

Projekt 8: Black-Scholes Modell

Clemens Ibrom

Prüfungsleistung

Computergestütztes Wissenschaftliches Rechnen, SoSe 24

Prof. Dr. S.R. Manmana

Georg-August Universität Göttingen

6.08.2024

Projektbetreuung: Leon Plickat
Projektauthor: Dr. Raju Lunkad

Inhaltsverzeichnis

1	Einleitung	1
2	Theorie	1
2.1	Die Black-Scholes Gleichung	1
2.2	Aktienkursverlauf	2
3	Methodik	2
3.1	Monte-Carlo	2
3.1.1	Fehler einer M-C Simulation	3
3.2	Zufallszahlen	3
3.2.1	Polarmethode	4
3.3	Call-Option	4
3.4	Histogramme	5
4	Implementation	5
4.1	Simulation	5
4.2	Simulationsparameter	5
4.3	Die C-Datei	6
5	Ergebnisse und Diskussion	6
5.1	Geometric Brownian Motion (GBM)	6
5.2	Europäische Call-Option	7
5.2.1	Die Dichteverteilung der Aktienwerte zum Verfallstermin	7
5.3	Varieerender Startpreis	8
5.4	Fehlerquellenanalyse	9
6	Zusammenfassung	9
7	Anhang	11

1 Einleitung

In diesem Projekt wird das Black-Scholes Modell (B-S) für Finanzderivatenmärkte numerisch behandelt. Im Fokus stehen sogenannte Optionen. Diese sind im Prinzip Wetten, darauf dass eine Aktie (oder ein Ensemble davon) zum Ablauf der Laufzeit im Wert steigen, oder fallen wird, im Vergleich zu einem festgelegten Preis (dem Ausübungspreis). Es gibt die Call-Option: eine Wette darauf, dass die Aktie steigt, und die Put-Option: eine Wette darauf, dass die Aktie fällt. Die Investierende Person hat dann das Recht die Optionen zu kaufen (Call) oder verkaufen (Put) zu dem Ausübungspreis. Ist die Aktie im Wert gestiegen, kann mit einer Call-Option Profit gemacht werden, indem sie zu einem billigeren Preis gekauft wird und sofort zu dem aktuellen Preis wieder verkauft wird. Analog kann mit Put-Optionen Gewinn gemacht werden[BS94].

Hier beschäftigen wir uns nur mit Call-Optionen, genauer mit europäischen Call-Optionen. Diese sind Call-Optionen wie oben beschrieben mit der Einschränkung, dass die Option nur zum festgelegtem Zeitpunkt eingelöst werden kann, d.h. nur zum Ende der Laufzeit gekauft werden kann.

Das Black-Scholes Modell ist ein mathematisches Modell das die Dynamik eines Finanzderivatenmarktes beschreibt. Insbesondere liefert B-S eine Formel für den Preis einer europäischen Call-Option in Abhängigkeit von dem Anfangswert der Aktie. Im folgenden wird diese analytische Formel mit Monte-Carlo Simulationen (M-C) numerisch genähert. Es wird untersucht wie gut M-C den Preis hervor sagt, was die Wahrscheinlichkeitsdichteverteilung der M-C Ergebnisse ist und ob Variation des Anfangspreises einen Einfluss hat.

2 Theorie

2.1 Die Black-Scholes Gleichung

Die Finanztheoretischen Annahmen über den Derivatenmarkt auf denen B-S beruht werden hier nicht Diskutiert. Die für das Projekt relevanten Annahmen sind, dass der Preis des Aktienwertes einer geeigneten geometrischen Brownschen Bewegung (GBM)[Ros14, Ch.10] folgt, es gibt einen risikofreien Zinssatz der Zeitlich konstant ist. Mehr dazu in den kommenden Abschnitt. Die B-S partielle Differentialgleichung ist:

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

wo $V = V(S, t)$ der Preis eines Derivates ist, $S = S(t)$ der Aktienpreis, und die Parameter r, σ für jeweils den risikofreien Zinssatz und die Volatilität stehen. Der Preis einer europäischen Call-Option der sich daraus ableiten lässt ist[BS73]:

$$C(S_0, T) = S_0 N(d_1) - K e^{-r(T-t)} N(d_2). \quad (2)$$

Hier sind:

$$d_1 = \frac{1}{\sigma\sqrt{T-t_0}} \left(\log\left(\frac{S_0}{K}\right) + \left(r + \frac{\sigma^2}{2}\right)(T-t_0) \right) \quad (3)$$

$$d_2 = d_1 - \sigma\sqrt{T-t_0}, \quad (4)$$

und t_0 der Startzeitpunkt, T die Laufzeit der Option, S_0 der Startwert der Aktie, K der Ausübungspreis und r, σ wie zuvor. Schließlich ist $N(d)$ die Verteilungsfunktion der Standard-Normalverteilung, gegeben durch:

$$N(d) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^d \exp\left(-\frac{x^2}{2}\right) dx. \quad (5)$$

2.2 Aktienkursverlauf

Der Verlauf eines Aktienkurses wird im B-S Modell mit Hilfe einer geometrischen Brownschen Bewegung simuliert[Le 16, Ch. 8.4.2]. Dies ist ein stochastischer Prozess der von der folgenden Gleichung beschrieben wird:

$$S_t = S_{t-1} \exp\left(\left(r - \frac{\sigma^2}{2}\right)(\Delta t) + \sigma\sqrt{(\Delta t)}Z\right). \quad (6)$$

Dabei sind die Parameter r, σ wie zuvor, Δt der Zeitschritt und Z ist eine standardnormalverteilte Zufallszahl. Aus letzterem zusammen mit Gleichung 6 folgt, dass die Ergebnisse logarithmisch normalverteilt sind.

3 Methodik

3.1 Monte-Carlo

Die numerische Berechnung des Erwartungswertes von dem Preis der Call-Option erfolgt mit Monte-Carlo Simulationen. Dafür wird folgendes Integral approximiert[Ber]:

$$\mathbb{E}(C_T(S_T))e^{-r(T-t_0)} = \int_0^1 C_T(f(x, S_T))dx. \quad (7)$$

Hier sind T, t_0 wie zuvor und C_T ist der Zahlungsbetrag (auch payoff). Die Funktion f ist eine Verteilungsfunktion. Der Payoff ist als

$$C_T = \max\{S_T - K, 0\}e^{-rT} \quad (8)$$

gegeben. Das Integral 7 wird approximiert indem der Integrand an vielen Standardnormalverteilten Zufallszahlen ausgewertet wird und dann die Ergebnisse gemittelt werden. Nach dem Gesetz der Großen Zahlen ist dies eine gute Approximation des gesuchten Wertes[Man24, Ch. A.3].

3.1.1 Fehler einer M-C Simulation

Bei einer Monte-Carlo Integration mit x_1, \dots, x_N Stützstellen ist der Fehler gegeben von:

$$\int f dV \simeq V \langle f \rangle \pm V \sqrt{\frac{\langle f^2 \rangle - \langle f \rangle^2}{N}}. \quad (9)$$

Hier ist $\langle f \rangle := \frac{1}{N} \sum_{i=1}^N f(x_i)$ [Pre+92]. In unserer Simulation ist $V = 1$, s. Gl 7. Da die Simulation einen empirischen Mittelwert von Zufallszahlen als Ergebniss hat, da die Funktion f hier aus Gl. 8 kommt und die Werte von S_T bereits mittels Simulation erzeugt wurden (und zufällig einer Dichteverteilung folgen 2.2) halten wir es für angebrachter den Fehler mittels des Standard Fehlers¹

$$\bar{\sigma} = \sqrt{\frac{1}{N(N-1)} \sum_{i=1}^N (X_i - \bar{X})^2} \quad (10)$$

anzugeben [Ber], [Man24, Ch. A.3]. Hierbei sind X_i die Ergebnisse der Simulationsschritte und \bar{X} das arithmetische Mittel über alle Simulationsschritte.

3.2 Zufallszahlen

Die für Gl. 6 benötigten standardnormalverteilten Zufallszahlen wurden mit der Polarmethode (3.2.1) und dem Zufallszahlengenerator MT-19937 aus der GSL-RNG bibliothek. Der Generator `gsl_rng_mt19937` ist ein Mersenne Twister generator und hat eine Periode von $2^{19937} - 1 \sim 10^{6000}$. Im Verlauf der Lösung werden folgende Anzahlen an Zufallszahlen generiert:

- die Funktion `gbm()` erzeugt 1 Zufallszahl mit MT-19937
- die Funktion `mc_step()` ruft `gbm()` N -Mal auf, wobei N die Anzahl der Schritte von der M-C Integration ist, also N zufallszahlen
- die Funktion `mc_simulation()` ruft `mc_step()` N -Mal auf.

Daher werden pro ausgeführter Simulation mit N Schritten genau N^2 mit dem Generator erzeugt. Insgesamt werden also pro Aufgabenteil:

- | | |
|-------------------|--|
| 1. Teil 1: 10^2 | 2. Teil 2: 10^8 |
| 3. Teil 3: 0 | 4. Teil 4: $10^4 + 2.5 \cdot 10^5 + 10^6 + 10^8$ |

Zufallszahlen erzeugt, was insgesamt $201260100 \sim 10^8$ Zufallszahlen erzeugt. Somit wird die Periode nicht überschritten und wir können von einer Gleichverteilung ohne Korrelation ausgehen. Im Anhang 7 kann an Hand des Histogramms bestätigt werden, dass die Implementierung der Polarmethode in der Tat Normalverteilte Zufallszahlen mit Mittelwert 0 und Standardabweichung 1 liefert.

¹[Man24] Gl. A.7

3.2.1 Polarmethode

Die Polarmethode [MB64] erzeugt aus zwei unabhängigen, gleichverteilten Zufallszahlen zwei unabhängige standardnormalverteilte Zufallszahlen. Der Algorithmus ist:

1. erzeuge zwei unabh., gleichverteilte Zahlen $u, v \in [-1, 1]$
2. berechne $r = u^2 + v^2$. Falls $r = 0$ oder $r \geq 1$, gehe zurück zu 1.
3. berechne $m = \sqrt{\frac{-2 \cdot \log(r)}{r}}$
4. die unabhängigen standardnormalverteilten Zahlen sind $x = u \cdot m$ und $y = v \cdot m$.

3.3 Call-Option

Die analytische Lösung für den Preis der Call-Option aus Gleichung 2 benötigt die Implementation der Verteilungsfunktion 5. In der Implementation wurde Folgende Beziehung ausgenutzt:

$$N(x) = \frac{1}{2} \left(1 + \operatorname{erf} \left(x / \sqrt{2} \right) \right). \quad (11)$$

wo $\operatorname{erf}(x) = 2/\sqrt{\pi} \int_0^x \exp(-t^2) dt$ die Gauß'sche Fehlerfunktion ist. Gleichung 11 lässt sich mit der Substitution $t = u \cdot \sqrt{2}$ herleiten:

$$\begin{aligned} \operatorname{erf}(x/\sqrt{2}) &= \frac{2}{\sqrt{\pi}} \int_0^{x/\sqrt{2}} e^{-t^2} dt = \frac{2}{\sqrt{2\pi}} \int_0^x e^{-u^2/2} du \\ &= 2 \left(\frac{1}{\sqrt{2\pi}} \int_{-\infty}^x e^{-u^2/2} du - \frac{1}{\sqrt{2\pi}} \int_{-\infty}^0 e^{-u^2/2} du \right) \\ &= 2 (N(x) - N(0)) = 2 \left(N(x) - \frac{1}{2} \right). \end{aligned} \quad (12)$$

Bei der Letzten Gleichung wurde genutzt, dass $N(d)$ die Wahrscheinlichkeit ist, dass eine standardnormalverteilte Zufallszahl kleiner gleich d ist, was für $d = 0$ gleich 0.5 ist.

Der Vorteil dessen ist, dass bereits eine eigene Implementation der Gaußschen Fehlerfunktion besteht. Diese benutzt die Simpson Integrationsregel [Atk78]. Hierbei wird die Domäne der Integration in gleichgroße Intervalle aufgeteilt auf denen der Integrand mit einer Parabel approximiert wird (Gl 5.1.13 in [Atk78]):

$$\int f \simeq \frac{h}{3} \left(f(a) + f \left(\frac{a+b}{2} \right) + f(b) \right), \quad h = \frac{b-a}{2} \quad (13)$$

wo a, b für die Intervallgrenzen stehen. Der Fehler der Simpson Regel ist (Gl 5.1.15 in [Atk78]):

$$\text{Fehler} \propto h^5. \quad (14)$$

3.4 Histogramme

Um die Wahrscheinlichkeitsdichtefunktion der Ergebnisse einer M-C Simulation zu ermitteln wurden die Ergebnisse als ein relatives Histogramm geplottet. Hierfür wurden eigene Histogrammroutinen eingesetzt, die im Laufe des Semesters auf Übungsblättern erarbeitet wurden.

4 Implementation

4.1 Simulation

Es werde die externen Bibliotheken *stdio.h*, *stdlib.h*, *math.h* für standard mathematische Funktionen wie die Wurzel, *gsl/gsl_rng.h* für den Zufallszahlen Generator (s. 3.2) und *time.h* für sowohl den Seed für den RNG als auch Laufzeitvermessung.

Die Funktionen die für eine M-C Simulation verwendet werden sind:

- *gbm*: berechnet einen Zeitschritt der geometrischen Brownschen Bewegung unter Benutzung von einer Normalverteilten Zufallszahl. Da die Polarmethode zwei solche Zahlen generiert, wird eine verworfen
- *mc_step*: ein Schritt der M-C Simulation. Berechnet den Aktienverlauf über die Dauer der Laufzeit und gibt den Wert der Aktie zum Verfallszeitpunkt zurück
- *mc_simulation*: führt N M-C Schritte durch. In jedem Schritt wird der von *mc_step* ausgegebene Wert mittels $\max\{S_T - K, 0\}e^{-rT}$ auf den Barwert umgerechnet. Zum Schluss wird über diese Barwerte gemittelt um das Ergebnis zu erhalten. Diese Funktion ist das Kernstück von Aufgabenteil 4. In Aufgabenteil 2 wird genau das gleiche umgesetzt, aber da die Werte von S_T für Aufgabenteil 3 benötigt werden wurde nicht die Funktion aufgerufen, sondern eine leicht angepasste Version verwendet.

Die Laufzeit der Simulationen in Aufgabenteil 4 ist erheblich länger als in den anderen Teilen, da hier 120 Simulationen mit 10 000 Schritten ausgeführt werden. Im Durchschnitt dauert hier eine Simulation ungefähr 4.87 Sekunden², was insgesamt zu einer Dauer von ungefähr 9.74 Minuten führt.

4.2 Simulationsparameter

Die gegebenen Parameter sind:

- | | |
|--|-------------------------------------|
| 1. Aktienkurs zu Beginn $S_0 = 70\text{€}$ | 2. Ausübungspreis $K = 100\text{€}$ |
| 3. Laufzeit $T = 1$ Jahr | 4. Risikofreier Zinssatz $r = 12\%$ |
| 5. Volatilität $\sigma = 10\%$ | |

²auf einem PC im CIP-Pool der Physik

4.3 Die C-Datei

In der Datei *projekt8.c* befinden sich alle für dieses Projekt geschriebenen Funktionen. Manche der Methoden benutzen bestehende Algorithmen und Funktionalitäten aus der *MyNumerics* Bibliothek, sowie der oben erwähnten externen Bibliotheken. In der *main* Funktion werden die Aufgabenteile nacheinander, mit der Ausnahme von Teil 3, gelöst (s. 4.1). Aufgabenteil 3 wird innerhalb von Aufgabenteil 2 gelöst, da sich die Aufgabe mit den dort generierten Daten beschäftigt, bevor diese für Teil 2 bearbeitet werden. Beim ausführen werden die Preise mit Fehler ausgegeben, sowie Fortschrittmarker.

In Aufgabenteil 4 werden zusätzlich die Laufzeiten der Simulation für jeden Startwert berechnet. Dies geschieht mittel der Messung der CPU-Laufzeit über die Funktionalitäten von *time.h*.

5 Ergebnisse und Diskussion

5.1 Geometric Brownian Motion (GBM)

Der Aktienkurs, Simuliert gemäß 6 über die Dauer von einem Jahr in Schritten von ~ 3.65 Tagen (1% der Gesamtdauer) sieht beispielsweise so aus:

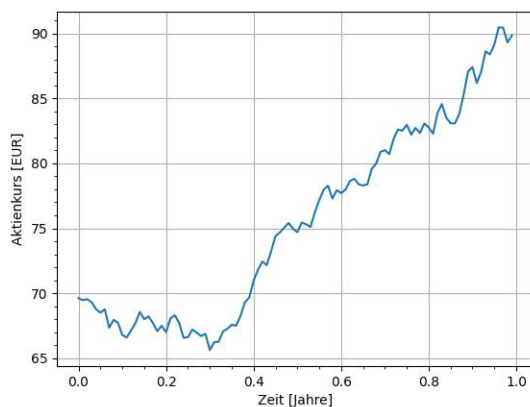


Abbildung 1: Ein möglicher Aktienkursverlauf in EUR über 1 Jahr.

Als Startwert wird der in der Aufgabenstellung gegebene Parameter verwendet, mit dem Wert $S_0 = 70\text{€}$. Da es sich um einen zufälligen Verlauf handelt, wird jede neue Simulation einen anderen Verlauf liefern. Was jedoch unverändert bleibt ist, dass die Werte immer positiv bleiben, was der Realität entspricht.

Ein Nachteil dieses Modells ist, dass zufällige äußere Ereignisse, die den Aktien Wert stark beeinflussen, nicht beachtet werden. Ein Aktienkurs kann unstetige Veränderungen (Sprünge) aufweisen, aber unser Modell setzt einen Stetig

Verlauf vor, da die Exponentialfunktion stetig ist. Aus diesem Grund wird der Simulierte Verlauf mit *plt.plot* dargestellt.

5.2 Europäische Call-Option

Der Preis einer europäischen Call-Option ist gegeben von Gl. 2. Für den Startwert $S_0 = 70\text{€}$ wurde der Preis der Call-Option mittels M-C mit $N = 10000$ Schritten Simuliert mit folgendem Ergebnis:

Monte-Carlo Callprice ist 0.0220002806833178 mit Fehler 0.0034733644873890
Analytischer Callprice ist 0.0232381239579704

Die Abweichung des Simulierten Preises ist also 0.0012378432746526016 ³. Das ist eine Größenordnung kleiner als die Ergebnisse und hat gleiche Größenordnung wie der Fehler von M-C, berechnet mit 10. Daher ist M-C eine gute numerische Näherung an den Wert. Im Anhang 5 befindet sich ein Plot von 2 in Abhängigkeit von S_0 .

5.2.1 Die Dichteverteilung der Aktienwerte zum Verfallstermin

Wie in 2.2 diskutiert erwarten wir, dass die Preise der Aktie zum Verfallstermin eine logarithmische Normalverteilung haben. Um das zu überprüfen wurden die für die Simulation in Aufgabenteil 2 10 000 Werte von S_T in einem relatives Histogramm geplottet:

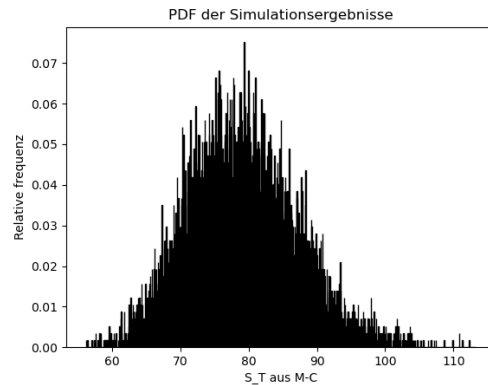


Abbildung 2: Dichteverteilung der Ergebnisse der M-C Simulation

Man erkennt, dass die Verteilung skewed nach Links ist und dass der tail rechts länger ist, was darauf deutet, dass es eine logarithmische Normalverteilung ist.

³Dieser Wert wird bei jeder Simulation unterschiedlich sein, die Größenordnung bleibt jedoch erhalten

Das entspricht unseren Erwartungen. Dies Validiert die Ergebnisse aus Aufgabenteil 2 , da die analytische Lösung laut B-S davon ausgeht, dass der Aktienpreis einer GBM6 folgt, also logarithmisch Normalverteilte Werte liefert. Somit ist es Sinnvoll M-C einzusetzen und mit der analytischen Lösung zu Vergleichen.

5.3 Varieerender Startpreis

Nun soll überprüft werden, wie genau die M-C Simulationsergebnisse an den Analytischen bleiben bei verschiedener Anzahl von Schritten und Startpreisen. Alle anderen Parameter werden hierbei fest gelassen wie in 4.2 angegeben. Hierfür wird der Startpreis S_0 systematisch um 0.50€ erhöht von 70€ bis 130€. Es wurden vier Simulationen durchgeführt mit jeweil $N = 100, 500, 1000$ und 10000 Schritten. Die Ergebnisse wurden zusammen mit der Analytischen Lösung und Fehlerbalken berechnet nach 10 geplottet:

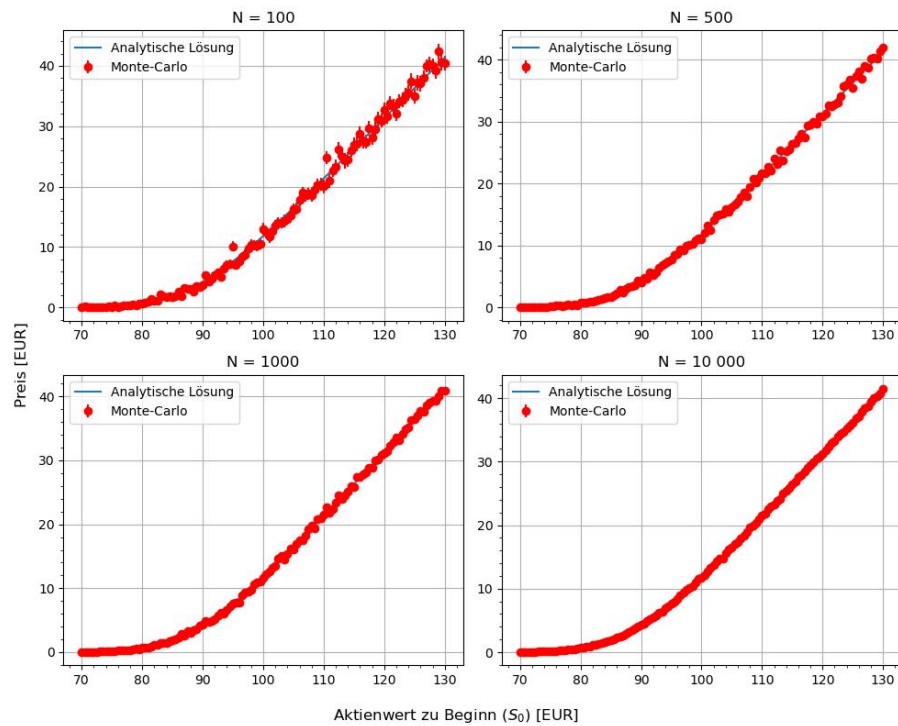


Abbildung 3: Vergleich von M-C und analytischer Lösung mit 100,500,1000 und 10000 Schritten bei variierendem Startpreis der Aktie

Die M-C Ergebnisse sind also auch bei verschiedenen Startwerten S_0 nah an der analytischen Lösung. Mit wachsender Anzahl von Schritte werden die Si-

mulierten Ergebnisse besser, sowohl im Sinne des Fehlers als auch der Nähe an der Analytischen Lösung. Bemerkenswert ist, dass unabhängig von Anzahl der Schritte ist die Simulierte Lösung immer sehr nah an der analytischen im Bereich $S_0 \in [70, 80]$. Dies liegt daran, dass der Payoff in der Simulation mit Gl. 8 das Maximum von 0 und dem Wert $(S_T - K) \cdot e^{-rT}$ und wie man bei 2 sieht, sind für $S_0 = 70\text{€}$ fast alle Werte von S_T fallen unter den Ausübungspreis K aus. Desweiteren werden mit mehr Simulationsschritten die Fehlerbalken kleiner. Im Anhang befinden sich die einzelnen Plots separiert.

5.4 Fehlerquellenanalyse

Der größte Beitrag zu dem Fehler der Simulationsergebnisse ist der Fehler der von M-C stammt. Wie in 3.1.1 diskutiert ist der Fehler von Größenordnung $N^{-1/2}$. Mit mehr Simulationsschritten wird er also kleiner, aber es bestehen Verbesserungs Möglichkeiten. Eine davon ist das sogenannte *stratified sampling*, was, gegeben die zu integrierende Funktion ist 'gleichmäßig' genug, einen Fehler von Ordnung N^{-1} [Pre+92, p.319] ergeben kann. Der Ansatz ist, die Integrationsdomäne aufzuteilen, wenn der Integrand in allen bis auf einen dieser Teile sehr klein ist, liefert eine naive Anwendung von der Strategie keine Informationen. Daher muss gegebenenfalls diese Strategie mit *importance sampling* [Pre+92, Ch.7.8] kombiniert werden.

Des weiteren können Rundungsfehler einen Einfluss auf das Ergebnis haben, besonders wenn in Gl. 8 der Wert $S_T - K$ sehr klein aber positiv ist. Dies trägt aber kaum zu Verfälschung der Ergebnisse bei, da *double* Präzision ausreichend genau ist für unsere Anwendung. Hier kann, um höhere Genauigkeit zu erzielen, *long double* Präzision verwendet werden, wenn der Rechenaufwand keine Rolle spielt.

Bei der Analytischen Lösung wird mit der Simpson Regel integriert, diese hat eine Fehler der Abhängig von der Schrittweite ist 3.3,14. Es wurde in der Implementation $h = 10^{-4}$ gewählt, also ist der Fehler von der Ordnung 10^{-20} . Für genauere Ergebnisse besteht hier die Möglichkeit einer besseren Integrationsmethode, z.B. Bode's Regel [Pre+92, Ch.4].

Zuletzt, basieren große Teile der Theorie darauf, dass passende⁴ Zufallszahlen generiert werden. In der Implementierung wird der *Mersenne Twister* MT-199373.2 verwendet. Dieser generiert **pseudo** Zufallszahlen, da es nicht möglich ist wahre Zufallszahlen programmatisch zu erzeugen. Der gewählte Generator erstellt stark gleichverteilte Zahlen [MN98] und ist deswegen eine gute und weit verbreitete Wahl für Zufallszahlen Erstellung, auch da er bereits in der *gsl/gsl_rng* Bibliothek implementiert ist.

6 Zusammenfassung

Die Ergebnisse der Monte-Carlo Simulation sind positiv ausgefallen. Das gewählte numerische Verfahren approximiert die von dem Black-Scholes Modell getroffene

⁴von einer bestimmten Verteilung

nen Prognosen sehr gut. Zudem wurde auch gezeigt, dass M-C Simulationen auch die erwartete Wahrscheinlichkeitsverteilung der Ergebnisse liefert. Schon mit relativ wenig Rechenaufwand sind die Simulierten Ergebnisse gut und für 10 000 Schritte kaum von den B-S Werten zu unterscheiden. Besteht das Bedürfnis für genauere simulierte Werte so kann immer die Anzahl an Schritten erhöht werden. Damit wächst aber auch der Rechenaufwand. Wenn dies vermieden werden soll, dann müssen verbesserte M-C verfahren, wie z.B. stratified sampling [Pre+92, Ch.7.8] verwendet werden.

In diesem Projekt wurde der Preis einer europäischen Call-Option mit Monte-Carlo Simulationen untersucht. Es ist naheliegend zu vermuten, dass für die europäische Put-Option ähnliche Ergebnisse erreicht werden können. Das B-S Modell kann auf ein Ensemble von n Aktien erweitert werden und es wäre auch Interessant in diesem Fall zu untersuchen wie gut M-C Simulationen die analytische Lösung nähern.

Das B-S Modell kann auch auf andere Arten von Optionen, z.B. amerikanische, erweitert werden. Dort könnte man auch untersuchen ob M-C Simulationen auch da eine gute Näherung liefern.

Schließlich ist das B-S Modell in der hier behandelten Form nicht realitätsgetreu durch die getroffenen Annahmen[Jan18]. Es existieren erweiterte Modelle die realistischer als B-S sind und es wäre Interessant zu prüfen, ob auch diese mit M-C Simualtionen genähert werden können.

7 Anhang

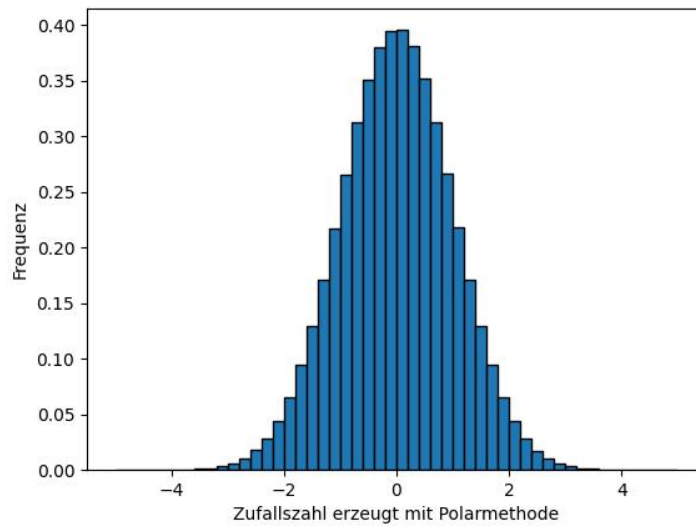


Abbildung 4: 10^7 Zufallszahlen mit der Polarmethode generiert und als Histogramm dargestellt, es hat die erwartete Form einer Gauß'schen Glocke, also sind die Zahlen Normalverteilt

Der Mittelwert und die Standardabweichung des Histogrammes, also der dazugehörigen Dichteverteilungsfunktion sind:

Erwartungswert von dem Histogramm ist -0.0000887000532200
Standardabweichung von dem Histogramm ist 1.0020775725760329.

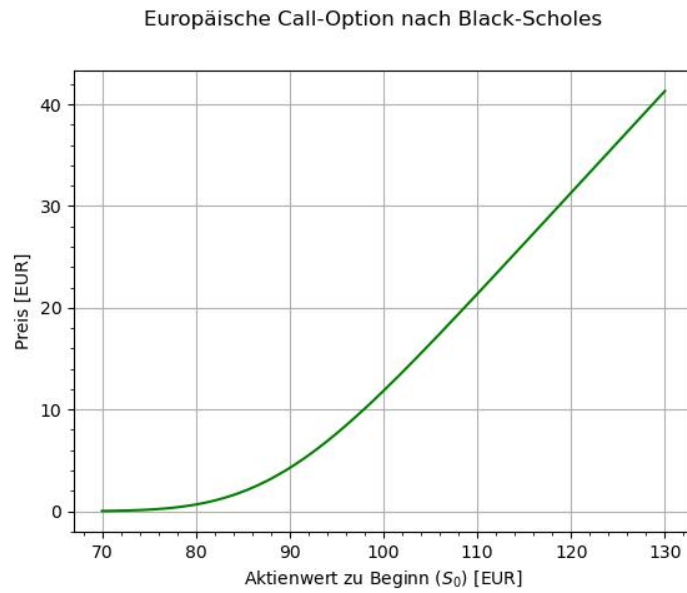


Abbildung 5: Preis einer europäischen Call-Option nach B-S 2, geplottet gegen den Aktienwert zu Beginn S_0 in Euro

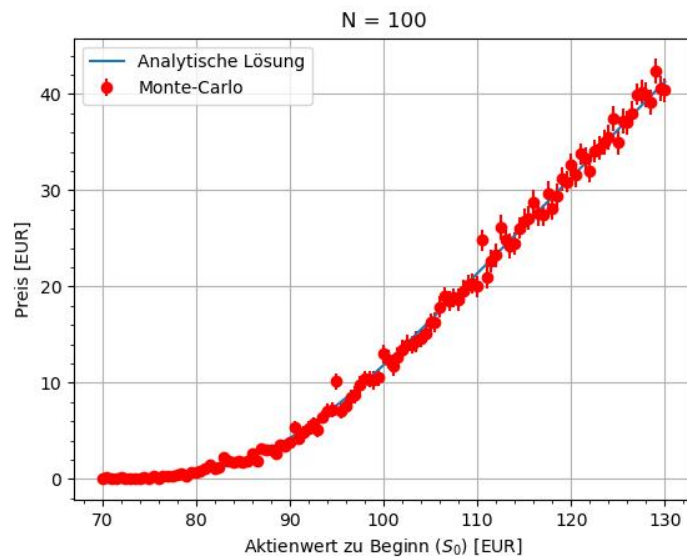


Abbildung 6: Preis einer europäischen Call-Option mit 100 M-C Schritten und Startwert $S_0 = 70\text{€}$

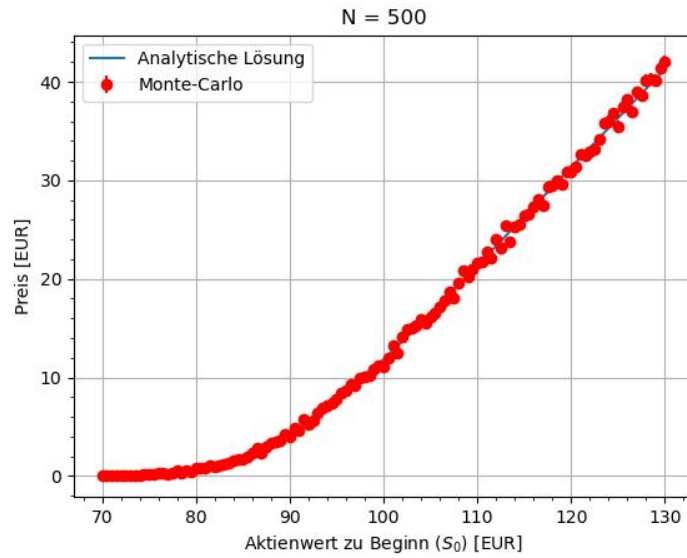


Abbildung 7: Preis einer europäischen Call-Option mit 500 M-C Schritten und Startwert $S_0 = 70\text{€}$

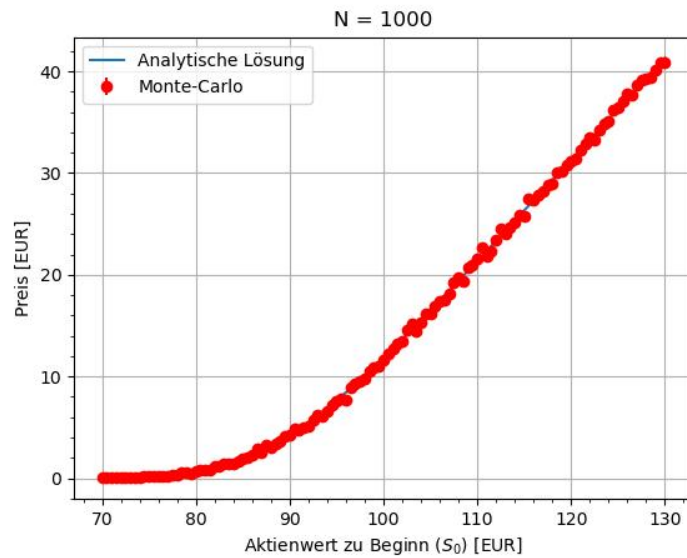


Abbildung 8: Preis einer europäischen Call-Option mit 1000 M-C Schritten und Startwert $S_0 = 70\text{€}$

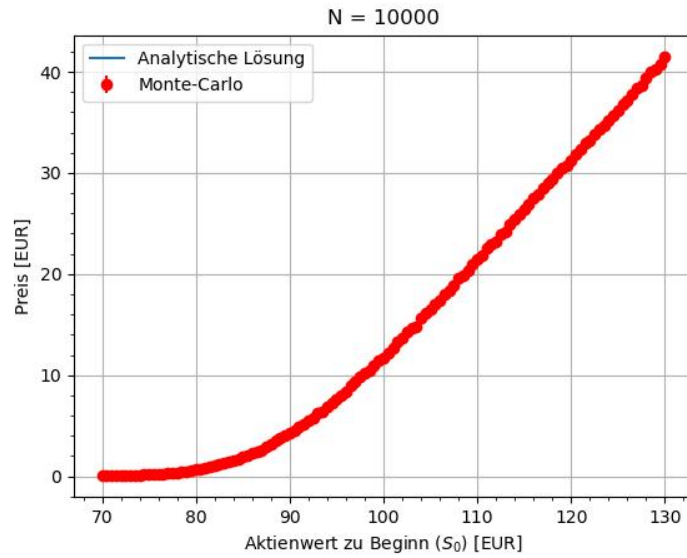


Abbildung 9: Preis einer europäischen Call-Option mit 10 000 M-C Schritten und Startwert $S_0 = 70\text{€}$

Literatur

- [MB64] G. Marsaglia und T. A. Bray. “A Convenient Method for Generating Normal Variables”. In: *SIAM Review* 6.3 (1964), S. 260–264. ISSN: 00361445. URL: <http://www.jstor.org/stable/2027592> (besucht am 02.08.2024).
- [GT71] John Gurland und Ram C. Tripathi. “A Simple Approximation for Unbiased Estimation of the Standard Deviation”. In: *The American Statistician* 25.4 (1971), S. 30–32. ISSN: 00031305. URL: <http://www.jstor.org/stable/2682923> (besucht am 02.08.2024).
- [BS73] Fischer Black und Myron Scholes. “The Pricing of Options and Corporate Liabilities”. In: *Journal of Political Economy* 81.3 (Mai 1973), S. 637–654. DOI: 10.1086/260062.
- [Atk78] Kendall E. Atkinson. *An introduction to numerical analysis*. John Wiley Sons, 1978. ISBN: 0471029858.
- [Pre+92] William H. Press u. a. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge: Cambridge University Press, 1992. ISBN: OCLC:870911424.
- [BS94] Jean-Philippe Bouchaud und Didier Sornette. “The Black-Scholes option pricing problem in mathematical finance: Generalization and extensions for a large class of stochastic processes”. In: *Journal de Physique I* 4.6 (Juni 1994), S. 863–881. DOI: 10.1051/jp1:1994233.
- [MN98] Makoto Matsumoto und Takuji Nishimura. “Mersenne twister”. In: *ACM Transactions on Modeling and Computer Simulation* 8.1 (Jan. 1998), S. 3–30. DOI: 10.1145/272991.272995.
- [Ros14] Sheldon M. Ross. *Introduction to Probability Models*. 11. Aufl. Academic Press, 2014. ISBN: 9780128143476.
- [Le 16] Jean-François Le Gall. *Brownian Motion, Martingales, and Stochastic Calculus*. Springer, Apr. 2016. ISBN: 9783319310893.
- [Jan18] Zuzana Janková. “Drawbacks and Limitations of Black-Scholes Model for Options Pricing”. In: *Journal of Financial Studies and Research* 2018.(2018) (Aug. 2018), S. 1–7. DOI: 10.5171/2018.79814.
- [Man24] Prof. Dr. S.R. Manmana. *Skriptum zur Vorlesung Computergestütztes Wissenschaftliches Rechnen, Sommersemester 2024*. [Accessed 02-08-2024]. 2024.
- [Ber] Humboldt Universität zu Berlin. *16.1 Simulation techniques for option pricing*. http://sfb649.wiwi.hu-berlin.de/fedc_homepage/xplore/tutorials/xfghtmlnode111.html. [Accessed 02-08-2024].