementieren, sodass in Kapitel 5 unter Verwendung dieses Algorithmus verschiedene numerische Ergebnisse präsentiert werden. Die verwendeten Programme zur Auswertung sind in Matlab implementiert und befinden sich im Anhang. Zuerst wird das MLMC Verfahren mit dem MC Verfahren für europäische und asiatische Optionen anhand jeweils eines Beispiels verglichen. Dabei werden die numerischen Resultate anhand mehrerer Grafiken dargestellt. Danach werden auch Optionen untersucht für die keine geschlossenen Lösungsformeln existieren. Dazu ist es interessant verschiedene Multi-Asset Optionen zu betrachten. Als letztes Beispiel wird eine Option mit variabler Volatilität herangezogen. Der Aktienkurs folgt hierbei dem Heston Model. Auch hier ist an verschiedenen Grafiken ersichtlich, dass das MLMC Verfahren eine deutliche Verbesserung erbringt. Zum Schluss werden die Ergebnisse nochmals zusammengefasst und ein Ausblick auf weitere Möglichkeiten gegeben, die den Rechenaufwand reduzieren.

2 Grundlagen

Um den Kurs einer Aktie mit numerischen Verfahren zu simulieren, werden die für diese Arbeit wichtigsten Grundlagen eingeführt.

2.1 Stochastische Numerik

Die Kursentwicklung einer Aktie wird mittels einer Stochastischen Differentialgleichung (SDE) beschrieben. Numerische Verfahren werden verwendet, um diese zu lösen. Eines dieser Verfahren wird in diesem Kapitel vorgestellt.

Definition 2.1.1 Seien $a, b : \mathbb{R} \times [0, T] \rightarrow \mathbb{R}$ zwei Funktionen und $W : \mathbb{R}^+ \rightarrow \mathbb{R}$ ein Wiener-Prozess. Dann ist durch

$$dS(t) = a(S(t), t)dt + b(S(t), t)dW(t) \quad (2.1)$$

eine Stochastische Differentialgleichung gegeben. Diese wird als Integralgleichung

$$S(t) = S(t_0) + \int_{t_0}^t a(S(u), u)du + \int_{t_0}^t b(S(u), u)dW(u)$$

interpretiert, wobei das zweite Integral ein Ito-Integral ist.

Im Zuge der Finanzmathematik ist durch die Funktion a der Drift und durch b die Volatilität der Aktie gegeben. $S(t_0) \in \mathbb{R}$ bezeichnet den jetzigen Kurs der Aktie. Zur Diskretisierung der SDE (2.1) gibt es verschiedene numerische Verfahren. Das einfachste ist das Euler-Maruyama-Verfahren. Hierbei folgt der Aktienkurs S der Dynamik

$$\hat{S}(t_{n+1}) = \hat{S}(t_n) + a(\hat{S}(t_n), t_n)h + b(\hat{S}(t_n), t_n)\Delta W_n, \quad (2.2)$$

wobei $h = T/N$ die Schrittweite zu $N \in \mathbb{N}$ Diskretisierungsschritte auf dem Gitter $t_n = nh$, für $n = 0, 1, \dots, N$, ist und $\Delta W_n = W(t_n) - W(t_{n-1})$.

Die Genauigkeit der Näherung $\hat{S}(T)$ bezüglich des exakten Wertes $S(T)$ wird durch zwei Formen der Konvergenz erfasst.

Definition 2.1.2 (Starke Konvergenz) Ein numerisches Verfahren zur Lösung einer SDE konvergiert stark mit Ordnung $\alpha > 0$, falls für alle $p \in \mathbb{N}$

$$\max_n E \left(|\hat{S}(t_n) - S(t_n)|^p \right)^{\frac{1}{p}} \leq C_T h^\alpha$$

gilt, mit einer von h unabhängigen Konstanten C_T .

Definition 2.1.3 (Schwache Konvergenz) Ein numerisches Verfahren zur Lösung einer SDE konvergiert schwach mit Ordnung $\beta > 0$, falls für alle Polynome P

$$\max_n |E[P(\hat{S}(t_n))] - E[P(S(t_n))]| \leq C_{T,P} h^\beta$$

gilt, mit einer von h unabhängigen Konstanten $C_{T,P}$.

Bei beiden Definitionen bezeichnet $\hat{S}(t_n)$ die numerische Approximation zur Schrittweite h .

Satz 2.1.4 (Konvergenz des Euler-Maruyama Verfahrens) Gilt für die Funktionen $a(S, t)$ und $b(S, t)$ aus (2.1) mit Konstanten K_1, K_2, K_3, K_4

1. $|a(x, t) + a(y, t)| + |b(x, t) + b(y, t)| \leq K_1 |x - y|$,
2. $|a(x, t)| + |b(x, t)| \leq K_2 (1 + |x|)$,
3. $|a(x, s) - a(x, t)| + |b(x, s) - b(x, t)| \leq K_3 (1 + |x|) |s - t|^{1/2}$,
4. $a, b \in C^{(2+\epsilon)}$ für ein $\epsilon > 0$ (mehr als 2-mal differenzierbar),

dann existiert eine eindeutige Lösung der SDE (2.1) und das Euler-Maruyama Verfahren konvergiert stark mit Ordnung $1/2$ und schwach mit Ordnung 1 .

Beweis: [15], Theorem 10.2.2 und 14.1.5

2.2 Grundlagen der Optionsbewertung

Es folgen benötigte Grundlagen aus der Optionsbewertung.

Definition 2.2.1 (Äquivalentes Martingalmaß) Als äquivalentes Martingalmaß P^* zur Wahrscheinlichkeitsverteilung P von $S(t)$ wird das Wahrscheinlichkeitsmaß bezeichnet, für das

$$e^{-r\Delta t} E^*[S(t + \Delta t)] = S(t)$$

gilt, wobei r der risikolosen Zinsrate entspricht.

Satz 2.2.2 Für eine Option die nur zum Endzeitpunkt T ausgeübt werden kann, ergibt sich der faire Preis unter dem äquivalenten Martingalmaß durch

$$V(S, 0) = e^{-rT} E^*[V(S, T)], \tag{2.3}$$

wobei r die risikolose Zinsrate und $V(S, t)$ den Wert der Option zum Zeitpunkt $t \in [0, T]$ bezeichnet.

Beweis: [9], S.21

Bemerkung 2.2.3 (Black-Scholes Modell) *Im Black-Scholes Modell folgt der Aktienkurs der SDE*

$$dS(t) = \mu S(t)dt + \sigma S(t)dW(t), \quad (2.4)$$

mit Drift μ und Volatilität σ . Ersetzt man μ durch die risikolose Zinsrate r , so erhält man das äquivalente Martingalmaß P^* zur Wahrscheinlichkeitsverteilung P von $S(t)$ (vgl. [4]).

2.3 Multi-Asset Optionen

Multi-Asset Optionen, sind Optionen, deren Wert von der Kursentwicklung mehrerer Underlyings abhängt. Dieses Kapitel widmet sich den Besonderheiten die bei der Simulation der Aktienkurse beachtet werden müssen. Im Folgenden wird eine Option auf n Aktien betrachtet. Der Kursverlauf jeder einzelnen Aktie wird mit (2.1) beschrieben, d.h.

$$dS_i(t) = a_i(S_i(t), t)dt + b_i(S_i(t), t)dW_i(t) \quad (2.5)$$

für $i = 1, \dots, n$. $S_i(t)$ bezeichnet den Kurs der i -ten Aktie zum Zeitpunkt t , die Funktionen $a_i : \mathbb{R} \times [0, T] \rightarrow \mathbb{R}$ und $b_i : \mathbb{R} \times [0, T] \rightarrow \mathbb{R}$ geben den Drift bzw. die Volatilität der i -ten Aktie an und $W_i : \mathbb{R}^+ \rightarrow \mathbb{R}$ beschreibt eine Standard Brownsche Bewegung. Die Inkremente dW_i der Brownschen Bewegungen sind normalverteilt mit Erwartungswert 0 und Varianz θ_i . Zwei Inkremente dW_i und dW_j sind miteinander korreliert, $p_{i,j}$ ist der entsprechende Korrelationskoeffizient. Die Korrelationsmatrix ist gegeben durch

$$\begin{pmatrix} p_{1,1} & p_{1,2} & p_{1,3} & \cdots & p_{1,n} \\ p_{2,1} & p_{2,2} & p_{2,3} & \cdots & p_{2,n} \\ \vdots & \vdots & \vdots & & \vdots \\ p_{n,1} & p_{n,2} & p_{n,3} & \cdots & p_{n,n} \end{pmatrix},$$

wobei $p_{i,j} = p_{j,i}$ und $p_{i,i} = 1$ für $i, j = 1, \dots, n$ gilt. Für $dW = (dW_1, dW_2, \dots, dW_n)^T$ folgt

$$dW \sim N(0, \Sigma),$$

mit Kovarianzmatrix

$$\Sigma = \begin{pmatrix} \theta_1^2 & \theta_2\theta_1p_{2,1} & \cdots & \theta_n\theta_1p_{n,1} \\ \theta_1\theta_2p_{1,2} & \theta_2^2 & \cdots & \theta_n\theta_2p_{n,2} \\ \vdots & \vdots & \ddots & \vdots \\ \theta_1\theta_np_{1,n} & \theta_2\theta_np_{2,n} & \cdots & \theta_n^2 \end{pmatrix}.$$

N bezeichnet dabei die multivariate Normalverteilung. Diese besitzt unter anderem folgende Eigenschaft (vgl. [9]).

Satz 2.3.1 (Lineare Transformation Eigenschaft)

Sei X multivariat normalverteilt, dann ist auch jede lineare Transformation von X , multivariat normalverteilt, d.h.

$$X \sim N(\mu, \Sigma) \implies AX \sim N(A\mu, A\Sigma A^T), \quad (2.6)$$

für jeden Vektor $\mu \in \mathbb{R}^d$, $\Sigma \in \mathbb{R}^{d \times d}$ und jede Matrix $A \in \mathbb{R}^{k \times d}$ mit $d, k \in \mathbb{N}$.

Beweis: [9], S.65

Sei $Q \sim N(0, I)$, wobei I die Einheitsmatrix der Größe n bezeichnet, dann gilt mit $AA^T = \Sigma$ und (2.6)

$$AQ \sim dW \sim N(0, AA^T).$$

Da die Kovarianzmatrix Σ stets positiv definit ist, kann A mittels Cholesky-Zerlegung bestimmt werden. Dies hat insbesondere den Vorteil, dass dann A einer unteren Dreiecksmatrix entspricht und sich (2.5) somit umschreiben lässt zu

$$\begin{aligned} dS_1(t) &= a_1(S_1(t), t)dt + b_1(S_1(t), t)A_{1,1}Q_1, \\ dS_2(t) &= a_2(S_2(t), t)dt + b_2(S_2(t), t)(A_{2,1}Q_1 + A_{2,2}Q_2), \\ &\vdots \\ dS_n(t) &= a_n(S_n(t), t)dt + b_n(S_n(t), t)(A_{n,1}Q_1 + A_{n,2}Q_2 + \cdots + A_{n,n}Q_n). \end{aligned}$$

Da Q_i für $i = 1, \dots, n$ standardnormalverteilt ist und Q_i unabhängig von Q_j für $i \neq j$, lassen sich mit dem Euler-Maruyama Verfahren, die Aktienkurse einfach simulieren.

3 Multilevel Monte-Carlo Simulationsverfahren

Um die Multilevel Monte-Carlo (MLMC) Methode mit der Monte-Carlo Methode vergleichen zu können, werden zuerst die für diese Arbeit wesentliche Merkmale des Monte-Carlo Verfahrens erfasst. Anschließend wird die Multilevel Monte-Carlo Methode beschrieben und die Reduzierung der Rechenkosten gezeigt. Danach werden die Ergebnisse im Komplexitätstheorem verallgemeinert. Zum Schluss dieses Kapitels werden einige Erweiterungen vorgestellt, die für die spätere Implementierung des Algorithmus wichtig sind. Dieses Kapitel bezieht sich auf [5] und [11].

3.1 Monte-Carlo Methode

Monte Carlo Methoden beschreiben eine Gruppe von Algorithmen, die Zufallszahlen zur Simulation von stochastischen Prozessen verwenden. Im Allgemeinen wird die Monte-Carlo Methode zur Approximation von Erwartungswerten benutzt. In der Finanzmathematik werden damit häufig Optionspreise näherungsweise berechnet für die keine geschlossene Lösungsformel existiert oder für die es zu aufwändig wäre den exakten Preis zu berechnen. Bei der Monte-Carlo Methode entspricht der approximierte Optionswert dem arithmetischen Mittel aus N unabhängigen Simulationen des Optionswerts.

Sei $f : \mathbb{R} \rightarrow \mathbb{R}$ dasjenige Funktional, sodass $f(S(T))$ die diskontierte Auszahlung einer Option auf die Aktie S angibt, bei der die Auszahlung nur vom Kurs der Aktie zum Ausübungszeitpunkt $S(T)$ abhängt. Sei die Funktion f weiterhin lipschitzstetig, so folgt für die Approximation der diskontierte Auszahlung $f(\hat{S}(T))$, mit $|f(\hat{S}(T)) - f(S(T))| \leq c|\hat{S}(T) - S(T)|$ für eine Konstante c , die gleiche schwache, sowie starke Konvergenz wie für die entsprechende Approximation von $S(T)$. Durch Bildung des arithmetischen Mittels aus N unabhängigen Approximationen von $f(S(T))$ zur Schrittweite h ergibt sich der Schätzer $\hat{Y} = N^{-1} \sum_{i=1}^N f(\hat{S}_{T/h}^{(i)})$ von $Y = E[f(S(T))]$, dessen Genauigkeit mit dem mean square error

$$\begin{aligned} MSE &= E[|\hat{Y} - Y|^2] \\ &= (E[\hat{Y} - Y])^2 + E[|\hat{Y} - Y|^2] - (E[\hat{Y} - Y])^2 \\ &= (E[\hat{Y} - Y])^2 + Var[\hat{Y}], \end{aligned} \tag{3.1}$$

bestimmt wird. Der MSE wird nun durch einzelne Abschätzungen des quadrierten Bias

und der Varianz bestimmt. Für die Varianz von \hat{Y} gilt

$$\begin{aligned} \text{Var}[\hat{Y}] &= \text{Var}\left[N^{-1} \sum_{i=1}^N f(\hat{S}_{T/h}^{(i)})\right] \\ &= N^{-2} \sum_{i=1}^N \text{Var}[f(\hat{S}_{T/h}^{(i)})] \\ &\leq c_1 N^{-1}, \end{aligned}$$

mit $c_1 = \sup_{h \in (0,1]} \text{Var}[f(\hat{S}_{T/h}^{(i)})]$. Für den quadrierten Bias gilt mit der schwachen Konvergenz des Euler-Maruyama Verfahrens, welches in diesem Kapitel stets zur Approximation des Aktienkurses verwendet wird und der Voraussetzung einer Lipschitzstetigen Auszahlungsfunktion,

$$\begin{aligned} NE[Y - \hat{Y}] &= NE\left[f(S(T)) - N^{-1} \sum_{i=1}^N f(\hat{S}_{T/h}^{(i)})\right] \\ &= E\left[Nf(S(T)) - \sum_{i=1}^N f(\hat{S}_{T/h}^{(i)})\right] \\ &\leq N \sup_i \left(E[|f(S(T)) - f(\hat{S}_{T/h}^{(i)})|]\right) \\ &\leq Nc'_2 h, \end{aligned}$$

mit einer Konstanten $c'_2 \in \mathbb{R}$. Daraus ergibt sich $E[Y - \hat{Y}] \approx c'_2 h$ und

$$MSE \approx c_1 N^{-1} + c_2 h^2, \quad (3.2)$$

mit Konstanten c_1 und c_2 . Da ein RMSE von $O(\epsilon)$ erzielt werden soll, ist ein MSE von $O(\epsilon^2)$ notwendig und somit

$$N = O(\epsilon^{-2}) \quad \text{und} \quad h = O(\epsilon). \quad (3.3)$$

Mit dieser Wahl von h folgt ein erwarteter Fehler $E[Y - \hat{Y}]$ der Größenordnung $O(\epsilon)$. Weiterhin folgt für die Rechenkosten $C = O(\epsilon^{-3})$, da zur Schrittweite h für jede Pfadsimulation $1/h$ Iterationen notwendig sind und die Rechenkosten somit bei N Simulationen proportional zu N/h sind.

3.2 Multilevel Monte-Carlo Methode

Bei der Multilevel Monte-Carlo Methode werden nun die Aktienkurse zu unterschiedlichen Schrittweiten simuliert. Dazu wird eine geometrische Reihenfolge von verschiedenen

Schrittweiten $h_l = M^{-l}T, l = 0, 1, \dots, L$, verwendet, wobei $M \geq 2$ gewählt wird. Welche M empfehlenswert sind, wird später in Kapitel 3.4.1 gezeigt. Die Idee, die sich dahinter verbirgt, ist dass eine Simulation zu einer kleinen Schrittweite zwar einen geringen Diskretisierungsfehler ergibt, allerdings dafür sehr hohe Rechenkosten. Simulationen mit großen Schrittweiten ergeben eine geringere Genauigkeit, dafür aber auch niedrigere Rechenkosten. Die MLMC Methode erreicht nun die Genauigkeit der feinsten Schrittweite, benutzt allerdings größere Schrittweiten, um die Varianz so zu verkleinern, damit geringere Rechenkosten entstehen.

In diesem Kapitel wird gezeigt, dass die Rechenkosten durch das MLMC Verfahren von $O(\epsilon^3)$ auf $O(\epsilon^2(\log(\epsilon))^2)$ im Vergleich zur Monte-Carlo Methode bei gleicher Genauigkeit sinken. Dabei wird stets eine lipschitzstetige Auszahlungsfunktion und das Euler-Maruyama Verfahren zur Simulation des Aktienkurses verwendet.

Um die Rechenkosten zu verringern, wird die Auszahlung der Option auf dem feinsten Level L umgeschrieben

$$E[\hat{P}_L] = E[\hat{P}_0] + \sum_{l=1}^L E[\hat{P}_l - \hat{P}_{l-1}]. \quad (3.4)$$

Dabei sei $P = f(S(T))$ der Wert der Option und \hat{P}_l die entsprechende Approximation zur Schrittweite $h_l = M^{-l}T$. Die Erwartungswerte der Samples $\hat{P}_l - \hat{P}_{l-1}$ werden für $l = 1, \dots, L$ unabhängig simuliert. Indem \hat{P}_l und \hat{P}_{l-1} jedes Samples aus einem Brownschen Pfad errechnet werden, können die Rechenkosten niedrig gehalten werden. Dazu wird ein Pfad zur Schrittweite h_l erzeugt und anschließend werden M Inkremente des Brownschen Pfads summiert, um ein Brownsches Inkrement zum Level $l-1$ zu erhalten. Die richtige Verteilung der Inkremente ist noch nachzuprüfen.

Für die Zuwächse der Brownschen Bewegung zum Level l gilt

$$\Delta W_{n,l} \sim N(0, h_l) \sim \sqrt{h_l} Z_n \quad \text{für } n = 1, \dots, T/h_l,$$

wobei $Z_n \sim N(0, 1)$ verteilt ist und N die Normalverteilung bezeichnet. Somit folgt für das Level $l-1$

$$\Delta W_{m,l-1} = \sum_{j=1}^M \sqrt{h_l} Z_{mj} \quad \text{für } m = 1, \dots, T/(h_l M)$$

und damit

$$E[\Delta W_{m,l-1}] = E\left[\sum_{j=1}^M \sqrt{h_l} Z_{mj}\right] = 0,$$

$$Var\left[\sum_{j=1}^M \sqrt{h_l} Z_{mj}\right] = h_l \sum_{j=1}^M Var[Z_{mj}] = h_l M = h_{l-1}.$$

Die Inkremente besitzen hiermit die geforderte Verteilung. Aus dem Schätzer

$$\hat{Y}_l = N_l^{-1} \sum_{i=1}^{N_l} (\hat{P}_l^{(i)} - \hat{P}_{l-1}^{(i)}), \quad \text{für } l = 1, \dots, L, \quad (3.5)$$

vom Erwartungswert der Samples $E[\hat{P}_l - \hat{P}_{l-1}]$ und dem Schätzer

$$\hat{Y}_0 = N_0^{-1} \sum_{i=1}^{N_0} \hat{P}_0^{(i)} \quad (3.6)$$

von $E[\hat{P}_0]$, ergibt sich der Multilevel-Schätzer

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

der $E[\hat{P}_L]$ approximiert. N_l bezeichnet dabei die Anzahl der Pfadsimulationen des jeweiligen Samples. Für die Varianz der Schätzer \hat{Y}_l gilt

$$\begin{aligned} \text{Var}[\hat{Y}_l] &= \text{Var} \left[N_l^{-1} \sum_{i=1}^{N_l} (\hat{P}_l^{(i)} - \hat{P}_{l-1}^{(i)}) \right] \\ &= N_l^{-2} \text{Var} \left[\sum_{i=1}^{N_l} (\hat{P}_l^{(i)} - \hat{P}_{l-1}^{(i)}) \right] \\ &= N_l^{-1} V_l, \end{aligned}$$

mit $V_l = \text{Var}[(\hat{P}_l - \hat{P}_{l-1})]$ für $l = 1, \dots, L$ und $V_0 = \text{Var}[\hat{P}_0]$. Daraus ergibt sich die Varianz des Multilevel-Schätzers

$$\text{Var}[\hat{Y}] = \text{Var} \left[\sum_{l=0}^L \hat{Y}_l \right] = \sum_{l=0}^L \text{Var}[\hat{Y}_l] = \sum_{l=0}^L N_l^{-1} V_l. \quad (3.7)$$

Als nächstes soll untersucht werden für welche N_l die Varianz minimiert wird, ohne die Rechenkosten zu verändern. Dies führt zu einem Extremwertproblem mit Nebenbedingung. Es wird $f(N) = \sum_{l=0}^L N_l^{-1} V_l$ und $g(N) = \sum_{l=0}^L N_l h_l^{-1} - K$ mit $N = (N_0, \dots, N_L)$ gesetzt. Wobei $K \in \mathbb{R}$ so gewählt wird, dass $g(N) = 0$ gilt. Dann ist N ein Extrema (vgl. [17], S.225), falls Lagrange-Multiplikatoren $\lambda_0, \dots, \lambda_L \in \mathbb{R}$ existieren, mit

$$\nabla f(N) = \lambda_0 \nabla g_0(N) + \dots + \lambda_L \nabla g_L(N).$$

N ist also genau dann ein Minimum, wenn

$$-V_l N_l^{-2} = \lambda_l h_l^{-1}, \quad \text{für } l = 0, \dots, L,$$

gilt. Daraus folgt $N_l = \sqrt{\lambda'_l V_l h_l}$ mit $\lambda'_l = -\lambda_l^{-1}$ und somit

$$N_l \sim \sqrt{V_l h_l}. \quad (3.8)$$

Zur Betrachtung des Grenzfalls von V_l für $l \rightarrow \infty$ wird $L \gg 1$ gewählt. Weiterhin wird der Aktienkurs S mit dem Euler-Maruyama Verfahren approximiert, dessen schwache und starke Konvergenz in den folgenden Abschätzungen verwendet wird und Lipschitzstetigkeit der Auszahlungsfunktion angenommen. \hat{S}_{l,M^l} bezeichnet die Approximation von $S(T)$ zur Schrittweite h_l . Nach Satz 2.1.4 gilt

$$E[\hat{P}_l - P] = O(h_l) \quad \text{und} \quad E[|\hat{S}_{l,M^l} - S(T)|^2] = O(h_l). \quad (3.9)$$

Mit der Lipschitzstetigkeit folgt

$$\begin{aligned} \text{Var}[\hat{P}_l - P] &= E[(\hat{P}_l - P)^2] - (E[\hat{P}_l - P])^2 \\ &\leq E[(\hat{P}_l - P)^2] \\ &\leq c^2 E[(\hat{S}_{l,M^l} - S(T))^2] \end{aligned}$$

und somit

$$\text{Var}[\hat{P}_l - P] = O(h_l). \quad (3.10)$$

Für die Varianz von $\hat{P}_l - \hat{P}_{l-1}$ gilt

$$\begin{aligned} \text{Var}[\hat{P}_l - \hat{P}_{l-1}] &= \text{Var}[(\hat{P}_l - P) - (\hat{P}_{l-1} - P)] \\ &= \text{Var}[\hat{P}_l - P] - 2\text{Cov}[\hat{P}_l - P, \hat{P}_{l-1} - P] + \text{Var}[\hat{P}_{l-1} - P] \\ &\leq \text{Var}[\hat{P}_l - P] + 2\sqrt{\text{Var}[\hat{P}_l - P]\text{Var}[\hat{P}_{l-1} - P]} + \text{Var}[\hat{P}_{l-1} - P] \\ &= ((\text{Var}[\hat{P}_l - P])^{1/2} + (\text{Var}[\hat{P}_{l-1} - P])^{1/2})^2. \end{aligned} \quad (3.11)$$

Zusammen mit (3.10) ergibt dies $V_l = O(h_l)$. Nach (3.8) muss das optimale N nun proportional zu h_l sein. Mit der Wahl von $N_l = O(\epsilon^{-2}(L+1)h_l)$ folgt zusammen mit (3.7) und einer Konstanten K

$$\text{Var}[\hat{Y}] = \sum_{l=0}^L N_l^{-1} V_l = \sum_{l=0}^L (K\epsilon^{-2}(L+1)h_l)^{-1} h_l = O(\epsilon^2). \quad (3.12)$$

Mit der Wahl von $L = \frac{\log \epsilon^{-1}}{\log M} + O(1)$ und $\epsilon \rightarrow 0$ folgt für die Schrittweite h_L auf dem feinsten Level

$$\begin{aligned}
h_L &= TM^{-L} \\
&= TM^{-\left(\frac{\log \epsilon^{-1}}{\log M} + K\right)} \\
&= Te^{-\log M \left(\frac{\log \epsilon^{-1}}{\log M} + K\right)} \\
&= T\epsilon e^{-(\log M)K} \\
&= O(\epsilon),
\end{aligned}$$

mit einer Konstanten K . Mit (3.9) ist $E[\hat{P}_L - P] = O(\epsilon)$. Dies ergibt zusammen mit (3.12) und (3.1) ein MSE von $O(\epsilon^2)$. Und für die Rechenkosten C gilt mit der Wahl von N_l und L , sowie Konstanten c_1, c_2 und c_3

$$\begin{aligned}
C &= N_0 + c_1 \sum_{l=1}^L N_l (h_l^{-1} + h_{l-1}^{-1}) \\
&\approx c_1 \sum_{l=0}^L N_l h_l^{-1} \\
&= c_1 \sum_{l=0}^L (c_2 \epsilon^{-2} (L+1) h_l) h_l^{-1} \\
&= c_3 \sum_{l=0}^L \epsilon^{-2} (L+1) \\
&= c_3 \epsilon^{-2} (L+1)^2 \\
&= O(\epsilon^{-2} (\log \epsilon)^2).
\end{aligned}$$

3.3 Komplexitätstheorem

Im Komplexitätstheorem werden die bisherigen Ergebnisse verallgemeinert. Es kann auf Auszahlungsfunktionen, die nicht lipschitzstetig sind, angewandt werden und das Euler-Maruyama Verfahren zur Simulation der Aktienkurse kann durch andere numerische Verfahren ersetzt werden.

Satz 3.3.1 Sei $S(T)$ die Lösung der SDE (2.1) für einen gegebenen Brownschen Pfad $W(t)$. Für eine Funktion f sei $P = f(S(T))$ und \hat{P}_l die entsprechende Approximation, die aus einem numerischen Diskretisierungsverfahren zur Schrittweite $h_l = M^{-l}T$ gewonnen wird.

Wenn unabhängige, aus N_l Monte-Carlo Simulationen entstandene Schätzer \hat{Y}_l und positive Konstanten $\alpha \geq 1/2, \beta, c_1, c_2, c_3$ existieren, sodass gilt

$$i) E[\hat{P}_l - P] \leq c_1 h_l^\alpha$$

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

$$iii) Var[\hat{Y}_l] \leq c_2 N_l^{-1} h_l^\beta$$

$$iv) C_l, \text{ die Rechenkosten von } \hat{Y}_l, \text{ sind beschränkt durch } C_l \leq c_3 N_l h_l^{-1},$$

dann gibt es eine Konstante $c_4 > 0$, sodass für jedes $\epsilon < e^{-1}$ Werte L und N_l existieren, für die der Multilevel-Schätzer

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

einen

$$MSE = E[(\hat{Y} - E[P])^2] < \epsilon^2$$

besitzt und Rechenkosten

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

Beweis: $\lceil x \rceil$ bezeichnet stets die ganze Zahl n für die $x \leq n < x + 1$ gilt. Zuerst wird ein L gesucht mit

$$\frac{1}{\sqrt{2}}M^{-\alpha}\epsilon < c_1 h_L^\alpha \leq \frac{1}{\sqrt{2}}\epsilon. \quad (3.13)$$

Wird

$$L = \left\lceil \frac{\log(\sqrt{2}c_1 T^\alpha \epsilon^{-1})}{\alpha \log M} \right\rceil \quad (3.14)$$

gewählt, so folgt (3.13) mit folgender Umformung

$$\begin{aligned} & \frac{\log(\sqrt{2}c_1 T^\alpha \epsilon^{-1})}{\alpha \log M} \leq L &< \frac{\log(\sqrt{2}c_1 T^\alpha \epsilon^{-1})}{\alpha \log M} + 1 \\ \implies & M^{-\left(\frac{\log(\sqrt{2}c_1 T^\alpha \epsilon^{-1})}{\alpha \log M}\right)} \geq M^{-L} &> M^{-\left(\frac{\log(\sqrt{2}c_1 T^\alpha \epsilon^{-1})}{\alpha \log M} + 1\right)} \\ \implies & e^{-(\alpha \log M)\left(\frac{\log(\sqrt{2}c_1 T^\alpha \epsilon^{-1})}{\alpha \log M}\right)} \geq M^{-L\alpha} &> e^{-(\alpha \log M)\left(\frac{\log(\sqrt{2}c_1 T^\alpha \epsilon^{-1})}{\alpha \log M} + 1\right)} \\ \implies & e^{-\log(\sqrt{2}c_1 T^\alpha \epsilon^{-1})} \geq M^{-L\alpha} &> e^{-\left(\log(\sqrt{2}c_1 T^\alpha \epsilon^{-1}) + \log M^\alpha\right)} \\ \implies & (\sqrt{2}c_1 T^\alpha \epsilon^{-1})^{-1} \geq M^{-L\alpha} &> (\sqrt{2}c_1 T^\alpha \epsilon^{-1} M^\alpha)^{-1} \\ \implies & \frac{1}{\sqrt{2}}\epsilon \geq c_1 T^\alpha M^{-L\alpha} &> \frac{1}{\sqrt{2}}\epsilon M^{-\alpha} \\ \xrightarrow{h_L = M^{-L}T} & \frac{1}{\sqrt{2}}\epsilon M^{-\alpha} < c_1 h_L^\alpha \leq \frac{1}{\sqrt{2}}\epsilon. \end{aligned}$$

Mit diesem Ergebnis lässt sich zusammen mit den Eigenschaften i) und ii) der quadrierte Bias abschätzen. Dazu ist eine Fallunterscheidung für L nötig:

1.Fall: $L = 0$

$$\begin{aligned} (E[\hat{Y}] - E[P])^2 &= (E[\hat{Y}_0] - E[P])^2 \\ &\stackrel{ii)}{=} (E[\hat{P}_0] - E[P])^2 \\ &\stackrel{i)}{\leq} (c_1 h_L^\alpha)^2 \\ &\stackrel{(3.13)}{\leq} \frac{1}{2}\epsilon^2 \end{aligned} \quad (3.15)$$

2.Fall: $L > 0$

$$\begin{aligned}
(E[\hat{Y}] - E[P])^2 &= \left(E\left[\sum_{l=0}^L \hat{Y}_l \right] - E[P] \right)^2 \\
&\stackrel{ii)}{=} \left(\sum_{l=1}^L E[\hat{P}_l - \hat{P}_{l-1}] + E[\hat{P}_0] - E[P] \right)^2 \\
&= (E[\hat{P}_L] - E[P])^2 \\
&\stackrel{i)}{\leq} (c_1 h_L^\alpha)^2 \\
&\stackrel{(3.13)}{\leq} \frac{1}{2} \epsilon^2
\end{aligned} \tag{3.16}$$

Dieses Ergebnis wird später benötigt, um zusammen mit der Varianz des Multilevel-Schätzers einen MSE kleiner ϵ^2 zu erhalten. Aus

$$\begin{aligned}
\sum_{l=0}^L h_l^{-1} &= \frac{1}{T} \sum_{l=0}^L M^l = \frac{1}{T} \sum_{l=0}^L M^{L-l} \\
&\leq \frac{1}{M^{-L} T} \sum_{l=0}^L M^{-l} \leq h_L^{-1} \lim_{L \rightarrow \infty} \sum_{l=0}^L M^{-l} \\
&\leq h_L^{-1} \frac{1}{1 - M^{-1}} = h_L^{-1} \frac{M}{M - 1}
\end{aligned} \tag{3.17}$$

und

$$\begin{aligned}
h_L &\stackrel{(3.13)}{>} M^{-1} \left(\frac{\epsilon}{\sqrt{2}c_1} \right)^{\frac{1}{\alpha}} \\
\implies h_L^{-1} &< M \left(\frac{\epsilon}{\sqrt{2}c_1} \right)^{-\frac{1}{\alpha}}
\end{aligned}$$

folgt zusammen mit $\epsilon^{-\frac{1}{\alpha}} \leq \epsilon^{-2}$ für $\alpha \geq \frac{1}{2}, \epsilon < e^{-1}$

$$\sum_{l=0}^L h_l^{-1} < \frac{M^2}{M-1} \left(\frac{\epsilon}{\sqrt{2}c_1} \right)^{-\frac{1}{\alpha}} < \frac{M^2}{M-1} (\sqrt{2}c_1)^{\frac{1}{\alpha}} \epsilon^{-2}. \tag{3.18}$$

Mit diesem Ergebnis werden die Rechenkosten abgeschätzt. Da die Rechenkosten C je nach β unterschiedlich sind, ist nun eine Fallunterscheidung für die Werte von β notwendig.

1.Fall: $\beta = 1$

Um einen $MSE < \epsilon^2$ zu erhalten wird

$$N_l = \lceil 2\epsilon^{-2}(L+1)c_2h_l \rceil \quad (3.19)$$

gesetzt. Damit folgt für die Varianz des Multilevel-Schätzers

$$\begin{aligned} \text{Var}[\hat{Y}] &= \text{Var}\left[\sum_{l=0}^L \hat{Y}_l\right] \\ &= \sum_{l=0}^L \text{Var}[\hat{Y}_l] \\ &\stackrel{iii)}{\leq} \sum_{l=0}^L c_2 N_l^{-1} h_l \\ &\leq \sum_{l=0}^L c_2 h_l (2\epsilon^{-2}(L+1)c_2 h_l)^{-1} \\ &\leq \frac{\epsilon^2}{2}. \end{aligned}$$

Mit (3.16) und (3.1) folgt nun ein $MSE < \epsilon^2$. Um eine obere Grenze für die Rechenkosten C zu finden, wird zuerst L abgeschätzt, dabei wird $1 < \log \epsilon^{-1}$ für $\epsilon < e^{-1}$ verwendet,

$$\begin{aligned} L+1 &\stackrel{(3.14)}{\leq} \frac{\log(\sqrt{2}c_1 T^\alpha \epsilon^{-1})}{\alpha \log M} + 2 \\ &\leq \frac{\log \epsilon^{-1}}{\alpha \log M} + \frac{\log(\sqrt{2}c_1 T^\alpha)}{\alpha \log M} + 2 \\ &\leq \log \epsilon^{-1} \left(\frac{1}{\alpha \log M} + \frac{\log(\sqrt{2}c_1 T^\alpha)}{\alpha \log M \log \epsilon^{-1}} + \frac{2}{\log \epsilon^{-1}} \right) \\ &\stackrel{1 < \log \epsilon^{-1}}{\leq} \underbrace{\log \epsilon^{-1} \left(\frac{1}{\alpha \log M} + \max\left(0, \frac{\log(\sqrt{2}c_1 T^\alpha)}{\alpha \log M}\right) \right)}_{=: c_5} + 2 \\ &= c_5 \log \epsilon^{-1}. \end{aligned} \quad (3.20)$$

Dies führt zu den geforderten Rechenkosten von \hat{Y}

$$\begin{aligned}
C &\leq \sum_{l=0}^L C_l \\
&\stackrel{(iv)}{\leq} \sum_{l=0}^L c_3 N_l h_l^{-1} \\
&\stackrel{(3.19)}{\leq} \sum_{l=0}^L c_3 (2\epsilon^{-2}(L+1)c_2 h_l + 1) h_l^{-1} \\
&\leq c_3 \left(2\epsilon^{-2}(L+1)^2 c_2 + \sum_{l=0}^L h_l^{-1} \right) \\
&\stackrel{(3.20)}{\leq} c_3 \left(2\epsilon^{-2}(c_5 \log \epsilon^{-1})^2 c_2 + \sum_{l=0}^L h_l^{-1} \right) \\
&\stackrel{(3.18)}{\leq} c_3 \left(2\epsilon^{-2}(c_5 \log \epsilon^{-1})^2 c_2 + \frac{M^2}{M-1} (\sqrt{2}c_1)^{\frac{1}{\alpha}} \epsilon^{-2} \right) \\
&\leq \epsilon^{-2} (\log \epsilon)^2 \underbrace{\left(2c_2 c_3 c_5^2 + c_3 \frac{M^2}{M-1} (\sqrt{2}c_1)^{\frac{1}{\alpha}} (\log \epsilon)^{-2} \right)}_{=: c_4} \\
&= c_4 \epsilon^{-2} (\log \epsilon)^2.
\end{aligned}$$

2.Fall: $\beta > 1$

Nun wird

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

verwendet. Folgende Abschätzung wird benötigt

$$\begin{aligned}
\sum_{l=0}^L h_l^{\frac{\beta-1}{2}} &= \sum_{l=0}^L (TM^{-l})^{\frac{\beta-1}{2}} \\
&= T^{\frac{\beta-1}{2}} \sum_{l=0}^L (M^{-\frac{\beta-1}{2}})^l \\
&< T^{\frac{\beta-1}{2}} \sum_{l=0}^{\infty} (M^{-\frac{\beta-1}{2}})^l \\
&\stackrel{M \geq 2, \beta > 1}{\leq} T^{\frac{\beta-1}{2}} (1 - M^{-\frac{\beta-1}{2}})^{-1}.
\end{aligned} \tag{3.21}$$

Damit folgt für die Varianz des Multilevel-Schätzers

$$\begin{aligned}
\text{Var}[\hat{Y}] &= \text{Var}\left[\sum_{l=0}^L \hat{Y}_l\right] \\
&= \sum_{l=0}^L \text{Var}[\hat{Y}_l] \\
&\stackrel{\text{iii)}}{\leq} \sum_{l=0}^L c_2 N_l^{-1} h_l^\beta \\
&\leq \sum_{l=0}^L c_2 h_l^\beta \left(2\epsilon^{-2} c_2 T^{\frac{\beta-1}{2}} (1 - M^{-\frac{\beta-1}{2}})^{-1} h_l^{\frac{\beta+1}{2}}\right)^{-1} \\
&\leq \frac{1}{2} \epsilon^2 T^{-\frac{\beta-1}{2}} (1 - M^{-\frac{\beta-1}{2}}) \sum_{l=0}^L h_l^{\frac{\beta-1}{2}} \\
&\stackrel{(3.21)}{\leq} \frac{1}{2} \epsilon^2. \tag{3.22}
\end{aligned}$$

Nach (3.1) und (3.16) ergibt sich ein $MSE < \epsilon^2$. Für die Rechenkosten gilt

$$\begin{aligned}
C &\leq \sum_{l=0}^L C_l \\
&\stackrel{\text{iv)}}{\leq} \sum_{l=0}^L c_3 N_l h_l^{-1} \\
&\leq \sum_{l=0}^L c_3 h_l^{-1} (2\epsilon^{-2} c_2 T^{\frac{\beta-1}{2}} (1 - M^{-\frac{\beta-1}{2}})^{-1} h_l^{\frac{\beta+1}{2}} + 1) \\
&\leq c_3 (2\epsilon^{-2} c_2 T^{\frac{\beta-1}{2}} (1 - M^{-\frac{\beta-1}{2}})^{-1}) \sum_{l=0}^L h_l^{\frac{\beta-1}{2}} + \sum_{l=0}^L h_l^{-1} \\
&\stackrel{(3.21)}{\leq} c_3 (2\epsilon^{-2} c_2 (T^{\frac{\beta-1}{2}} (1 - M^{-\frac{\beta-1}{2}})^{-1})^2 + \sum_{l=0}^L h_l^{-1}) \\
&\stackrel{(3.18)}{\leq} c_3 \left(2\epsilon^{-2} c_2 (T^{\frac{\beta-1}{2}} (1 - M^{-\frac{\beta-1}{2}})^{-1})^2 + \frac{M^2}{M-1} (\sqrt{2} c_1)^{\frac{1}{\alpha}} \epsilon^{-2}\right) \\
&\leq \underbrace{\epsilon^{-2} c_3 \left(2c_2 T^{\beta-1} (1 - M^{-\frac{\beta-1}{2}})^{-2} + \frac{M^2}{M-1} (\sqrt{2} c_1)^{\frac{1}{\alpha}}\right)}_{=: c_4} \\
&\leq c_4 \epsilon^{-2}.
\end{aligned}$$

3.Fall: $0 < \beta < 1$

Nun wird

$$N_l = \left[2\epsilon^{-2} c_2 h_L^{-\frac{1-\beta}{2}} (1 - M^{-\frac{1-\beta}{2}})^{-1} h_l^{\frac{1+\beta}{2}} \right] \quad (3.23)$$

gewählt, die Vorgehensweise ist analog zu den vorigen Fällen. Zuerst ist

$$\begin{aligned} \sum_{l=0}^L h_l^{-\frac{1-\beta}{2}} &= h_L^{-\frac{1-\beta}{2}} \sum_{l=0}^L (h_l^{-\frac{1-\beta}{2}} h_L^{\frac{1-\beta}{2}}) \\ &\stackrel{h_l = M^{-lT}}{=} h_L^{-\frac{1-\beta}{2}} \sum_{l=0}^L (M^{-\frac{1-\beta}{2}})^l \\ &\leq h_L^{-\frac{1-\beta}{2}} \sum_{l=0}^{\infty} (M^{-\frac{1-\beta}{2}})^l \\ &\stackrel{M > 2, \beta < 1}{=} h_L^{-\frac{1-\beta}{2}} (1 - M^{-\frac{1-\beta}{2}})^{-1} \end{aligned} \quad (3.24)$$

nötig um die Varianz abzuschätzen

$$\begin{aligned} \text{Var}[\hat{Y}] &= \text{Var} \left[\sum_{l=0}^L \hat{Y}_l \right] \\ &= \sum_{l=0}^L \text{Var}[\hat{Y}_l] \\ &\stackrel{iii)}{\leq} \sum_{l=0}^L c_2 N_l^{-1} h_l^\beta \\ &\leq \sum_{l=0}^L c_2 h_l^\beta \left(2\epsilon^{-2} c_2 h_L^{-\frac{1-\beta}{2}} (1 - M^{-\frac{1-\beta}{2}})^{-1} h_l^{\frac{1+\beta}{2}} \right)^{-1} \\ &\leq \frac{1}{2} \epsilon^2 h_L^{\frac{1-\beta}{2}} (1 - M^{-\frac{1-\beta}{2}}) \sum_{l=0}^L h_l^{-\frac{1-\beta}{2}} \\ &\stackrel{(3.24)}{\leq} \frac{1}{2} \epsilon^2. \end{aligned}$$

Mit (3.1) und (3.16) folgt ein MSE von $O(\epsilon^2)$.

Für die Rechenkosten C gilt

$$\begin{aligned}
C &= \sum_{l=0}^L C_l \\
&\stackrel{iv)}{\leq} \sum_{l=0}^L c_3 N_l h_l^{-1} \\
&\stackrel{(3.23)}{\leq} \sum_{l=0}^L c_3 h_l^{-1} (2\epsilon^{-2} c_2 h_L^{-\frac{1-\beta}{2}} (1 - M^{-\frac{1-\beta}{2}})^{-1} h_l^{\frac{\beta+1}{2}} + 1) \\
&\leq c_3 \left(2\epsilon^{-2} c_2 h_L^{\frac{\beta-1}{2}} (1 - M^{\frac{\beta-1}{2}})^{-1} \sum_{l=0}^L h_l^{\frac{\beta-1}{2}} + \sum_{l=0}^L h_l^{-1} \right) \\
&\stackrel{(3.24)}{\leq} c_3 \left(2\epsilon^{-2} c_2 h_L^{\beta-1} (1 - M^{\frac{\beta-1}{2}})^{-2} + \sum_{l=0}^L h_l^{-1} \right).
\end{aligned}$$

Aus (3.13) folgt

$$\begin{aligned}
h_L^\alpha &> (\sqrt{2}c_1)^{-1} \epsilon M^{-\alpha} \\
\stackrel{\frac{\beta+1}{\alpha} < 0}{\implies} & (h_L^\alpha)^{\frac{\beta-1}{\alpha}} < ((\sqrt{2}c_1)^{-1} \epsilon M^{-\alpha})^{\frac{\beta-1}{\alpha}} \\
\implies & h_L^{\beta-1} < (\sqrt{2}c_1)^{\frac{1-\beta}{\alpha}} M^{1-\beta} \epsilon^{\frac{\beta-1}{\alpha}},
\end{aligned}$$

sodass für C, mit $\epsilon^{-2} < \epsilon^{-2 - \frac{1-\beta}{\alpha}}$ für $\epsilon < e^{-1}$, gilt

$$\begin{aligned}
C &\leq c_3 (2\epsilon^{-2} c_2 ((\sqrt{2}c_1)^{\frac{1-\beta}{\alpha}} M^{1-\beta} \epsilon^{\frac{\beta-1}{\alpha}}) (1 - M^{\frac{\beta-1}{2}})^{-2} + \sum_{l=0}^L h_l^{-1}) \\
&\stackrel{(3.18)}{\leq} c_3 (2\epsilon^{-2 - \frac{1-\beta}{\alpha}} c_2 (\sqrt{2}c_1)^{\frac{1-\beta}{\alpha}} M^{1-\beta} (1 - M^{\frac{\beta-1}{2}})^{-2} + \frac{M^2}{M-1} (\sqrt{2}c_1)^{\frac{1}{\alpha}} \epsilon^{-2}) \\
&\leq \underbrace{\epsilon^{-2 - \frac{1-\beta}{\alpha}} (c_3 2c_2 (\sqrt{2}c_1)^{\frac{1-\beta}{\alpha}} M^{1-\beta} (1 - M^{\frac{\beta-1}{2}})^{-2} + c_3 \frac{M^2}{M-1} (\sqrt{2}c_1)^{\frac{1}{\alpha}})}_{=: c_4} \\
&\leq c_4 \epsilon^{-2 - \frac{1-\beta}{\alpha}}.
\end{aligned}$$

□

Bemerkung 3.3.2 Bei Verwendung des Euler-Maruyama Verfahrens mit lipschitzstetiger Auszahlungsfunktion gilt nach (3.9) $E[\hat{P}_l - P] = O(h_l)$ und somit $\alpha = 1$ im Komplexitätstheorem. Weiterhin gilt $\text{Var}[\hat{Y}_l] = N_l^{-1} V_l$ und aus der starken Konvergenz wurde $V_l = O(h_l)$ ermittelt, sodass dies $\beta = 1$ im Komplexitätstheorem ergibt. Damit

folgen aus dem Komplexitätstheorem die bereits aus Kapitel 3.2 bekannten Rechenkosten.

3.4 Erweiterungen

3.4.1 Optimales M

Dieser Abschnitt widmet sich der Wahl des Parameters M , der die Schrittweiten $h_l = M^{-l}T$ der jeweiligen Level festlegt. Die folgenden Analysen zielen auf eine Minimierung der Rechenkosten. Weiterhin soll wieder das Euler-Maruyama Verfahren zur Diskretisierung der SDE (2.1) verwendet werden und die Auszahlungsfunktion lipschitzstetig sein, sodass die aus Kapitel 3.2 bereits bekannten Ergebnisse verwendet werden können.

Als nächstes werden Schranken für die Varianz des Schätzers \hat{Y}_l gesucht. Nach (3.7) bzw. (3.10) ist $Var[\hat{Y}_l] = N_l^{-1}V_l$ und $Var[P_l - P] \approx c_0 h_l$ bekannt. Zuerst wird $V_l = Var[\hat{P}_l - \hat{P}_{l-1}]$ abgeschätzt

$$\begin{aligned}
V_l &= Var[(\hat{P}_l - P) - (\hat{P}_{l-1} - P)] \\
&= Var[\hat{P}_l - P] + Var[\hat{P}_{l-1} - P] - 2Cov[(\hat{P}_l - P), (\hat{P}_{l-1} - P)] \\
&= Var[\hat{P}_l - P] + Var[\hat{P}_{l-1} - P] - 2p\sqrt{Var[\hat{P}_l - P]Var[\hat{P}_{l-1} - P]} \\
&\approx c_0 h_l + c_0 h_{l-1} - 2p\sqrt{c_0 h_l c_0 h_{l-1}} \\
&\stackrel{h_l = h_{l-1} M^{-1}}{=} c_0 h_l + c_0 h_l M - 2p\sqrt{c_0 h_l} \sqrt{c_0 h_l M} \\
&= c_0 h_l (1 - 2p\sqrt{M} + M),
\end{aligned}$$

wobei $p \in [-1, 1]$ die Korrelation bezeichnet. Mit dem Übergang zum Minimum für $p = 1$ und dem Maximum für $p = -1$ folgt

$$\begin{aligned}
(1 - 2\sqrt{M} + M)c_0 h_l &\leq V_l \leq (1 + 2\sqrt{M} + M)c_0 h_l \\
\stackrel{\text{bin. Formel}}{\implies} (\sqrt{M} - 1)^2 c_0 h_l &\leq V_l \leq (\sqrt{M} + 1)^2 c_0 h_l.
\end{aligned}$$

Zum weiteren Abschätzen wird das geometrische Mittel aus oberer und unterer Schranke gebildet

$$V_l \approx \sqrt{\left((\sqrt{M} - 1)^2 c_0 h_l\right) \left((\sqrt{M} + 1)^2 c_0 h_l\right)} = (M - 1)c_0 h_l. \quad (3.25)$$

Daraus folgt $\text{Var}[\hat{Y}_l] \approx N_l^{-1}(M-1)c_0h_l$. Weiterhin ist aus Satz 3.3.1 $\text{Var}[\hat{Y}_l] \leq c_2N_l^{-1}h_l$ bekannt. Dies führt zu der Schätzung

$$c_2 \approx (M-1)c_0$$

und somit nach (3.19) zu

$$N_l \approx 2\epsilon^2(L+1)(M-1)c_0h_l.$$

Daraus lassen sich nun die Rechenkosten für jedes Level ableiten

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

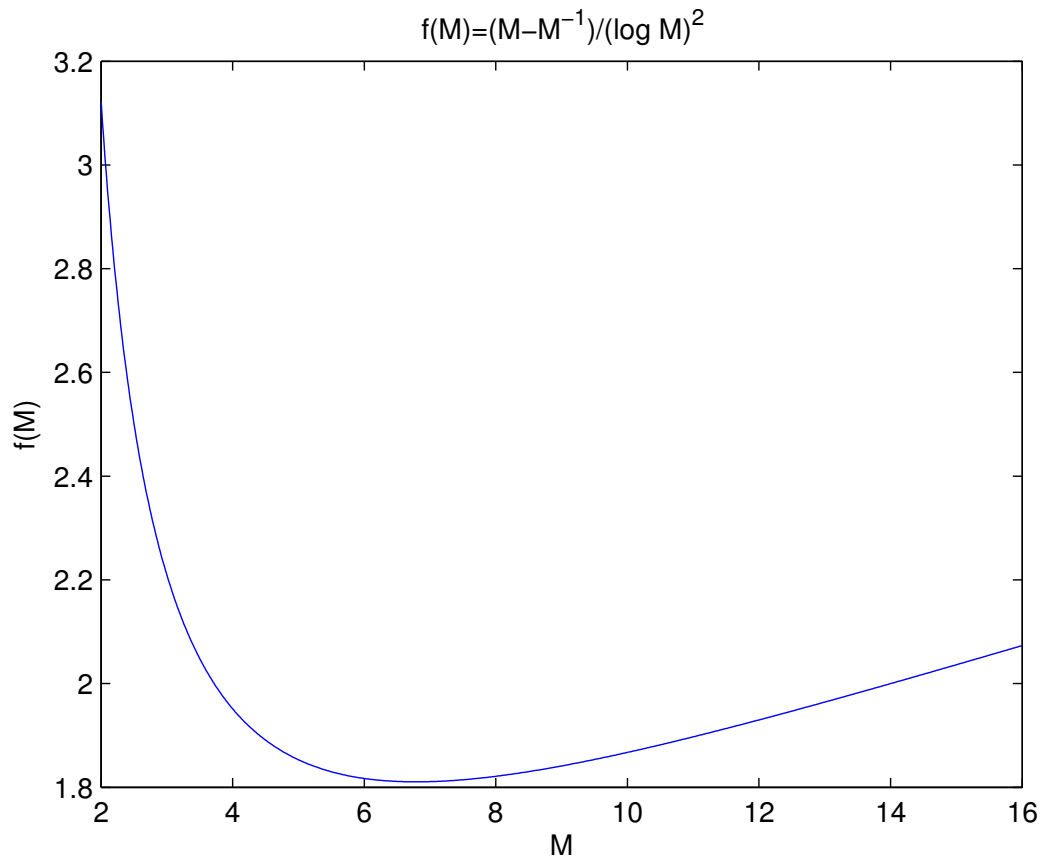


Abbildung 1: Optimales M (geometrisches Mittel)

Jetzt können die Gesamtrechenkosten C bestimmt werden

$$\begin{aligned} C &= \sum_{l=0}^L C_l \approx \sum_{l=0}^L 2\epsilon^{-2}(L+1)(M - M^{-1})c_0 \\ &= 2\epsilon^{-2}(L+1)^2(M - M^{-1})c_0. \end{aligned}$$

Mit der Wahl von $L = O(\frac{\log \epsilon^{-1}}{\log M})$ für $\epsilon \rightarrow 0$ folgt für die Rechenkosten

$$C \approx 2\epsilon^{-2}(\log \epsilon)^2 f(M),$$

mit $f(M) = \frac{M - M^{-1}}{(\log M)^2}$.

In Abbildung 1 ist erkennbar, dass sich das optimale M bei 7 befindet. Der optimale Wert für M ist allerdings nur als grobe Schätzung verwendbar, denn es reicht in (3.25) anstatt das geometrische, das arithmetische Mittel zu verwenden und bei analoger Rechnung

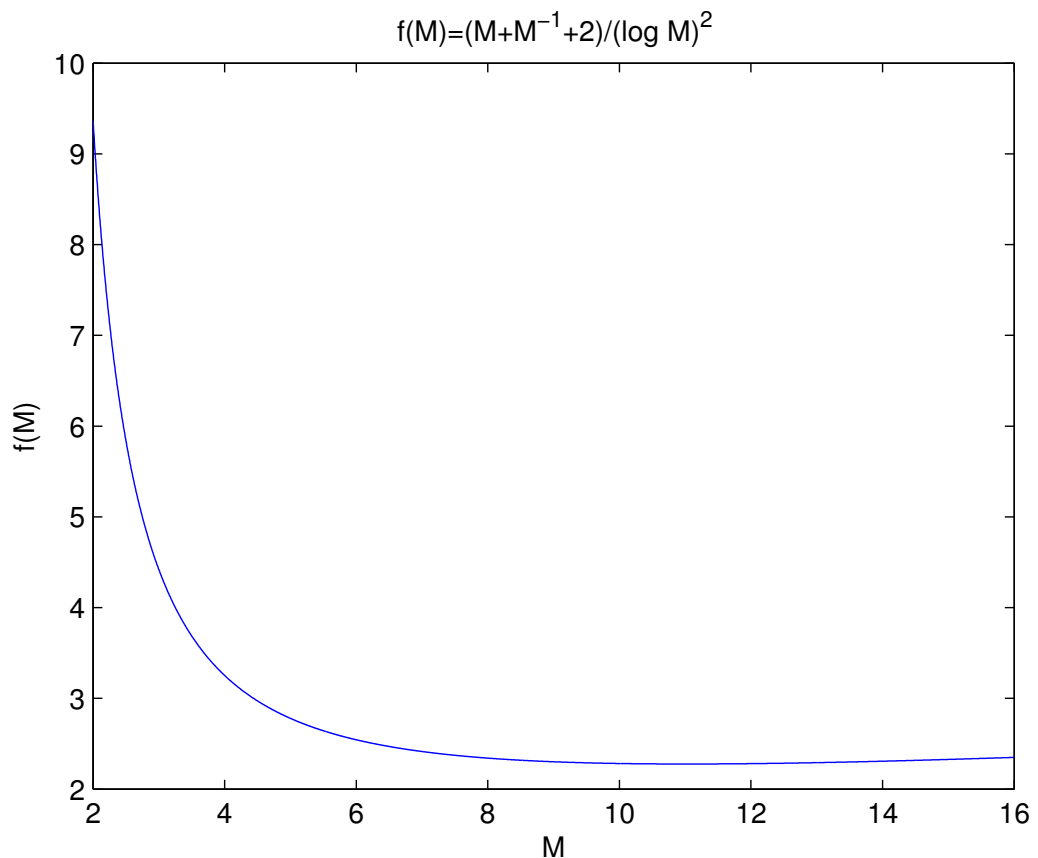


Abbildung 2: Optimales M (arithmetisches Mittel)

ergeben sich Rechenkosten $C \approx 2\epsilon^{-2}(\log \epsilon)^2 f(M)$ mit $f(M) = \frac{M+M^{-1}+2}{(\log M)^2}$. Somit ist zwar $M = 7$ weiterhin ein guter Wert, das optimale M liegt nun allerdings bei 11, wie in Abbildung 2 erkennbar ist.

Im nächsten Kapitel wird eine Konvergenzbedingung für den Algorithmus in Kapitel 4 aufgestellt. Aus dieser geht hervor, dass die Anzahl der zu simulierenden Level von M abhängt. Daher wird in Kapitel 5 bei der numerischen Bewertung von verschiedenen Optionen, $M = 4$ verwendet. Dadurch werden genügend Level zur Simulation benötigt, sodass die Eigenschaften der MLMC Methode deutlich werden.

3.4.2 Fehlerschätzung

Der erwartete Fehler $E[\hat{P}_l - P]$ wird nun durch die Samples $E[\hat{P}_l - \hat{P}_{l-1}]$ abgeschätzt. Dazu wird anfangs die entsprechende Verbindung hergestellt.

Weiterhin wird das Euler-Maruyama Verfahren zur Diskretisierung der SDE (2.1) verwendet und eine lipschitzstetige Auszahlungsfunktion vorausgesetzt, sodass für $l \rightarrow \infty$ nach Satz 3.3.1

$$E[P - \hat{P}_l] \approx c_1 h_l$$

gilt, mit einer Konstanten c_1 . Daraus ergibt sich

$$\begin{aligned} E[\hat{P}_l - \hat{P}_{l-1}] &= E[P - \hat{P}_{l-1}] - E[P - \hat{P}_l] \\ &\approx c_1 h_{l-1} - c_1 h_l \\ &= c_1 h_l M - c_1 h_l \\ &= c_1 h_l (M - 1) \\ &\approx (M - 1)E[P - \hat{P}_l]. \end{aligned} \tag{3.26}$$

Dies wird nun verwendet um eine Konvergenzbedingung für den Algorithmus in Kapitel 4 aufzustellen. Ziel ist es ein $MSE < \epsilon^2$ zu erhalten, daher wird

$$(E[P - \hat{P}_l])^2 < \frac{\epsilon^2}{2}$$

benötigt und mit (3.26) gilt

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

$$\begin{aligned}
&\implies |E[P - \hat{P}_l](M - 1)| < \frac{1}{\sqrt{2}}(M - 1)\epsilon \\
&\implies |E[\hat{P}_l - \hat{P}_{l-1}]| < \frac{1}{\sqrt{2}}(M - 1)\epsilon. \tag{3.27}
\end{aligned}$$

Dies ergibt die Konvergenzbedingung. Das aktuell feinste Level L muss also erhöht werden, bis der Schätzer \hat{Y}_L für $E[\hat{P}_l - \hat{P}_{l-1}]$ diese Bedingung erfüllt, d.h. bis

$$|\hat{Y}_L| < \frac{1}{\sqrt{2}}(M - 1)\epsilon \tag{3.28}$$

gilt. Damit die beiden feinsten Level die Bedingung (3.27) erfüllen, kann (3.28) durch die stärkere Bedingung

$$\max(M^{-1}|\hat{Y}_{L-1}|, |\hat{Y}_L|) < \frac{1}{\sqrt{2}}(M - 1)\epsilon \tag{3.29}$$

ersetzt werden. $M^{-1}|\hat{Y}_{L-1}|$ wird verwendet, da aus (3.26)

$\hat{Y}_L - \hat{Y}_{L-1} \approx (M - 1)E[P - \hat{P}_L] - (M - 1)E[P - \hat{P}_{L-1}]$ folgt und somit $\hat{Y}_L \approx M^{-1}\hat{Y}_{L-1}$.

3.4.3 Richardson Extrapolation

Die Richardson Extrapolation Methode eignet sich zur Konvergenzbeschleunigung vor allem dann, wenn eine Problemstellung zwei Diskretisierungen zu unterschiedlichen Schrittweiten erfordert. Indem der Fehler durch seine einzelne Fehlerterme dargestellt wird, ist es mit der Richardson Extrapolationsmethode möglich den Fehler führender Ordnung zu eliminieren. Angewandt auf die MLMC Methode wird somit der erwartete Fehler $E[P - \hat{P}_l]$ für $l \rightarrow \infty$ von $O(h_l)$ auf $O(h_l^2)$ gesenkt (vgl. [11], Kapitel 3.3.2). Der Schätzer \hat{Y} ändert sich dabei zu

$$\left(\sum_{l=0}^L \hat{Y}_l\right) + (M - 1)^{-1}\hat{Y}_L = \frac{M}{M - 1} \left(\hat{Y}_0 + \sum_{l=1}^L (\hat{Y}_l - M^{-1}\hat{Y}_{l-1})\right).$$

Der reduzierte Fehler wirkt sich nun auf die Konvergenz aus und mit ähnlichen Rechnungen wie in Kapitel 3.4.2 ergibt sich folgende Konvergenzbedingung

$$|\hat{Y}_L - \frac{1}{M}\hat{Y}_{L-1}| < \frac{1}{\sqrt{2}}(M^2 - 1)\epsilon. \tag{3.30}$$

Diese kann anstatt der Konvergenzbedingung (3.29) verwendet werden. In Kapitel 5 wird für verschiedene Optionstypen geprüft, ob durch Richardson Extrapolation Rechenkosten eingespart werden können.

4 Implementierung

Nun wird gezeigt, wie der Multilevel Monte-Carlo Algorithmus unter Verwendung der bisherigen Ergebnisse implementiert wird.

1. Start mit $L=0$
2. schätze V_l unter Verwendung von $N_L = 10^4$ Samples
3. errechne das optimale N'_l für $l = 0, 1, \dots, L$ mit Gleichung (4.1)
4. falls $N'_l > N_l$, berechne weitere $N'_l - N_l$ Samples und schätze V_l neu für $l = 0, 1, \dots, L$
5. setze $N_l = N'_l$
6. falls $L \geq 2$ führe Konvergenztest (3.29) oder (3.30) durch und falls die entsprechende Bedingung erfüllt ist, breche Algorithmus ab
7. falls $L < 2$ oder falls Konvergenztest aus Schritt 6 nicht erfüllt ist, setze $L = L + 1$ und gehe zu Schritt 2

In Schritt 3 soll N_l so gewählt werden, dass $Var[\hat{Y}] < \frac{\epsilon^2}{2}$ ist. Die Konvergenzbedingungen versuchen sicherzustellen, dass $(E[P - \hat{P}_L])^2 < \frac{\epsilon^2}{2}$ ist. Demnach wäre nach (3.1) ein $MSE < \epsilon^2$ erreicht.

$$N_l = \left\lceil 2\epsilon^{-2} \sqrt{V_l h_l} \left(\sum_{l=0}^L \sqrt{V_l/h_l} \right) \right\rceil \quad (4.1)$$

erfüllt die Bedingung, dies lässt sich mit Gleichung (3.7) bestätigen, demnach ist

$$Var[\hat{Y}] = \sum_{l=0}^L N_l^{-1} V_l$$

und somit gilt

$$Var[\hat{Y}] \leq \sum_{l=0}^L V_l \left(2\epsilon^{-2} \sqrt{V_l h_l} \left(\sum_{l=0}^L \sqrt{(V_l/h_l)} \right) \right)^{-1}$$

$$\begin{aligned}
&= \frac{\epsilon^2}{2} \left(\sum_{l=0}^L V_l (V_l h_l)^{-\frac{1}{2}} \left(\sum_{l=0}^L \sqrt{(V_l/h_l)} \right)^{-1} \right) \\
&= \frac{\epsilon^2}{2} \left(\sum_{l=0}^L \sqrt{V_l/h_l} \right) \left(\sum_{l=0}^L \sqrt{(V_l/h_l)} \right)^{-1} \\
&= \frac{\epsilon^2}{2}.
\end{aligned}$$

Weiterhin werden die Rechenkosten C benötigt, diese ergeben sich nach Kapitel 3.2 durch

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

Um die Multilevel Monte-Carlo Methode mit der Monte-Carlo Methode vergleichen zu können wird das entsprechende N_l aus Schritt 3 für das Monte-Carlo Verfahren benötigt, deswegen wird

$$N_l = 2\epsilon^{-2} \text{Var}[\hat{P}_l] \quad (4.2)$$

gesetzt. Eine kurze Rechnung zeigt, dass auch hier die gewünschte Schranke von $\frac{\epsilon^2}{2}$ für die Varianz erreicht wird

$$\begin{aligned}
\text{Var}[\hat{Y}] &= \text{Var} \left[\sum_{l=0}^L \left(\frac{1}{N_l} \sum_{i=1}^{N_l} \hat{P}_l^{(i)} \right) \right] \\
&= \text{Var} \left[\frac{\sum_{l=0}^L \sum_{i=1}^{N_l} \hat{P}_l^{(i)}}{\sum_{l=0}^L N_l} \right] \\
&= \frac{\sum_{l=0}^L \sum_{i=1}^{N_l} \text{Var}[\hat{P}_l^{(i)}]}{\left(\sum_{l=0}^L N_l \right)^2} \\
&= \frac{\sum_{l=0}^L N_l \text{Var}[\hat{P}_l]}{\left(\sum_{l=0}^L N_l \right)^2} \\
&\stackrel{(4.2)}{=} \frac{1}{2} \epsilon^2 \underbrace{\frac{\sum_{l=0}^L N_l^2}{\left(\sum_{l=0}^L N_l \right)^2}}_{\leq 1} \\
&\leq \frac{1}{2} \epsilon^2.
\end{aligned}$$

Und für die Rechenkosten C gilt bei der Monte-Carlo Methode $C = \sum_{l=0}^L N_l h_l^{-1}$.

Die Vorgehensweise bei der Implementierung der MC Methode ist nun ähnlich. Anstatt der Samples $\hat{P}_l - \hat{P}_{l-1}$ werden nun die Payoffs \hat{P}_l zu jedem Level berechnet, bis die modifizierten Konvergenzbedingungen erfüllt sind.

Die Konvergenzbedingung ohne Richardson Extrapolation wird angepasst zu

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

und mit Richardson Extrapolation zu

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

Zu beachten ist, dass es sich hierbei um einen heuristischen Algorithmus handelt. Denn die Schwachstelle des MLMC Algorithmus ist, dass die Konstanten c_1 und c_2 (siehe Komplexitätstheorem) erst im Algorithmus geschätzt werden. Nach (3.1) setzt sich der MSE aus dem quadrierten Bias und der Varianz des Schätzers \hat{Y} zusammen. Oben wurde zwar gezeigt, dass die Varianz die geforderte Schranke von $\frac{\epsilon^2}{2}$ erfüllt. Für den quadrierten Bias ist dies allerdings nicht sicher. Somit ist ein MSE der Größe $O(\epsilon^2)$ nicht garantiert.

5 Numerische Ergebnisse

Die Multilevel Monte-Carlo Methode wird nun mit der Monte-Carlo Methode verglichen, indem verschiedene Optionstypen mit beiden Verfahren numerisch bewertet werden. Dabei wird das Euler-Maruyama Verfahren zur Diskretisierung der Black Scholes SDE verwendet. Im Black Scholes Modell wird der Drift μ der risikolosen Zinsrate gleichgesetzt, sodass nach Satz 2.2.2 der faire Optionspreis der diskontierten Auszahlung entspricht. Der Parameter M , wurde wie bereits in Kapitel 3.4.1 erwähnt, gleich 4 gesetzt.

5.1 Europäische Option

Zuerst werden beide Methoden anhand einer europäischen Option mit einjähriger Laufzeit verglichen. Die diskontierte Auszahlungsfunktion dazu ist

$$P = \exp(-r) \max(0, S(1) - K).$$

Der Kassapreis der Aktie sei $S(0) = 1$, der Ausübungspreis $K = 1$, die Volatilität $\sigma = 0.2$ und die risikolose Zinsrate $r = 0.05$.

In Abbildung 3 sind die numerischen Resultate der beiden Methoden gegenübergestellt. Im linken oberen Plot wird der Logarithmus (zur Basis M) von der Varianz der Samples $\hat{P}_l - \hat{P}_{l-1}$ bzw. der Payoffs \hat{P}_l gegenüber den einzelnen Level dargestellt. Der Graph der Samples zeigt eine Steigung von -1 und aus

$$\begin{aligned} & \frac{\log_M(\text{Var}[\hat{P}_l - \hat{P}_{l-1}]) - \log_M(\text{Var}[\hat{P}_0])}{l} = -1 \\ \Leftrightarrow & \log_M \left(\frac{\text{Var}[\hat{P}_l - \hat{P}_{l-1}]}{\text{Var}[\hat{P}_0]} \right) = -l \\ \Leftrightarrow & M^{\log_M \left(\frac{\text{Var}[\hat{P}_l - \hat{P}_{l-1}]}{\text{Var}[\hat{P}_0]} \right)} = M^{-l} \\ \Leftrightarrow & \text{Var}[\hat{P}_l - \hat{P}_{l-1}] = M^{-l} \text{Var}[\hat{P}_0] \end{aligned}$$

folgt mit $M^{-l} = h_l$ eine zu h_l proportionale Varianz. Dies stimmt mit den theoretischen Ergebnissen überein. In Kapitel 3 wurde $\text{Var}[\hat{P}_l - \hat{P}_{l-1}] = O(h_l)$ ermittelt. Die Varianz der Samples $\hat{P}_l - \hat{P}_{l-1}$ beim MLMC Verfahren ist für das Level $l = 2$ bereits 200 mal kleiner als die Varianz von \hat{P}_l beim Monte-Carlo Verfahren und zum Level $l = 3$ sogar 800 mal.

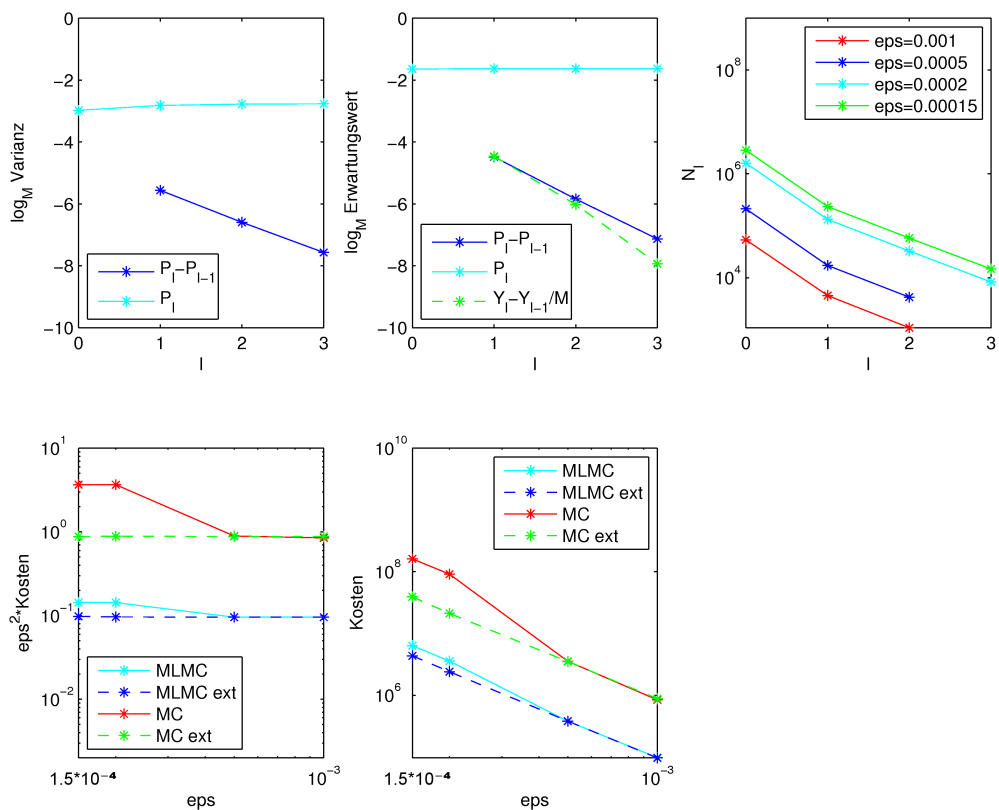


Abbildung 3: Europäische Option

Der mittlere obere Plot stellt den Logarithmus vom Erwartungswert der Samples bzw. der Payoffs zu den einzelnen Level dar. Die Steigung des Graphen der Samples $\hat{P}_l - \hat{P}_{l-1}$ von ungefähr -1 deutet mit einer ähnlichen Rechnung wie für die Varianz wieder auf eine $O(h_l)$ Konvergenz von $E[\hat{P}_l - \hat{P}_{l-1}]$ hin. Dies impliziert mit (3.26) eine schwache Konvergenz der Ordnung 1 und entspricht den Ergebnissen aus Kapitel 3. Der Graph $\hat{Y}_l - \hat{Y}_{l-1}/M$ zu der MLMC Methode mit Richardson Extrapolation zeigt nochmals eine deutliche Verbesserung. So wird beinahe eine aus Kapitel 3.4.3 erwartete schwache Konvergenz der Ordnung 2 erreicht.

Im rechten oberen Plot wird die Anzahl der benötigten Samples zu den einzelnen Level für verschiedene Werte von ϵ abgebildet, die bei der MLMC Methode nötig waren um die Konvergenzbedingung (3.29) zu erfüllen. Für jedes Level nimmt die Anzahl der notwendigen Simulationen mit kleiner werdendem ϵ zu und für jedes ϵ werden mit steigendem Level weniger Samples benötigt.

Im linken unteren Plot werden die Rechenkosten verglichen. Dazu wird $\epsilon^2 C$ mit ϵ ins

Verhältnis gesetzt. Es ist erkennbar, dass für das MLMC Verfahren bei gleicher geforderter Genauigkeit deutlich weniger Rechnungen als für die MC Methode notwendig sind und dass die Richardson Extrapolation bei beiden Verfahren eine Verbesserung ab $\epsilon = 0.0002$ erbringt. Für $\epsilon = 0.00015$ benötigt die MC Methode mit Richardson Extrapolation, 10 mal so viele Berechnungen wie die MLMC Methode. Ohne Richardson Extrapolation sind es sogar 25 mal so viele Rechnungen. Der starke Anstieg bei $\epsilon = 0.0005$ wird von der Erhöhung der benötigten Level ab $\epsilon = 0.0002$ verursacht. Für $\epsilon = 0.0005$ waren bei den Berechnungen nur 3 Level nötig, bei $\epsilon = 0.0002$ bereits 4 Level. Im mittleren unteren Plot sind die reinen Rechenkosten zur geforderten Genauigkeit ϵ abgebildet.

Der Wert dieser Option auf vier Nachkommastellen gerundet, ist 0.1045. Der mit der MLMC Methode approximierte Optionswert lag bei 0.1044 für $\epsilon = 0.00015$. Wie bereits in Kapitel 4 erwähnt, ist nicht garantiert, dass dieser Algorithmus die geforderte Genauigkeit erbringt. Daher ist es wichtig, die Verlässlichkeit der Ergebnisse zu überprüfen. Dazu wird der root mean square error (RMSE) mit der geforderten Genauigkeit ϵ verglichen. Um den MSE zu bestimmen wird Gleichung (3.1) verwendet und nach (3.26) kann $(E[\hat{Y} - Y])^2$ durch $(E[\hat{P}_L - \hat{P}_{L-1}]/(M-1))^2$ bzw. unter Verwendung der Richardson Extrapolation durch $(E[\hat{P}_L - \hat{P}_{L-1}]/(M^2-1))^2$ (vgl. [11]) abgeschätzt werden. Der RMSE/ϵ befand sich bei der MLMC Methode für alle Werte von ϵ bei 0.7. Auch mit Richardson Extrapolation scheinen die Ergebnisse verlässlich. Der RMSE/ϵ lag auch hier bei 0.7.

5.2 Asiatische Option

In diesem Kapitel wird eine asiatische Option betrachtet. Die diskontierte Auszahlungsfunktion ist gegeben durch

$$P = \exp(-r) \max(0, \bar{S} - K),$$

wobei $\bar{S} = \int_0^1 S(t) dt$ durch $\bar{S}_l = \sum_{n=1}^{N_l} \frac{1}{2} (\hat{S}_n + \hat{S}_{n-1}) h_l$ approximiert wird. Wie bei der europäischen Option sei wieder $S(0) = 1$, $K = 1$, $\sigma = 0.2$ und $r = 0.05$.

Die drei oberen Plots in Abbildung 4 zeigen ähnliche Ergebnisse wie die der europäischen Option. Im linken oberen Plot weist der Graph der Samples $\hat{P}_l - \hat{P}_{l-1}$ sogar eine Steigung von -1.65 auf, dies deutet bzgl. der Varianz eine schnellere Konvergenz als $O(h_l)$ an. Bei $l = 3$ ist die Varianz der Samples 400 mal kleiner als die Varianz der Payoffs. Auch beim Graph der Samples im mittlerem oberen Plot ist eine Steigung von -1.65 gegeben,

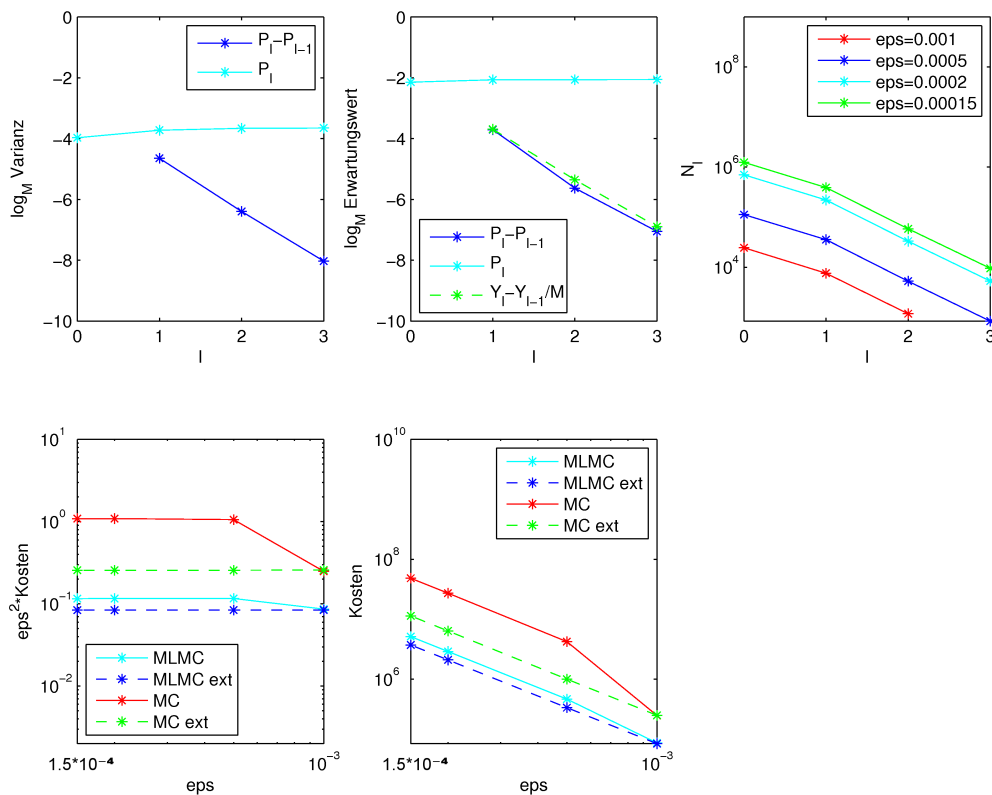


Abbildung 4: Asiatische Option

sodass das MLMC Verfahren in diesem Beispiel eine schwache Konvergenz von größerer Ordnung als 1 besitzt.

Im unteren linken Plot wird wieder deutlich, dass die MLMC Methode weniger Rechnungen benötigt, als die MC Methode. Ohne Richardson Extrapolation, ist bei der MC Methode der Rechenaufwand 10 mal so hoch wie bei der MLMC Methode, bei einer geforderten Genauigkeit von $\epsilon = 0.00015$. Mit Richardson Extrapolation scheint eine Reduzierung der Rechenkosten einzutreten, allerdings wurde die in Kapitel 3 errechnete $O(h_l^2)$ Konvergenz (siehe mittlerer oberer Plot) des erwarteten Fehlers nicht erreicht. Deswegen sind die Ergebnisse zu hinterfragen, da die im Algorithmus verwendete Konvergenzbedingung auf dieser Konvergenzgeschwindigkeit aufbaut. Im mittleren unteren Plot ist wieder ein Gesamtüberblick über den Rechenaufwand gegeben.

Der genaue Wert dieser Option beträgt auf vier Nachkommastellen gerundet 0.0576. Für $\epsilon = 0.00015$ simulierte die MLMC Methode einen Optionswert von 0.0574. Der RMSE/ϵ liegt zwischen 0.68 und 0.79 mit, sowie ohne Richardson Extrapolation.

5.3 Multi-Asset Optionen

Im Folgenden wird die Multilevel Monte-Carlo Methode bzgl. Multi-Asset Optionen getestet. Dazu werden zwei verschiedene Optionen betrachtet.

5.3.1 Geometrischer Basket

Zuerst wird ein geometrischer Basket mit einjähriger Laufzeit beobachtet. Die Auszahlungsfunktion dieser Option ist

$$P = \exp(-r) \max \left(0, \left(\prod_{i=1}^3 S_i(1) \right)^{\frac{1}{3}} - K \right).$$

Die risikolose Zinsrate sei $r = 0.05$, der Ausübungspreis $K = 1$ und die Kassapreise $S_i = 1$ für $i = 1, 2, 3$. Die zugehörigen Volatilitäten betragen $\sigma_1 = 0.1, \sigma_2 = 0.15$ und $\sigma_3 = 0.2$. Die Korrelation der Brownschen Bewegungen sei jeweils 0.25.

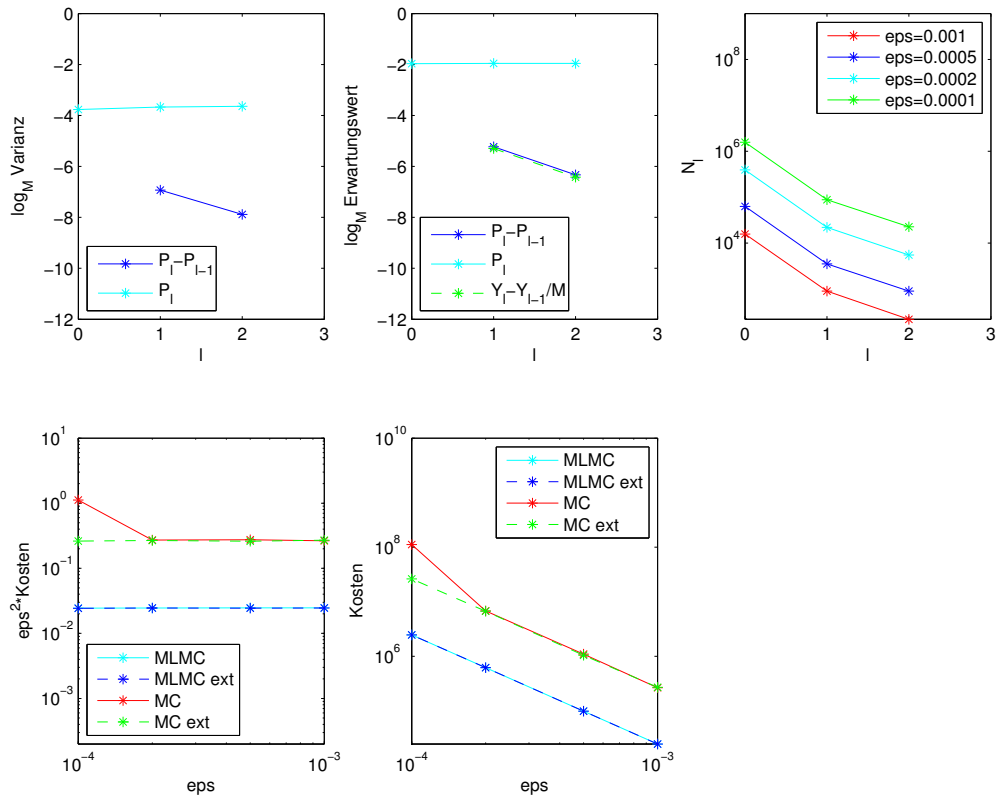


Abbildung 5: Geometrischer Basket

Im linken oberen Plot in Abbildung 5 ist wieder der Logarithmus von der Varianz der Samples $\hat{P}_l - \hat{P}_{l-1}$ bzw. der Payoffs \hat{P}_l gegenüber den einzelnen Level dargestellt. Die Steigung des Graphen der Samples beträgt -1, sodass $Var[\hat{P}_l - \hat{P}_{l-1}] = O(h_l)$ folgt. Bei $l = 2$ ist die Varianz der Samples bereits 360 mal kleiner als die Varianz der Payoffs.

In der mittleren oberen Grafik ist der Logarithmus vom Erwartungswert der Samples bzw. der Payoffs gegenüber den Levels geplottet. Der Graph der Samples weist eine Steigung von -1 auf. Daraus folgt $E[\hat{P}_l - \hat{P}_{l-1}] = O(h_l)$ und eine schwache Konvergenz der Ordnung 1.

Der rechte obere Plot zeigt ähnliche Ergebnisse wie die vorigen Beispiele. In den unteren Plots sind die Rechenkosten dargestellt. Die MLMC Methode reduziert den Rechenaufwand gegenüber der MC Methode bei $\epsilon = 0.0001$ um den Faktor 45. Für die MLMC Methode ergeben sich auf diesem Genauigkeitslevel keine weiteren Vorteile durch Richardson Extrapolation. Bei einer geforderten Genauigkeit von $\epsilon = 0.0001$ errechnete die MLMC Methode einen Optionswert von 0.0665. Ohne Richardson Extrapolation deutet ein $RMSE/\epsilon$ zwischen 0.70 und 0.89 verlässliche Ergebnisse an.

5.3.2 Arithmetischer Basket

Als zweite Multi-Asset Option wird ein arithmetischer Basket betrachtet, dessen Auszahlungsfunktion durch

$$P = \exp(-r) \max\left(\frac{1}{3} \sum_{i=1}^3 S_i(1) - K, 0\right)$$

gegeben ist. Die Eingabeparameter sind identisch zum geometrischen Basket, allerdings wird jetzt eine Korrelation der Brownschen Bewegungen von -0.25 angenommen. Die numerischen Ergebnisse sind in Abbildung 6 wiedergegeben. Der linke obere Plot zeigt, wie bereits im letzten Beispiel, eine Steigung der Samples von -1, dies führt zu $Var[\hat{P}_l - \hat{P}_{l-1}] = O(h_l)$. Der Graph der Samples in der mittleren oberen Grafik weist eine Steigung von -1 auf und somit folgt eine schwache Konvergenz von $O(h_l)$. Durch Richardson Extrapolation wird keine Verbesserung der schwachen Konvergenz bewirkt. Somit sind die Ergebnisse bzgl. Richardson Extrapolation nicht verlässlich, da die erwartete Konvergenzgeschwindigkeit von $O(h_l^2)$ nicht erreicht wird.

In den unteren Plots ist zu erkennen, dass die MLMC Methode den Rechenaufwand deutlich reduziert. Durch Richardson Extrapolation kann dieser nochmals verringert werden. Ohne Richardson Extrapolation benötigt das MC Verfahren, bei einer geforderten Genauigkeit von $\epsilon = 0.0001$, 20 mal mehr Rechnungen als das MLMC Verfahren.

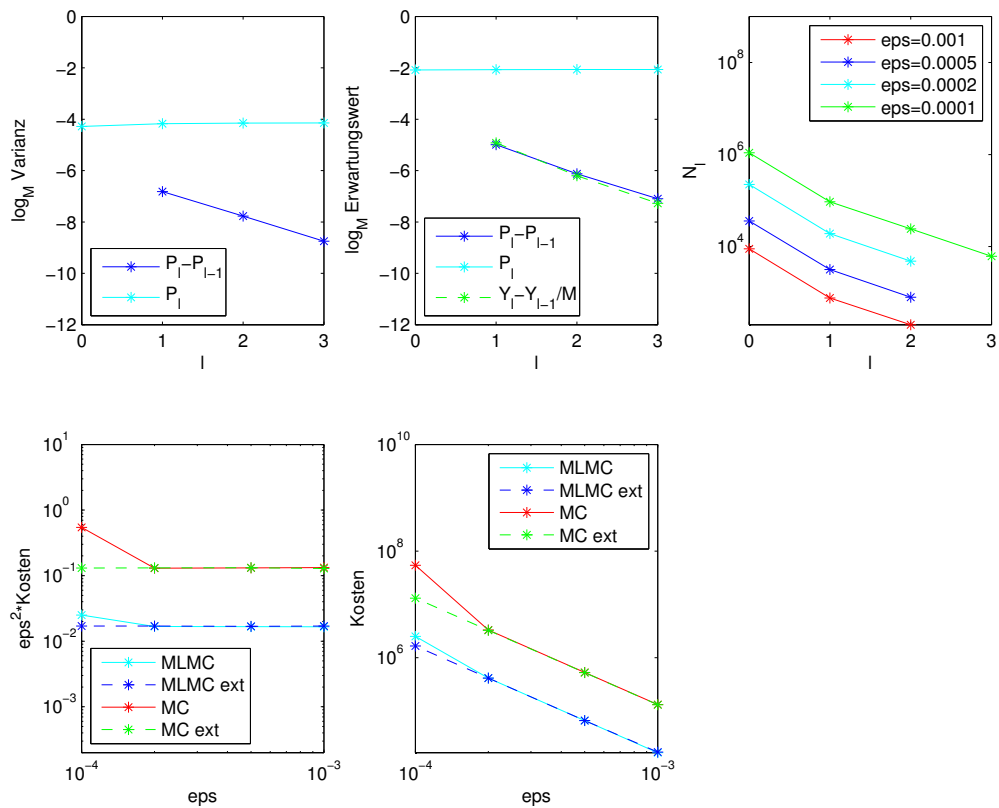


Abbildung 6: Arithmetischer Basket

Mit Richardson Extrapolation betragen die Rechenkosten noch das 8-fache der MLMC Methode.

Der ermittelte approximierte Optionspreis liegt bei 0.0571. Ohne Richardson Extrapolation befindet sich der $RMSE/\epsilon$ zwischen 0.71 und 0.79.

5.4 Heston stochastic volatility Model

Als letztes wird dieselbe europäische Option wie in 5.1 betrachtet. Anstatt der Black Scholes SDE folgt der Aktienkurs allerdings diesmal dem Heston Model (vgl. [12])

$$\begin{aligned} dS &= rSdt + \sqrt{V}SdW_1, & 0 < t < 1, \\ dV &= \lambda(\sigma^2 - V)dt + \xi\sqrt{V}dW_2. \end{aligned}$$

Die Eingabeparameter sind $S(0) = 1, r = 0.05, \sigma = 0.2, K = 1, V(0) = 0.04, \lambda = 5, \xi = 0.25$ und $p = -0.5$, wobei p die Korrelation der beiden Brownschen Bewegungen W_1 und W_2 bezeichnet.

Um die Genauigkeit und die Varianz zu verbessern (vgl. [14]), wird eine weitere Variable

$$Q = e^{\lambda t}(V - \sigma^2)$$

eingeführt. Anschließend werden die SDE's von Q und S mit dem Euler-Maruyama Verfahren diskretisiert (vgl. [14] und [5]). Daraus ergeben sich die Gleichungen

$$\begin{aligned}\hat{S}_{n+1} &= \hat{S}_n + r\hat{S}_nh + \sqrt{\hat{V}_n^+}\hat{S}_n\Delta W_{1,n}, \\ \hat{V}_{n+1} &= \sigma^2 + e^{-\lambda h}\left((\hat{V}_n - \sigma^2) + \xi\sqrt{\hat{V}_n^+}\Delta W_{2,n}\right).\end{aligned}$$

Wobei zu beachten ist, dass \sqrt{V} durch $\sqrt{V_n^+} = \sqrt{\max(V, 0)}$ ersetzt wurde. Für die gewählten Parameter λ, ξ und σ liegt allerdings die Wahrscheinlichkeit, dass eine negative Volatilität im Diskretisierungsverfahren vorkommt, bei 0 für $h \rightarrow 0$. Der Beweis hierzu ist aufwändig, daher wird an dieser Stelle nur auf [14] verwiesen.

In Abbildung 7 sind nun die numerischen Ergebnisse dargestellt. So lässt sich zwar auf diesem Genauigkeitslevel ($\epsilon = 0.0002$) über die Varianz kaum eine Aussage treffen, allerdings weist der Graph des Erwartungswertes der Samples eine Steigung von -1.35 auf, sodass dies eine schwache Konvergenz schneller als $O(h_l)$ andeutet.

Der obere rechte Plot zeigt wieder ähnliche Ergebnisse wie zuvor. Anhand des unteren linken Plots ist zu erkennen, dass ohne Richardson Extrapolation die MC Methode 12 mal so viele Rechnungen benötigt wie die MLMC Methode, bei einer geforderten Genauigkeit von $\epsilon = 0.0002$.

Die Verwendung der Richardson Extrapolation scheint zwar eine Reduzierung der Rechenkosten zu bewirken, allerdings wurde in der Konvergenzbedingung des Algorithmus angenommen, dass der verbleibende Fehler nach Richardson Extrapolation der Ordnung 2 entspricht. Dies konnte im mittleren oberen Plot nicht bestätigt werden. Ohne Richardson Extrapolation deutet ein RMSE/ϵ von ca. 0.7 verlässliche Ergebnisse an. Als approximierter Optionswert, bei einer geforderten Genauigkeit von $\epsilon = 0.0002$, ergibt sich mit der MLMC Methode 0.1044.

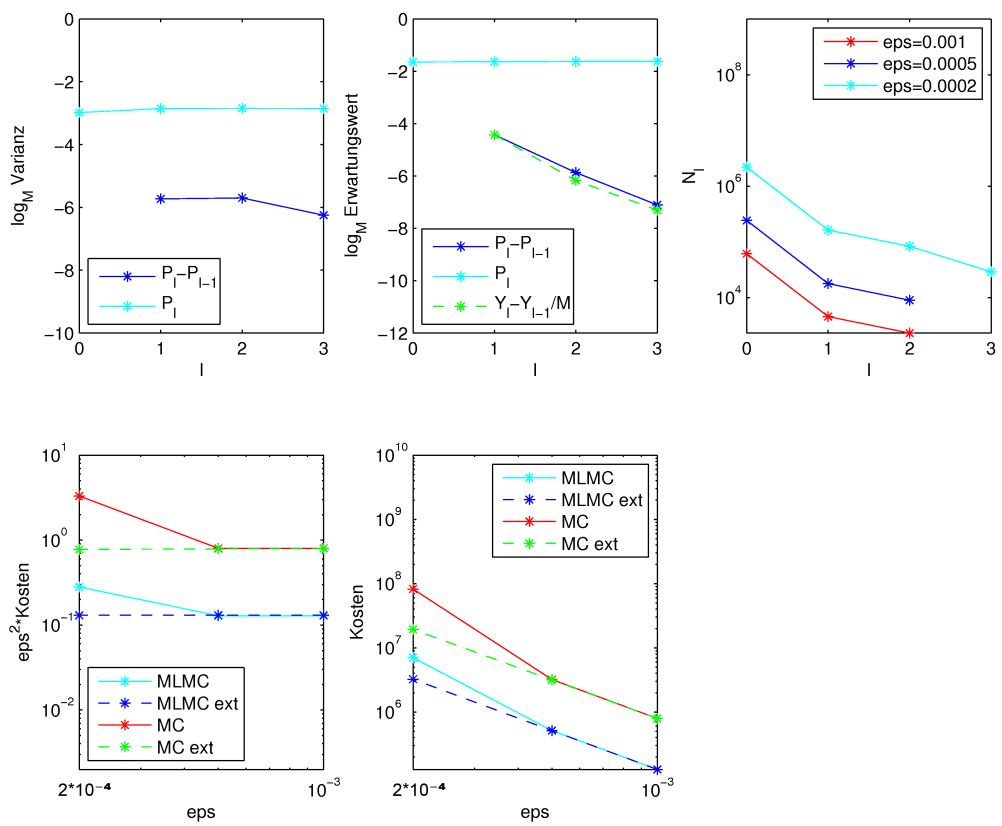


Abbildung 7: Europäische Option mit Heston Model

6 Schlussbetrachtungen und Ausblick

Ziel dieser Arbeit war es zu zeigen, dass die Multilevel Monte-Carlo Methode eine Verbesserung gegenüber der Standard Monte-Carlo Methode bzgl. Multi-Asset Optionen bewirkt. Die Ergebnisse in Kapitel 5.3 zeigen dies deutlich. Der Rechenaufwand konnte teilweise auf ein zwanzigstel reduziert werden. Auch die Verlässlichkeit der Ergebnisse scheint gewährt, da ein RMSE/ϵ deutlich kleiner 1 für alle Genauigkeiten erzielt wurde. Um die Effizienz der MLMC Methode für Multi-Asset Optionen testen zu können, waren allerdings die Ergebnisse aus Kapitel 3 nötig. Zuerst wurden in Kapitel 3.1 und 3.2 die wesentlichen Merkmale des MLMC Verfahrens erfasst, sodass schließlich die Rechenkosten abschätzbar waren. Unter Verwendung des Euler-Maruyama Verfahrens und einer Lipschitzstetigen Auszahlungsfunktion wurde eine schwache Konvergenz der Ordnung 1 ermittelt. Die numerischen Resultate für europäische, asiatische und Multi-Asset Optionen bestätigten diese Konvergenzordnung oder zeigten sogar bessere Konvergenzeigenschaften. Auch für die Varianz der Samples stimmten die theoretischen Ergebnisse aus Kapitel 3 mit den numerischen Resultaten in Kapitel 5 überein. Anschließend wurden im Komplexitätstheorem die Aussagen verallgemeinert. Kapitel 3.4 stellte dann eine Konvergenzbedingung für den späteren Algorithmus auf, indem der Fehler abgeschätzt wurde. Weiterhin wurde untersucht, ob durch Richardson Extrapolation der Rechenaufwand reduziert wird. Allerdings konnte nur bei dem Beispiel zur europäischen Option mit verlässlichen Ergebnissen gezeigt werden, dass die Rechenkosten sinken. Bei den restlichen Beispielen fand zwar meistens eine Verbesserung statt, aber die Verlässlichkeit der Ergebnisse konnte nicht bestätigt werden. Ohne Verwendung der Richardson Extrapolation wurde allerdings für alle untersuchten Optionsarten gezeigt, dass der Multilevel Monte-Carlo Algorithmus effizient und verlässlich bzgl. der geforderten Genauigkeit ist und die Rechenkosten gegenüber der Monte-Carlo Methode stets bedeutend sinken.

In dieser Arbeit wurde stets das Euler Maruyama Verfahren zur Diskretisierung des Aktienkurses verwendet. Das Komplexitätstheorem zeigt, dass durch Verwendung eines Diskretisierungsverfahrens mit besseren Konvergenzeigenschaften die Rechenkosten auf $O(\epsilon^2)$ reduzierbar sind. Die Milstein Diskretisierung bewirkt diesen Effekt (vgl. [6] und [7]). Eine weitere Möglichkeit die Multilevel Monte-Carlo Methode zu verbessern, ist die Verwendung von Quasi-Zufallszahlen. So ist es möglich, durch Kombination mit der Quasi-Monte-Carlo Methode, den Rechenaufwand weiter zu reduzieren (vgl. [8]).

Bisher wurde die MLMC Methode nur für wenige Optionen getestet. Die Monte-Carlo Methode besitzt allerdings gerade in der Vielfältigkeit ihre Stärke. Sie ist in der Lage

mit wenigen Veränderungen in der Auszahlungsfunktion jeden Optionstyp zu bewerten. Daher wird ein Vergleich der MLMC Methode mit der Standard MC Methode für weitere Optionen sehr interessant sein.

A Anhang: Programmcodes

Im Folgenden ist der Programmcode des MLMC Algorithmus zur Bewertung einer europäischen Option in Matlab gegeben. Zur Bewertung anderer Optionen muss lediglich die Auszahlungsfunktion angepasst werden. Soll die Richardson-Extrapolations Methode zur Konvergenzbeschleunigung verwendet werden, ist nur die Konvergenzbedingung zu ändern. Zuerst wird das Hauptprogramm angegeben und anschließend die verwendeten Unterprogramme.

'Hauptprogramm mlmc

```
function Wert = mlmc(S0, SP, T, M, mu, sigma, eps)
```

```
'Eingabeparameter:
```

```
'SO=Kassapreis
```

```
'SP=Ausübungspreis
```

```
'T=Laufzeit
```

```
'M=Paramter M (siehe Kapitel 3.4.2)
```

```
'mu = Drift bzw. risikoloser Zinssatz
```

```
'sigma = Volatilität
```

```
'eps = geforderte Genauigkeit (siehe MSE)
```

```
'Ausgabe: Wert = Preis der Option
```

```
'Der Fall  $L = 1$  muss gesondert behandelt werden, da anstatt Samples
```

```
' $P_l - P_{l-1}$  Payoffs  $P_l$  berechnet werden.
```

```
'Zuerst werden 10000 Samples berechnet, mlmceulerauswertung2 ist dabei
```

```
'eine Unterfunktion zur Auswertung von  $P_0$ .
```

```
N(1) = 10000;
```

```
for i = 1 : N(1)
```

```
y = mlmceulerauswertung2(S0, SP, T, mu, sigma);
```

```
X(1, i) = y(1);
```

```
end
```

```
' Berechnung der Varianz
```

```
V(1)=var(X(1, 1 : N(1)));
```

```
'Berechnung des optimalen  $N_l$ 
```

```
h(1) = T;
```

```
s = sqrt(V(1)/h(1));
```

```
Nneu(1) = ceil(2 * sqrt(V(1) * h(1)) * s/eps^2);
```

```

'Berechnung weiterer Samples
if  $N_{neu}(1) > N(1)$ 
for  $i = 10001 : N_{neu}(1)$ 
 $y = \text{mlmceulerauswertung2}(S0, SP, T, mu, sigma);$ 
 $X(1, i) = y(1);$ 
end
 $V(1) = \text{var}(X(1, 1 : N_{neu}(1)));$ 
else
end
 $N(1) = N_{neu}(1);$ 
'Die Variable  $K$  wird nur verwendet um den Algorithmus abubrechen, falls
'die Konvergenzbedingung erfüllt ist.
 $K = 0;$ 
 $L = 2;$ 
while ( $K < 1$ )
 $N(L) = 10000;$ 
'mlmceulerauswertung ist eine Unterfunktion zur Auswertung von
' $P_L - P_{L-1}$ ,  $y(1)$  entspricht  $P_L$  und  $y(2)$  entspricht  $P_{L-1}$ .
for  $i = 1 : N(L)$ 
 $y = \text{mlmceulerauswertung}(S0, SP, T, M, L, mu, sigma);$ 
 $X(L, i) = y(1) - y(2);$ 
end
 $V(L) = \text{var}(X(L, 1 : N(L)));$ 
 $h(L) = M^{-L+1} * T;$ 
 $s = 0;$ 
for  $l = 1 : L$ 
 $s = s + \text{sqrt}(V(l)/h(l));$ 
end
for  $l = 1 : L$ 
 $N_{neu}(l) = \text{ceil}(2 * \text{sqrt}(V(l) * h(l)) * s/eps^2);$ 
'Berechnung weiterer Samples falls nötig
if ( $N_{neu}(l) > N(l)$  and  $N_{neu}(l) > 10000$ )
if  $l < 2$ 
for  $i = \max(10001, N(l) + 1) : N_{neu}(l)$ 
 $y = \text{mlmceulerauswertung2}(S0, SP, T, mu, sigma);$ 

```

```

 $X(l, i) = y(1);$ 
end
else
for  $i = \max(10001, N(l) + 1) : Nneu(l)$ 
 $y = \text{mlmceulerauswertung}(S0, SP, T, M, l, mu, sigma);$ 
 $X(l, i) = y(1) - y(2);$ 
end
end
 $V(l) = \text{var}(X(l, 1 : Nneu(l)));$ 
else
end
 $N(l) = Nneu(l);$ 
end
'Berechnung der einfachen Schätzer  $Y(L)$  und  $Y(L - 1)$  für
'den Konvergenztest
if  $L > 2$ 
for  $l = L - 1 : L$ 
 $Y(l) = \text{mean}(X(l, 1 : N(l)));$ 
end
'Konvergenztest ohne Richardson Extrapolation
if  $\max((1/M) * \text{abs}(Y(L - 1)), \text{abs}(Y(L))) < (1/\text{sqrt}(2)) * (M - 1) * eps$ 
 $K = 2;$  'Abbruchsbedingung (siehe while-Schleife)
break
else
 $L = L + 1;$ 
end
else
 $L = L + 1;$ 
end
end
'Berechnung des Optionswerts mit den zuvor berechneten Samples
Wert=0;
for  $l = 1 : L$ 
 $m(l) = \text{mean}(X(l, 1 : N(l)));$ 
Wert=Wert+m(l);

```

```

end
'Berechnung der Rechenkosten
qq = 0;
for l = 2 : L
qq = qq + N(l) * (M(l-1) + M(l-2));
end
'Zur Anzeige der Kosten*epsilon2
Kosten=(N(1) + qq) * eps2
'Zur Anzeige der benötigten Samples zu jedem Level
N
'Zur Anzeige des Logarithmus zur Basis 4 vom Erwartungswert der Samples
Erwartungswert=log4(m)
'Zur Anzeige des Logarithmus zur Basis 4 von der Varianz der Samples
Varianz=log4(V);
'Bemerkung: log4 ist eine weitere Unterfunktion, die den Logarithmus zur
'Basis 4 berechnet.

```

'Unterprogramm mlmceulerauswertung2

```

'Berechnet den Optionswert zum ersten Level.
function y = mlmceulerauswertung2(S0, K, T, mu, sigma)
r = mu;
G1 = 0;
G1=sqrt(T).*randn; 'Generierung von Zufallszahlen
q1 = S0;
'Der Aktienkurs wird mit dem Euler-Maruyama Verfahren diskretisiert.
q1 = q1 + mu * T * q1 + sigma * G1 * q1;
y(1)=exp(-r * T)*max(0, q1 - K);

```

'Unterprogramm mlmceulerauswertung

```

function y = mlmceulerauswertung(S0, K, T, M, l, mu, sigma)
r = mu;
h1 = M-l+1 * T; 'Schrittweite zum Level L
h2 = M-l+2 * T; 'Schrittweite zum Level L-1
n1 = T/h1;

```

```

n2 = T/h2;
G1 = 0;
G2 = 0;
G1=sqrt(h1).*randn(1,n1); 'Generierung von Zufallszahlen
'Aufsummieren von M Brownschen Inkremente des Levels L
for k = 1 : Ml-2
G2(1, k) = 0;
for i = 1 : M
G2(1, k) = G1(1, i + (k - 1) * M) + G2(1, k);
end
end
'Berechnung der Optionswerte zu den Level l und l - 1
q1 = S0;
q2 = S0;
for k = 1 : n1
q1 = q1 + mu * h1 * q1 + sigma * G1(1, k) * q1;
end
y(1)=exp(-r * T)*max(0, q1 - K);
for k = 1 : n2
q2 = q2 + mu * h2 * q2 + sigma * G2(1, k) * q2;
end
y(2)=exp(-r * T)*max(0, q2 - K);

```


Literaturverzeichnis

- [1] BALLY, V.; TALAY, D.: *The law of the Euler scheme for stochastic differential equations, I: convergence rate of the distribution function*, Probability Theory and Related Fields, 104(1):43-60, 1995.
- [2] CAMBANIS, STAMATIS; HU, YAOZHONG: *Exact Convergence Rate of the Euler-Maruyama Scheme, with Application to Sampling Design*, Stochastics, An International Journal of Probability and Stochastic Processes, Volume 59, Issue 3 and 4, pp. 211-240, 1996.
- [3] GERSTNER, THOMAS: *Monte Carlo Methoden*, Vorlesungsskript, Frankfurt, 2010.
- [4] GERSTNER, THOMAS: *Finanznumerik (Computational Finance)*, Vorlesungsskript, Frankfurt, 2010.
- [5] GILES, MICHAEL B.: *Multilevel Monte Carlo path Simulation*, operations Research, Vol. 56, No. 3, pp. 607-617, 2008.
- [6] GILES, MICHAEL B.: *Improved multilevel Monte Carlo convergence using the Milstein scheme*, In: Monte Carlo and Quasi-Monte Carlo Methods 2006, Springer, 2007
- [7] GILES, MICHAEL B.: *Multilevel Monte Carlo for basket Options*, Winter Simulation Conference, 2009.
- [8] GILES, MICHAEL B.; WATERHOUSE, B.J.: *Multilevel quasi-Monte Carlo path simulation*, Radon Series on Computational and Applied Mathematics, Springer, 2009.
- [9] GLASSERMANN, PAUL: *Monte Carlo Methods in Financial Engineering*, Springer, New York, 2004.
- [10] GÜNTHER MICHAEL; JÜNGEL ANSGAR: *Finanzderivate mit MATLAB*, Mathematische Modellierung und numerische Simulation, Vieweg, 1.Auflage, Wiesbaden, 2003.
- [11] HEINZ, STEFAN W.: *Multilevel Monte Carlo Methoden und deren Anwendung*, Frankfurt, 2009.

- [12] HESTON, STEVEN L.: *A Closed-Form Solution for Options with Stochastic Volatility with Applications to Bond and Currency Options*, Review of Financial Studies, 6:327-343, 1993.
- [13] HULL, JOHN C.: *Optionen, Futures und andere Derivate*, Pearson Studium, 7.te aktualisierte Auflage, 2009.
- [14] KAHL, CHRISTIAN; JÄCKEL PETER: *Fast strong approximation Monte-Carlo schemes for stochastic volatility models*, Working Paper, ABN AMRO, 2006.
- [15] KLOEDEN, PETER E.; PLATEN, ECKHARD: *Numerical Solution of Stochastic Differential Equations*, Springer, 2nd ed., Berlin, 1995.
- [16] KLOEDEN, PETER E.: *Skript zur Vorlesung: Numerik stochastischer Differentialgleichungen*, Vorlesungsskript, Frankfurt, 2010.
- [17] WETH, TOBIAS: *Vorlesung Analysis*, Vorlesungsskript, Frankfurt, 2009.