

Strukturerhaltende Approximationen von Wurzel-Diffusionsgleichungen

Bachelorarbeit

20. September 2011

vorgelegt von

Kirsten Bernhard
Geb. am: 7. August 1987
in: Frankfurt am Main
Matrikelnummer: 3608409
Studienrichtung: Mathematik

eingereicht bei

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

Frankfurt am Main, den 20. September 2011

Inhaltsverzeichnis

| | |
|--|-----------|
| Abbildungsverzeichnis | 3 |
| Ehrenwörtliche Erklärung | 4 |
| 1. Einleitung | 5 |
| 2. Stochastische Prozesse | 6 |
| 2.1. Wiener Prozess | 6 |
| 2.2. Lemma von Ito | 8 |
| 2.3. Spezielle stochastische Prozesse | 8 |
| 3. Diskretisierungsverfahren | 12 |
| 3.1. Existenz- und Eindeigkeitssatz | 12 |
| 3.2. Euler-Verfahren | 13 |
| 3.3. Milstein-Verfahren | 14 |
| 3.4. Verbesserung zum Euler-Verfahren | 14 |
| 3.5. Konvergenzordnung | 15 |
| 3.6. Methoden 2. Ordnung | 19 |
| 3.7. Simulation stochastischer Multi-Integrale | 20 |
| 3.8. Vektorwertige stochastische Differentialgleichungen | 21 |
| 3.9. Kommutativitätsbedingung | 22 |
| 4. Wurzel-Diffusionsprozess | 23 |
| 4.1. Definition [CIR-Modell] | 24 |
| 4.2. Übergangsdichte eines stochastischen Prozesses | 24 |
| 4.3. Parameterbeschreibung | 24 |
| 4.4. Proposition | 25 |
| 4.5. Chi-Quadratverteilung / nichtzentrale Chi-Quadratverteilung | 25 |
| 4.6. Simulation eines Pfades des CIR-Modells | 27 |
| 4.7. Diskretisierungsverfahren für das CIR-Modell | 28 |
| 4.7.1. Positivität | 29 |
| 4.7.2. Euler-Maruyama-Verfahren | 29 |
| 4.7.3. Milstein-Verfahren | 31 |
| 4.7.4. Satz [Drift-implizites Milstein-Verfahren & Positivität] | 31 |
| 4.7.5. Taylor Verfahren schwacher Ordnung 2 | 32 |
| 5. Numerische Ergebnisse | 33 |
| 5.1. Starke Konvergenz | 33 |
| 5.2. Schwache Konvergenz | 38 |
| 6. Schlussbetrachtung | 40 |
| A. Programmcodes | 42 |

Abbildungsverzeichnis

| | | |
|-----|--|----|
| 1. | Pfad eines Wiener Prozesses | 7 |
| 2. | Prozess mit konstanten Koeffizienten | 9 |
| 3. | Prozess mit geometrischer SDGL | 10 |
| 4. | Wurzel-Prozess | 11 |
| 5. | starke Konvergenzordnung | 18 |
| 6. | schwache Konvergenzordnung | 18 |
| 7. | Pfad des CIR-Modells für $d > 1$ | 28 |
| 8. | Pfad des CIR-Modells für $d \leq 1$ | 29 |
| 9. | Empirische Verteilung des CIR-Prozesses mittels der exakten Simulation und der Simulation durch das Euler-Verfahren nach Deelstra und Delbaen (Euler-DD) mit $\kappa = 0.3$, $\theta = 0.04$, $\sigma = 0.2$, $X_0 = 0.04$ und $T = 0.25$ | 35 |
| 10. | Fehler $\epsilon(\Delta)$ als eine Funktion mit $\kappa = 1$, $\theta = 1$, $X_0 = 0.01$ und $\sigma = 1$. . . | 35 |
| 11. | Fehler $\epsilon(\Delta)$ als eine Funktion mit $\kappa = 1$, $\theta = 1$, $X_0 = 0.01$ und $\sigma = \sqrt{3}$. . | 36 |
| 12. | Regressionsgeraden für die starke Konvergenz mit $\kappa = 1$, $\theta = 1$, $X_0 = 0.01$ und $\sigma = 1$ | 36 |
| 13. | Regressionsgeraden für die starke Konvergenz mit $\kappa = 1$, $\theta = 1$, $X_0 = 0.01$ und $\sigma = \sqrt{3}$ | 37 |
| 14. | Regressionsgeraden für die schwache Konvergenz mit $\kappa = 1$, $\theta = 1$, $X_0 =$ 0.01 und $\sigma = 1$ für 500 Runden | 38 |
| 15. | Regressionsgeraden für die schwache Konvergenz mit $\kappa = 1$, $\theta = 1$, $X_0 =$ 0.01 und $\sigma = 1$ für 1000 Runden | 39 |

Ehrenwörtliche Erklärung

Hiermit versichere ich, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe, dass alle Stellen der Arbeit, die wörtlich oder sinngemäss aus anderen Quellen übernommen wurden, als solche kenntlich gemacht sind und dass die Arbeit in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegt wurde.

Frankfurt am Main, den 20. September 2011

1. Einleitung

Wurzel-Diffusionsprozesse finden heute oft Anwendung in der Finanzmathematik. Sie werden beispielsweise in dem Short-Rate-Modell von Cox, Ingersoll und Ross (kurz CIR-Modell) verwendet. Short-Rate bezeichnet im Zusammenhang mit Zinsderivaten den Zinssatz einer sicheren Anlage für unendlich kleinen Zeitraum. Die Short-Rate stellt einen stetigen Zinssatz dar. Folglich ist es nur ein theoretisches Modell, welches man auf dem Kapitalmarkt so nicht beobachten kann. Mit Hilfe der Short-Rate wird versucht zukünftige Entwicklungen der Zinsstruktur zu beschreiben. Auf dem Finanzmarkt sind auch andere Short-Rate-Modelle bekannt, wie beispielsweise das Modell von Vasicek (1977). Dieses Modell verwendet einen gaußschen Ornstein-Uhlenbeck-Prozess für die Entwicklung der Short-Rate. Damit ist die Short-Rate in diesem Modell normalverteilt, das heißt die Zinssätze können mit positiver Wahrscheinlichkeit negative Werte annehmen. Für eine detaillierte Beschreibung des Vasicek-Modells, sowie dem Ornstein-Uhlenbeck-Prozess wird auf ([5], pp. 108-120) verwiesen. Das CIR-Modell verbessert dieses Modell durch Verwendung eines Wurzel-Diffusionsprozesses, für den, durch strukturerhaltende Diskretisierungsverfahren, verhindert wird, dass die Zinssätze negative Werte annehmen können.

Was genau sind numerische Diskretisierungsverfahren?

Für viele stochastische Differentialgleichungen existieren keine geschlossenen Lösungsformeln. In diesen Fällen werden numerische Diskretisierungsverfahren eingesetzt. Mit der Hilfe dieser numerischen Verfahren sollen die Lösungen der stochastischen Differentialgleichungen möglichst genau approximiert werden. Stochastische Differentialgleichungen werden zum Beispiel für die Bewertung von Optionen auf dem Finanzmarkt eingesetzt.

Für die Anwendung von Diskretisierungsverfahren auf Wurzel-Diffusionsprozesse, wie beispielsweise das CIR-Modell, muss auf die Strukturerhaltung des Prozesses geachtet werden.

Aber was genau bedeutet das?

Um strukturerhaltende Diskretisierungsverfahren für Wurzel-Diffusionsprozesse herzuleiten muss darauf geachtet werden, dass das Verfahren positivitätserhaltend sind. Das hat den folgenden Grund:

Im Wurzel-Diffusionsprozess haben wir eine Wurzel im Diffusionsterm, was der Name bereits erkennen lässt. Wenn man nicht auf die Positivität des Verfahrens achtet, kann der Wert des Prozesses in einem Schritt negativ werden und das Verfahren würde im nächsten Schritt scheitern, da man die Wurzel aus einer negativen Zahl ziehen müsste.

Ziel dieser Bachelorarbeit ist es, die Modelle, die genau diese positivitätserhaltende Eigenschaft besitzen, vorzustellen und diese mit den Standard-Diskretisierungsverfahren, also die, die die Positivität gerade nicht gewährleisten, zu vergleichen.

Anfangs werden im Kapitel 2 die wichtigsten Grundlagen zu stochastischen Prozessen gegeben. Im Kapitel 3 wird auf die Standard-Diskretisierungsverfahren eingegangen. Es werden wichtige Begriffe, wie beispielsweise die Konvergenzordnung, eingeführt. Im Kapitel 4 beginnt der Hauptteil dieser Arbeit. Es wird auf die Wurzel-Diffusionsgleichung eingegangen und das CIR-Modell näher vorgestellt. Außerdem werden die positivitätserhaltenden Diskretisierungsverfahren für das CIR-Modell gegeben. Im letzten Kapitel 5 werden numerische Ergebnisse dieser Diskretisierungsverfahren aufgezeigt. Die verwendeten Programme dieser Bachelorarbeit wurden in Matlab implementiert. Sie befinden sich im Anhang.

2. Stochastische Prozesse

In diesem Kapitel wird in die Theorie stochastischer Prozesse eingeführt. Der wichtigste stochastische Prozess für diese Bachelorarbeit, der Wiener Prozess, wird vorgestellt. Außerdem beschäftigt sich dieses Kapitel mit dem Begriff des Ito-Prozesses, sowie dem Lemma von Ito. Die Herangehensweise dieses Kapitels orientiert sich an [3], sowie [10].

Definition [Stochastischer Prozess]

Unter einem stochastischer Prozess versteht man eine Zufallsgröße $X(t)$, deren Wert über die Zeit t zufälligen Änderungen ausgesetzt ist. Genauer gesagt handelt es sich bei einem stochastischen Prozess um eine Familie von Zufallsvariablen.

Für eine stetige, kontinuierlich variierende Zeitvariable $t \in I = [0, T] \subseteq \mathbb{R}$ bezeichnet $\{X(t)|t \in I\}$ einen stetigen stochastischen Prozess.

2.1. Wiener Prozess

Ein Wiener-Prozess, auch Brownsche Bewegung genannt, ist ein zeitstetiger stochastischer Prozess, dessen Zuwächse $dW(t)$, mit denen sich der Prozess über die Zeit t ändert, unabhängig und normalverteilt mit Mittelwert 0 und dem Zeitintervall dt entsprechende Varianz sind. Das heißt es gilt:

$$dW(t) \sim N(0, dt). \quad (2.1.1)$$

Die Verteilung der Zuwächse eines Wiener Prozesses lässt sich mittels einer standardnormalverteilten Zufallsvariablen simulieren. Es gilt:

$$dW(t) \sim \sqrt{dt}Z(t) \quad \text{mit} \quad Z(t) \sim N(0, 1). \quad (2.1.2)$$

Das zeigt, dass die Zuwächse des Wiener Prozesses entsprechend einer \sqrt{dt} -mal standardnormalverteilten Zufallsvariablen verteilt sind. Dies ist bedeutsam für die Implementierung stochastischer Differentialgleichungen.

Eine weitere Eigenschaft des Wiener Prozesses ist, dass die nichtüberlappenden Zuwächse unabhängig sind. Es gilt demnach für $0 \leq t_1 < t_2 \leq t_3 < t_4$

$$W_{t_2} - W_{t_1} \quad \text{und} \quad W_{t_4} - W_{t_3} \quad \text{sind unabhängig.} \quad (2.1.3)$$

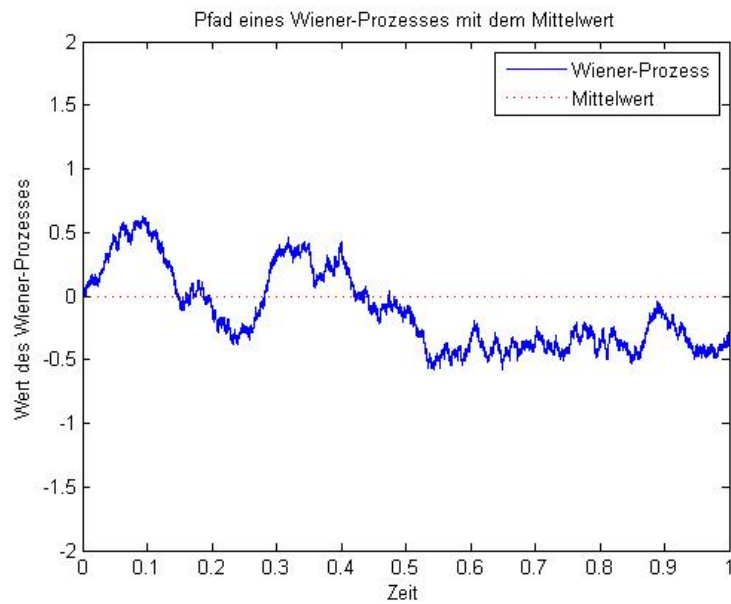


Abbildung 1: Pfad eines Wiener Prozesses

Mit dem nachstehenden Algorithmus kann man einen Pfad eines Wiener Prozesses simulieren:

```

% Simulation eines Pfades des Wiener Prozesses
Input h = 0.01   % Schrittweite
W(1) = 0   % Startwert
for i = 1 : 100
W(i + 1) = W(i) + randn * sqrt(h)
end

```

Der Befehl “randn” erzeugt standardnormalverteilte Zufallsvariablen in Matlab. In Abbildung (4.5.3) wird ein Pfad eines Wiener Prozesses und seinem Mittelwert dargestellt.

Als nächstes wird ein weiterer wichtiger stochastischer Prozess eingeführt, der Ito-Prozess.

Definition [Ito-Prozess]

Ein stochastischer Prozess $X(t)$, der folgender stochastischen Differentialgleichung

$$dX(t) = a(X, t)dt + b(X, t)dW(t) \quad (2.1.4)$$

mit $dW(t) \sim Z(t)\sqrt{dt}$ und $Z(t) \sim N(0, 1)$ genügt, heißt Ito-Prozess. $W(t)$ stellt einen Wiener Prozess dar und $Z(t)$ ist eine standardnormalverteilte Zufallsvariable. Die stochastische Differentialgleichung (2.1.4) wird auch Ito-stochastische Differentialgleichung

genannt. Der Parameter $a(X, t)$ heißt Drift- und der Parameter $b(X, t)$ Diffusionskoeffizient. Diese können sowohl von der Zeit t , als auch von der stochastischen Variablen $X(t)$ abhängen. Durch die Wahl der Parameter $a(X, t)$ und $b(X, t)$ kann man den in Definition (2.3) beschriebenen Prozess $X(t)$ spezifizieren.

Satz [Erwartungswert und Varianz der Ito-Gleichung]

[vgl. [3], p.20]

Für den Erwartungswert und die Varianz der Ito-Gleichung gilt:

$$E[dX(t)] = a(X, t)dt \quad (2.1.5)$$

$$Var[dX(t)] = E[dX(t) - E[dX(t)]]^2 = E[(b(x, t)dW(t))^2] = b(X, t)^2 dt \quad (2.1.6)$$

2.2. Lemma von Ito

Sei $X(t)$ ein Ito-Prozess und $f : \mathbb{R}_+ \times \mathbb{R} \rightarrow \mathbb{R}$ eine in der ersten Komponente einfach und in der zweiten Komponente zweifach stetig differenzierbare Funktion, so ist auch $Y_t := f(X_t, t)$ ein Ito-Prozess und es gilt:

$$df = \left(\frac{\partial f}{\partial t} + a \frac{\partial f}{\partial x} + \frac{1}{2} b^2 \frac{\partial^2 f}{\partial x^2} \right) dt + b \frac{\partial f}{\partial x} dW. \quad (2.2.1)$$

Beweis

Eine Beweisskizze dieses Lemmas findet man beispielsweise in [3].

2.3. Spezielle stochastische Prozesse

In diesem Kapitel der Bachelorarbeit werden einige spezielle stochastische Prozesse vorgestellt. Auf den Wurzel-Diffusionsprozess wird im Kapitel 4.0.1 näher eingegangen. Die Abbildungen in diesem Kapitel wurden mittels dem Euler-Maruyama-Verfahren erstellt. Dieses sowie andere Diskretisierungsverfahren werden im nächsten Kapitel erläutert.

Prozesse, die auf konstanten Koeffizienten beruhen

Bei Prozessen, die auf stochastischen Differentialgleichungen mit konstanten Koeffizienten beruhen werden der Drift- und Diffusionsparameter konstant gewählt. Sie sind somit weder von der Zeit t , noch von der stochastischen Zufallsvariable $X(t)$ abhängig. Dies ist der einfachste Fall einer stochastischen Differentialgleichung. Es gilt:

$$a(x, t) = \mu, b(X, t) = \sigma \rightarrow dX(t) = \mu dt + \sigma dW(t)$$

Es ergibt sich nach (2.1) für den Erwartungswert und die Varianz

$$E[dX(t)] = \mu dt \quad \text{und} \quad Var[dX(t)] = \sigma^2 dW(t).$$

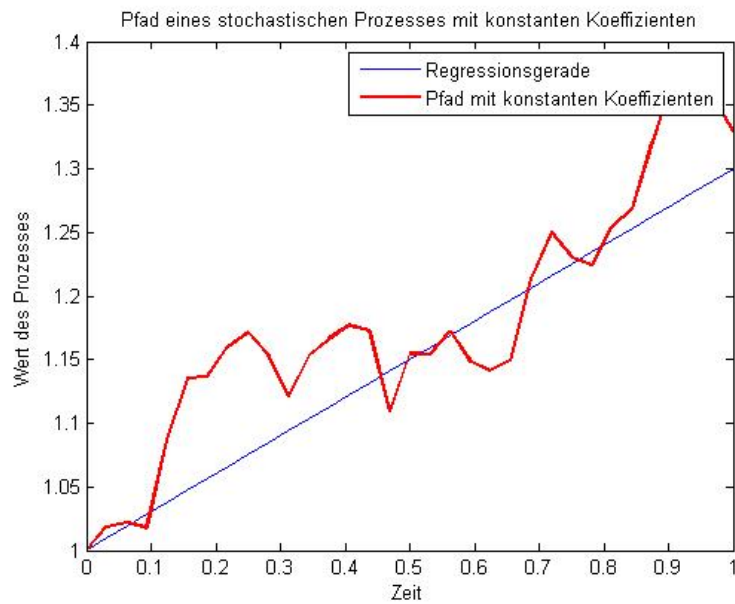


Abbildung 2: Prozess mit konstanten Koeffizienten

Entsprechend des Erwartungswerts seiner Änderungen entwickelt sich der Pfad des Prozesses um die Trendgerade $b(t) = \mu * t + X(0)$. Die Schwankungen um diese Gerade bleiben gleich stark auf Grund des konstanten Diffusionsparameters. Die Abbildung (2) zeigt einen Pfad des Prozesses mit $X(0) = 1$, $\mu = 0.3$ und $\sigma = 0.2$. In Blau eingezeichnet ist die eben erläuterte Trendgerade, um die der Prozess schwankt.

Prozesse mit geometrischer stochastischer Differentialgleichung

In diesem Fall sind der Drift- und der Diffusionsparameter von den von der Zeit t verfügbaren Informationen abhängig. Es gilt:

$$a(x, t) = \mu X(t), b(X, t) = \sigma X(t) \rightarrow dX(t) = \mu X(t)dt + \sigma X(t)dW(t)$$

Für den Erwartungswert und die Varianz bei Prozessen mit geometrischer stochastischer Differentialgleichung ergibt sich nach (2.1):

$$E[dX(t)] = \mu X(t)dt \quad \text{und} \quad Var[dX(t)] = \sigma^2 X(t)^2 dW(t).$$

Die Pfade des Prozesses folgen einem exponentiellen Trend. Wir erhalten für die Trendgerade in diesem Fall $a(t) = e^{\mu t}$. Die Schwankungen des Prozesses um diese Gerade nehmen mit der Zeit t entsprechend der Proportionalität der Varianz zu $X(t)^2$ zu, da auch $X(t)$ steigt. Dieser Prozesstyp eignet sich für die Modellierung von Aktienkursen. In Abbildung (3) ist ein Pfad des Prozesses mit geometrischer stochastischer Differentialgleichung mit $X(0) = 1$, $\mu = 0.3$ und $\sigma = 0.2$ und die Trendgerade $a(t) = e^{\mu t}$ abgebildet.

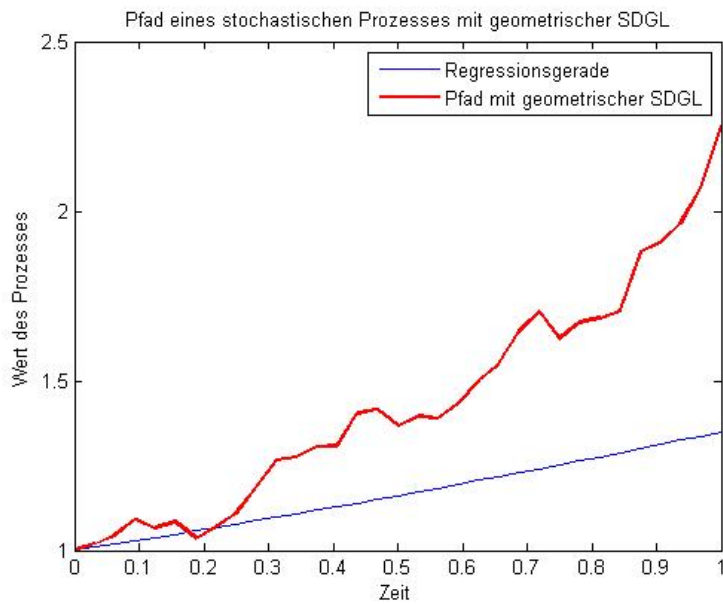


Abbildung 3: Prozess mit geometrischer SDGL

Wurzel-Prozess

Auch in diesem Fall hängen der Drift- und der Diffusionsparameter von den von der Zeit t verfügbaren Informationen ab. Es gilt:

$$a(x, t) = \mu X(t), b(X, t) = \sigma \sqrt{X(t)} \rightarrow dX(t) = \mu X(t)dt + \sigma \sqrt{X(t)}dW(t)$$

Für Erwartungswert und Varianz gilt hier nach (2.1):

$$E[dX(t)] = \mu X(t)dt \quad \text{und} \quad Var[dX(t)] = \sigma^2 X(t)dW(t).$$

Man erkennt schnell, dass dieser Prozess sich nur in der Varianz vom geometrischen Prozess unterscheidet. Aus diesem Grund ist die Trendgerade hier gleich der im geometrischen Fall $a(t) = e^{\mu t}$. Die Schwankungen fallen entsprechend schwächer aus, da die Varianz hier nur proportional zu $X(t)$ ist. Abbildung (4) zeigt genau das. Hier ist auch wieder $X(0) = 1$, $\mu = 0.3$ und $\sigma = 0.2$. Die Trendgerade ist $a(t) = e^{\mu t}$.

Mean-Reversion-Prozess

Auch im Fall der Mean-Reversion-Prozesse sind der Drift- und der Diffusionskoeffizient von den von der Zeit t verfügbaren Informationen abhängig. Hier gilt:

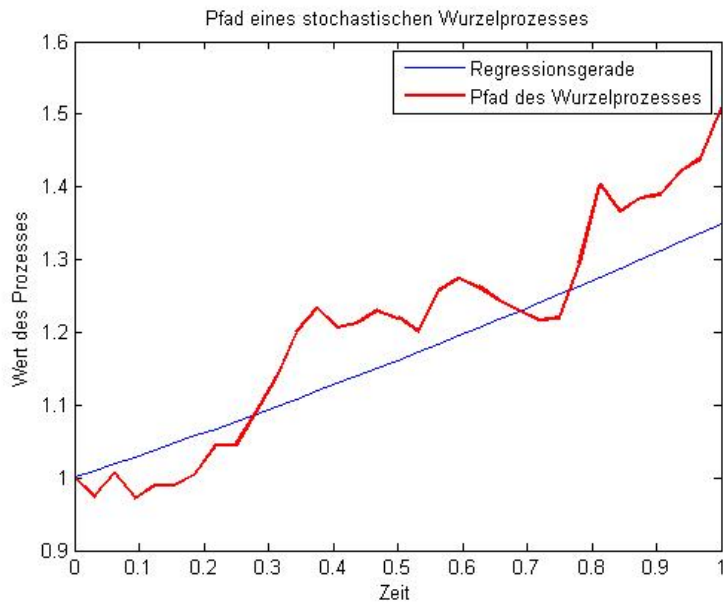


Abbildung 4: Wurzel-Prozess

$$a(x, t) = \kappa(\theta - X(t)), b(X, t) = \sigma X(t) \rightarrow dX(t) = \kappa(\theta - X(t))dt + \sigma X(t)dW(t)$$

Für den Erwartungswert und die Varianz gilt hier, nach (2.1):

$$E[dX(t)] = \kappa(\theta - X(t))dt \quad \text{und} \quad Var[dX(t)] = \sigma^2 X(t)^2 dW(t).$$

Mean-reversion-Effekt

Mean-Reversion-Prozesse besitzen in dem Driffterm die folgende Eigenschaft:

Wenn der Prozess $X(t)$ unter das Gleichgewichtsniveau θ kommt, dann besitzt der Prozess eine positive Driftrate, geht der Prozess über das Gleichgewichtsniveau, so hat er dementsprechend eine negative Driftrate. Das bedeutet, dass der Prozess von der Stärke der Regulierungsfunktion κ in Abhängigkeit der Entfernung des Prozesses zum Gleichgewichtsniveau wieder zum Gleichgewichtsniveau gezogen wird. Aus dieser Eigenschaft folgt, dass der Prozess in Erwartung stets gegen das Gleichgewichtsniveau driftet. Dieser tendenzielle Zug wird durch den stochastischen Diffusionsanteil überlagert. So ist es möglich, dass man trotz negativer Driftrate, durch hohen Diffusionseinfluss, einen positiven Zuwachs erhält. Je weiter sich der Prozess vom Gleichgewichtsniveau entfernt, desto größer wird die Wahrscheinlichkeit, dass der Prozess $X(t)$ im nächsten Schritt in Richtung θ zurückkehrt. Dieser Effekt wird Mean-reversion-Effekt genannt.

Wählt man $\kappa < 0$, so hätte der Prozess völlig andere Eigenschaften:

Für $X(0) > \theta$ würde der Prozess $X(t)$ gegen $+\infty$ divergieren. Anderenfalls, also $X(0) < \theta$, würde der Prozess $X(t)$ schnell negative Werte annehmen. Dann würde der Prozess

seinen eigenen Definitionsbereich verlassen.

Für $\kappa \geq 0$ gilt für alle $t \geq 0$:

$$P(X(t) \geq 0) = 1. \quad (2.3.1)$$

3. Diskretisierungsverfahren

Nachdem wir jetzt einige Typen von stochastischen Differentialgleichungen kennengelernt haben, beschäftigt sich dieses Kapitel mit verschiedenen Diskretisierungsverfahren. Außerdem wird der Begriff der Konvergenzordnung eingeführt. Die in diesem Kapitel verwendeten Definitionen und Sätze wurden aus [5] und [10] entnommen. Wenn eine andere Literaturquelle verwendet wird, wird dies separat gekennzeichnet. Im vorherigen Kapitel wurde die stochastische Differentialgleichung nach Ito bereits vorgestellt. Es gilt:

$$dX(t) = a(X, t)dt + b(X, t)dW(t). \quad (3.0.2)$$

Die Interpretation dieser Gleichung ist eine Integralgleichung der Form

$$X(t) = X(t_0) + \int_{t_0}^t a(X, s)ds + \int_{t_0}^t b(X, s)dW(s). \quad (3.0.3)$$

Das $W(t)$ in unseren Gleichungen stellt einen Wiener Prozess dar. Das erste Integral der Integralgleichung ist ein pfadweises Riemann-Integral, das zweite Integral ist ein stochastisches Ito-Integral. Die Gleichung besitzt eine eindeutige Lösung, wenn $a, b : [t_0, T] \times \mathbb{R} \rightarrow \mathbb{R}$ den Voraussetzungen des Existenz- und Eindeutigkeitssatz genügen. Die Lösung $X(t)$ ist ein stochastischer Prozess, welcher der stochastischen Integralgleichung der stochastischen Differentialgleichung genügt, mit den folgenden Eigenschaften:

- (1) $X(t)$ ist ZV mit $E(X(t)^2) < \infty$ für jedes $t \in [t_0, T]$
- (2) $X(t)$ und $W_{t+h} - W_t$ sind unabhängig für jedes $h > 0, t \in [0, T]$
- (3) Pfade $t \rightarrow X(t)(\omega)$ sind fast sicher stetig.

3.1. Existenz- und Eindeutigkeitssatz

1. Die Koeffizienten $f, g : [t_0, T] \times \mathbb{R} \rightarrow \mathbb{R}$ genügen den folgenden Eigenschaften:

- stetig in (t, x)
- Linearwachstum

$$|f(t, x)| \leq k(1 + |x|)$$

$$|g(t, x)| \leq k(1 + |x|)$$

- Lipschitz-Bedingung

$$|f(t, x) - f(t, y)| \leq L|x - y|$$

$$|g(t, x) - g(t, y)| \leq L|x - y|$$

gleichmäßig in $t \in [t_0, T]$.

2. Die Anfangszufallsvariable X_{t_0} sei nicht antizipativ bzgl. des Wiener Prozesses W_t mit

$$E(X_{t_0}^2) < \infty.$$

Unter diesen beiden Voraussetzungen besitzt die Ito-stochastische Differentialgleichung

$$dX_t = f(t, X_t)dt + g(t, X_t)dW_t$$

eine eindeutige Lösung auf $[t_0, T]$ mit Anfangswert X_{t_0} .

Beweis

Eine Idee des Beweises gibt [10].

3.2. Euler-Verfahren

Herleitung

Die Herleitung des stochastischen Euler-Verfahrens hat dieselbe Motivation, wie die Herleitung des deterministischen Euler-Verfahrens. Dazu betrachten wir die stochastische Differentialgleichung (Ito-SDGL)

$$dX(t) = a(X, t)dt + b(X, t)dW(t)$$

auf dem Intervall $[t_0, T]$ mit Anfangswert $X(0)$. Die Lösung $X(t)$ genügt der stochastischen Ito-Integralgleichung

$$X(t) = X(0) + \int_{t_0}^t a(X, s)ds + \int_{t_0}^t b(X, s)dW(s)$$

auf dem Intervall $[t_0, T]$, sowie der Integralgleichung

$$X(t_{n+1}) = X(t_n) + \int_{t_n}^{t_{n+1}} a(X, s)ds + \int_{t_n}^{t_{n+1}} b(X, s)dW(s)$$

auf dem Teilintervall $[t_n, t_{n+1}]$ von $[t_0, T]$. Wie im deterministischen Fall kann man jetzt die Integrandenfunktionen a und b zu der Auswertungsstelle $t_n = s$ "einfrieren". Daraus erhalten wir die folgende Approximation

$$\begin{aligned} X(t_{n+1}) &\approx X(t_n) + \int_{t_n}^{t_{n+1}} a(X, t_n)ds + \int_{t_n}^{t_{n+1}} b(X, t_n)dW(s) \\ &= X(t_n) + a(X, t_n) \int_{t_n}^{t_{n+1}} 1 \cdot ds + b(X, t_n) \int_{t_n}^{t_{n+1}} 1 \cdot dW(s) \\ &= X(t_n) + a(X, t_n)\Delta_n + b(X, t_n)\Delta W_n \end{aligned}$$

mit

$$\Delta_n = t_{n+1} - t_n = \int_{t_n}^{t_{n+1}} 1 \cdot ds \text{ mit einem Fehler } O(h)$$

$$\Delta W_n = W(t_{n+1}) - W(t_n) = \int_{t_n}^{t_{n+1}} 1 \cdot dW(s) \text{ mit einem Fehler } O(\sqrt{h})$$

Diese Approximation scheint konsistent mit der Definition des Ito-Integrals zu sein.

Euler-Verfahren

Aus der obigen Approximation erhalten wir das stochastische Euler-Verfahren.

$$X(n+1) = X(n) + a(X(n))\Delta_n + b(X(n))\Delta W(n),$$

$$n = 0, 1, 2, \dots, N-1$$

Das stochastische Euler-Verfahren ist das einfachste Diskretisierungsverfahren für stochastische Differentialgleichungen. Dieses Verfahren wird auch Euler-Maruyama-Verfahren genannt. Das Euler-Verfahren ist eine stochastische Differentialgleichung, die eine Zufallsvariable $X(n)$ in eine neue Zufallsvariable $X(n+1)$ abbildet. Man kann jedoch nur einzelne Pfade des stochastischen Prozesses berechnen, das heißt $X_n(\omega) \rightarrow X_{n+1}(\omega)$ für festes ω . Dafür benötigt man die entsprechenden Zuwächse $\Delta W_n(\omega)$ des Wiener Prozesses W_t . Diese Berechnungen laufen rekursiv. Man kann die Zuwächse z.B. durch einen Pseudo-Zufallsvariablen-Erzeuger simulieren.

3.3. Milstein-Verfahren

Das zweite Diskretisierungsverfahren, welches vorgestellt werden soll, ist das Milstein-Verfahren. Dieses Verfahren erweitert den Drift und den Diffusionsterm auf einen Fehler von $O(h)$. Das Milstein-Verfahren ist eine Verfeinerung des Euler-Verfahrens und sieht wie folgt aus

$$X(n+1) = X(n) + a(X(n))\Delta_n + b(X(n))\Delta W(n) + \frac{1}{2}b'(X(n))b(X(n))\{(\Delta W(n))^2 - \Delta_n\}$$

Im Vergleich zum Euler-Verfahren kommt nur ein zusätzlicher Term dazu.

Herleitungsskizze des zusätzlichen Terms

Der zusätzliche Term $+\frac{1}{2}b'(X(n))b(X(n))\{(\Delta W(n))^2 - \Delta_n\}$ entsteht durch einen weiteren Schritt der stochastischen Taylorentwicklung. Wir verwenden die Ito-Formel auf das Doppelintegral

$$\int_0^t \int_0^s L^1 b(X_\tau) dW_\tau dW_s$$

des Restterms und erhalten einen neuen Term $L^1 b(X_0) \int_0^t \int_0^s dW_\tau dW_s$ der Taylor-Approximation. Durch den Hilfssatz: $\int_{t_n}^{t_{n+1}} \int_{t_n}^t dW_s dW_t = \frac{1}{2}(\Delta W_n)^2 - \frac{1}{2}\Delta_n$ und der Bemerkung, dass $L^1 b(t, x) = b(t, x) \frac{\partial b}{\partial x}$ ist, erhalten wir das Milstein-Verfahren.

3.4. Verbesserung zum Euler-Verfahren

Das Milstein-Verfahren ist eine Verbesserung zum Euler-Verfahren im starken Sinn. Der Grund dafür ist, dass das doppelte stochastische Integral mehr Informationen über das Verhalten des Wiener Prozesses innerhalb des Teilintervalls $[t_n, t_{n+1}]$ enthält als die einfachen Zuwächse $\Delta W_n = W_{t_{n+1}} - W_{t_n} = \int_{t_n}^{t_{n+1}} dW_s$ des Euler-Verfahrens.

3.5. Konvergenzordnung

Es gibt zwei Arten von stochastischen Approximationen. Wir unterscheiden zwischen starker Konvergenz, die Approximation der Pfade und schwacher Konvergenz, die Approximation der Wahrscheinlichkeitsmaße. Für die einen soll die Realisierung nah sein, für die anderen Anwendungen reicht es, wenn die Wahrscheinlichkeiten verschiedener Ereignisse nah ist.

Starke Konvergenz

Typische Kriterien für starke Konvergenz sind:

$$\begin{aligned} E(\|X_n^{(\Delta)} - X_{t_n}\|^2) &\rightarrow \text{qm-Fehler} \\ E(\|X_n^{(\Delta)} - X_{t_n}\|) &\rightarrow \text{erster Moment} \end{aligned}$$

In der Praxis ist der quadratische Mittelwertfehler günstig, da er z.B. technisch leichte Abschätzungen liefert. Wir verwenden aber den Fehler des ersten Moments, da es die natürliche Verallgemeinerung des deterministischen Diskretisierungsfehlers ist.

Satz Starke Konvergenz

Wir sagen, eine Diskretisierung $X(t_n)$ besitzt starke Konvergenzordnung $\beta > 0$, falls

$$E(\max_n \|X_n^{(\Delta)} - X_{t_n}\|) \leq K_T \Delta^\beta,$$

für eine Konstante K , die von T abhängt und alle hinreichend kleine Δ .

Satz Schwache Konvergenz

Das Diskretisierungsverfahren hat schwache Konvergenzordnung $\beta > 0$, falls

$$\max_n \|E(P(X_n^{(\Delta)})) - E(P(X_{t_n}))\| \leq K_{P,T} \Delta^\beta$$

wobei die Konstante hier von T und von P abhängt. Sie gilt für alle Polynome P (oder einer anderen geeigneten Klasse von Testfunktionen).

Absoluter Fehler

[vgl. [6], pp.118-123]

Der absolute Fehler von $X_n^\Delta - X_{t_n}$ zur Zeit T ist definiert durch

$$\epsilon(\Delta) = E(|X_n^\Delta - X_{t_n}|). \quad (3.5.1)$$

Aber wie wird der Fehler (3.5.1) konkret berechnet?

Sind X_1, \dots, X_n als Stichproben einer Zufallsvariablen X gegeben, so verwendet man den Schätzer

$$\Theta_n = \frac{1}{n} \sum_{i=1}^n X_i$$

für den Erwartungswert $E(X)$. Dieser Schätzer ist erwartungstreu, das heißt es gilt

$$E(\Theta_n) = \frac{1}{n} \sum_{i=1}^n E(X_i) = \frac{1}{n} \sum_{i=1}^n E(X) = E(X).$$

Und die Varianz des Schätzers konvergiert gegen Null für $n \rightarrow \infty$.

$$Var(\Theta_n) = \frac{1}{n^2} \sum_{i=1}^n Var(X_i) = \frac{1}{n} Var(X) \rightarrow 0.$$

Wir wollen nun das Euler-Maruyama-Verfahren, angewandt auf die stochastische Differentialgleichung

$$dX_t = \mu X_t dt + \sigma X_t dW$$

empirisch auf starke Konvergenz testen. Dieser Test nennt sich lineare Ausgleichsrechnung oder lineare Regressionsanalyse. Dazu gehen wir davon aus, dass wir die Ungleichung

$$\epsilon(\Delta) \leq K \Delta^\beta$$

näherungsweise als Gleichung auffassen können. Durch logarithmieren erhalten wir

$$\log \epsilon(\Delta) \approx \log K + \beta \log \Delta.$$

Plotten wir nun also $\epsilon(\Delta)$ und Δ mit doppel-logarithmischer Skala, so können wir die Konvergenzordnung als die Steigung der Geraden

$$y(x) \approx \log C + \beta x$$

mit $y(x) = \log \epsilon(\Delta)$ und $x = \log \Delta$ auffassen. Diese Gerade wird auch lineare Ausgleichsgerade oder Regressionsgerade genannt.

Methode der kleinsten Quadrate

Da die Punkte $(\log \Delta, \log \epsilon(\Delta))$ in der Praxis nicht exakt auf der eben beschriebenen Geraden liegen werden, bestimmt man die Geradengleichung γ mittels der Methode der kleinsten Quadrate.

Man berechnet $z^* = (\log C, \gamma)$ als Lösung des folgenden Minimierungsproblem

$$z^* = \operatorname{argmin}_{z \in \mathbb{R}} \|Az - b\|_2.$$

Hier ist $\|\cdot\|_2$ die euklidische Norm. Es seien

$$A = \begin{pmatrix} 1 & \log \Delta_1 \\ \vdots & \vdots \\ 1 & \log \Delta_m \end{pmatrix} \in \mathbb{R}^{m \times 2}, \quad (3.5.2)$$

$$b = \begin{pmatrix} \log \epsilon(\Delta_1) \\ \vdots \\ \log \epsilon(\Delta_m) \end{pmatrix} \in \mathbb{R}^m. \quad (3.5.3)$$

Das Argumentenminimum bedeutet, dass z^* das folgende Problem löst :

$$\|Az^* - b\|_2 = \min_{z \in \mathbb{R}^2} \|Az - b\|_2.$$

z^* kann man direkt über eine QR-Zerlegung lösen. In Matlab kann die Methode zur Lösung des Minimierungsproblem durch den Befehl $A \backslash b$ gelöst werden. Als Ergebnis erhält man einen zweidimensionalen Vektor, dessen zweiter Eintrag die Steigung der Gerade angibt.

Beispiel

In Abbildung (5) kann man jetzt die starken Konvergenzordnungen des Euler-Maruyama-Verfahrens und des Milstein-Verfahrens, angewandt auf die stochastische Differentialgleichung

$$dX_t = \mu X_t dt + \sigma X_t dW$$

mit $\mu = -0.5$, $\sigma = 0.9$, $X_0 = 1$ und $T = 1$, erkennen. Zusätzlich wurden Kontrolllinien mit den Steigungen $-\frac{1}{2}$ und -1 in die Abbildung eingezeichnet. Man sieht, dass das Euler-Maruyama-Verfahren eine starke Konvergenz von $\frac{1}{2}$ hat und das Milstein-Verfahren hat eine starke Konvergenz von 1. Die lineare Ausgleichsrechnung ergibt bei 500 Runden:

| Verfahren | γ | Residuum |
|--------------------------|----------|----------|
| Euler-Maruyama-Verfahren | 0.4505 | 0.2495 |
| Milstein-Verfahren | 1.0516 | 0.2142 |

Um die Ausgleichsrechnung für die schwache Konvergenzordnung zu bestimmen, geht man analog zu der oben vorgestellten Methode vor. Nur das Fehlerkriterium muss angepasst werden. In Abbildung (6) sieht man die Ausgleichsgeraden für die schwachen Konvergenzen des Euler-Maruyama-Verfahrens und des Milstein-Verfahrens. Da wir vermuten, dass beide Verfahren eine schwache Konvergenzordnung 1 haben, ist zusätzlich eine Kontrolllinie mit der Steigung -1 eingezeichnet. Man kann gut erkennen, dass beide Ausgleichsgeraden der Verfahren eine Steigung von -1 haben, was bedeutet, dass unsere Vermutung richtig war. Tatsächlich liefert die lineare Ausgleichsrechnung bei 500 Runden:

| Verfahren | γ | Residuum |
|--------------------------|----------|----------|
| Euler-Maruyama-Verfahren | 0.5748 | 3.4001 |
| Milstein-Verfahren | 0.9549 | 0.2455 |

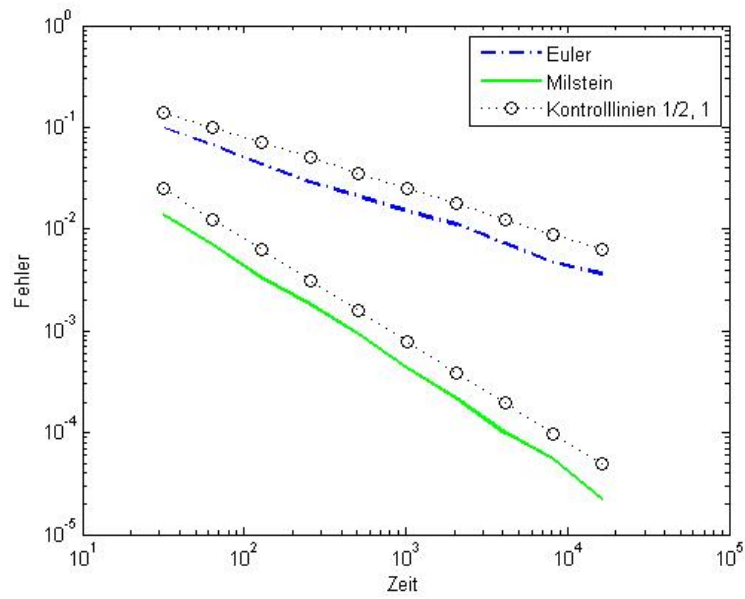


Abbildung 5: starke Konvergenzordnung

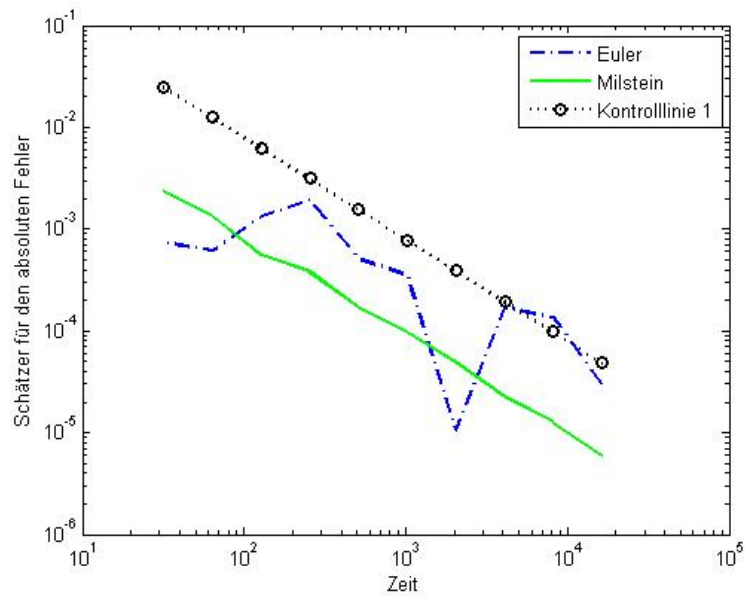


Abbildung 6: schwache Konvergenzordnung

3.6. Methoden 2. Ordnung

Betrachte Ito-SDGL

$$dX(t) = a(X, t)dt + b(X, t)dW(t)$$

Eine Lösung $X(t)$ auf dem Intervall $[0, T]$ genügt der stochastischen Integralgleichung

$$X(t) = X(0) + \int_0^t a(X, s)ds + \int_0^t b(X, s)dW(s)$$

Für jedes zweimal stetig differenzierbare U lautet die stochastische Kettenregel

$$dU(X, t) = L^0U(X, t)dt + L^1U(X, t)dW(t)$$

oder in Integralform

$$U(X, t) = U(X(0)) + \int_0^t L^0U(X, s)ds + \int_0^t L^1U(X, s)dW(s)$$

mit

$$L^0U = \frac{\partial U}{\partial t} + a \frac{\partial U}{\partial x} + \frac{1}{2} b^2 \frac{\partial^2 U}{\partial x^2}$$

$$L^1U = b \frac{\partial U}{\partial x}$$

Jetzt haben wir 2 Integrale, die wir weiterentwickeln können.

$$U \equiv a \quad \text{im Riemann-Integral}$$

$$\int_0^t a(X, s)ds = \int_0^t \left(a(X, 0) + \int_0^s L^0 a(X, \tau) d\tau + \int_0^s L^1 a(X, \tau) dW_\tau \right) ds$$

$$U \equiv b \quad \text{im Ito-Integral}$$

$$\int_0^t b(X, s)dW_s = \int_0^t \left(b(X, 0) + \int_0^s L^0 b(X, \tau) d\tau + \int_0^s L^1 b(X, \tau) dW_\tau \right) dW_s$$

Daraus erhalten wir den ersten Schritt der Taylor-Approximation

$$X(t) = X(0) + a(X, 0) \int_0^t dt + b(X, 0) \int_0^t dW_s$$

$$+ \int_0^t \int_0^s L^0 a(X, \tau) d\tau ds + \int_0^t \int_0^s L^1 a(X, \tau) dW_\tau ds$$

$$+ \int_0^t \int_0^s L^0 b(X, \tau) d\tau dW_s + \int_0^t \int_0^s L^1 b(X, \tau) dW_\tau dW_s$$

Einen Schritt weiter erhalten wir

$$\begin{aligned}
X(t) &= X(0) + a(X, 0) \int_0^t dt + b(X, 0) \int_0^t dW_s \\
&+ L^0 a(X, 0) \int_0^t \int_0^s d\tau ds + L^1 a(X, 0) \int_0^t \int_0^s dW_\tau ds \\
&+ L^0 b(X, 0) \int_0^t \int_0^s d\tau dW_s + L^1 b(X, 0) \int_0^t \int_0^s dW_\tau dW_s \\
&\quad + \text{neuer Rest}
\end{aligned}$$

Der neue Restterm besteht jetzt aus Dreifach-Integralen. Durch das Einsetzen der Operatoren L^0 und L^1 erhalten wir die Approximation

$$\begin{aligned}
X(n+1) &\approx X(n) + a\Delta_n + b\Delta W + (aa' + \frac{1}{2}b^2a'')I_{(0,0)} \\
&\quad + (ab' + \frac{1}{2}b^2b'')I_{(0,1)} + ba'I_{(1,0)} + bb'I_{(1,1)}
\end{aligned}$$

mit $\Delta W = W(t_{n+1}) - W(t_n)$. Um das Verfahren anwenden zu können, müssen wir jetzt noch die stochastischen Multi-Integrale auswerten.

3.7. Simulation stochastischer Multi-Integrale

[vgl. [5], pp.350-351 und [9], pp.168-172]

- $I_{(0,0)} = \int_{t_n}^{t_{n+1}} \int_{t_n}^s d\tau ds = \frac{1}{2}\Delta_n^2$
- $I_{(1,1)} = \int_{t_n}^{t_{n+1}} \int_{t_n}^s dW_\tau dW_s = \frac{1}{2}[(\Delta W)^2 - \Delta_n]$
- Für stochastische Doppelintegrale mit $i_1 = i_2$ haben wir die "einfache" Formel

$$\begin{aligned}
&\int_{t_n}^{t_{n+1}} \int_{t_n}^s dW_\tau dW_s \\
&= \frac{1}{2} \left(\int_{t_n}^{t_{n+1}} dW_s \right)^2 - \frac{1}{2} \int_{t_n}^{t_{n+1}} dt \\
&= \frac{1}{2} [(\Delta W)^2 - \Delta_n]
\end{aligned}$$

Einfach im Sinne, dass sie nur die einfachen Zuwächse $\Delta W, \Delta_n$ enthält.

- $I_{(0,1)} = \int_{t_n}^{t_{n+1}} \int_{t_n}^s d\tau dW(s) = \int_{t_n}^{t_{n+1}} (s-t)dW(s)$

- Durch pfadweise Integration erhalten wir

$$\begin{aligned}
I_{(0,1)} &= \Delta_n W(t_{n+1}) - \int_{t_n}^{t_{n+1}} W(s) ds \\
&= \Delta_n [W(t_{n+1}) - W(t_n)] - \int_{t_n}^{t_{n+1}} [W(s) - W(t_n)] ds \\
&= \Delta_n \Delta W - I_{(1,0)}
\end{aligned}$$

Jetzt müssen wir uns nur noch

$$I_{(1,0)} = \int_{t_n}^{t_{n+1}} [W(s) - W(t_n)] ds$$

anschauen. Wir wissen: Es gilt Identität $I_{(1,0)} + I_{(0,1)} = I_{(0)} I_{(1)}$, ausserdem kennen wir die Verteilungen $I_{(0,1)} \sim N(0, \frac{\Delta_n^3}{3})$. Und zusätzlich kennen wir die Korrelation $E[I_{(1,0)} I_{(1)}] = \frac{\Delta_n^2}{2}$. Wir können ΔW und $I_{(1,0)}$ mit ΔW und ΔZ mit folgender Verteilung

$$\begin{pmatrix} \Delta W \\ \Delta Z \end{pmatrix} \sim N\left(0, \begin{pmatrix} \Delta_n & \frac{1}{2} \Delta_n^2 \\ \frac{1}{2} \Delta_n^2 & \frac{1}{3} \Delta_n^3 \end{pmatrix}\right)$$

simulieren. Daraus erhalten wir das folgende Schema 2.Ordnung

$$\begin{aligned}
X(n+1) &= X(n) + a\Delta_n + b\Delta W + (ab' + \frac{1}{2}b^2b'')[\Delta W\Delta_n - \Delta Z] \\
&\quad + a'b\Delta Z + \frac{1}{2}bb''[\Delta W^2 - \Delta_n] \\
&\quad + (aa' + \frac{1}{2}b^2a'')\frac{1}{2}\Delta_n^2
\end{aligned}$$

3.8. Vektorwertige stochastische Differentialgleichungen

Wir haben die Darstellung

$$X_i(t_{n+1}) = X_i(t_n) + \int_{t_n}^{t_{n+1}} a_i(u) du + \sum_{k=1}^m \int_{t_n}^{t_{n+1}} b_{i,k} dW_k(u), \quad i = 1, \dots, d$$

mit X_i d-dimensionaler Zufallsvektor und W_k m-dimensionaler Wiener-Prozess. Sei $X(t) = (X_t^1, \dots, X_t^d)^T$ eine Lösung der d-dimensionalen Ito-SDGL

$$dX(t) = a(X, t)dt + \sum_{k=1}^m b_k(X, t)dW_t^k$$

wobei $W(t) = (W_t^1, \dots, W_t^m)^T$ ein m-dimensionaler Wiener-Prozess ist.

Sei $U : [0, T] \times \mathbb{R}^d \rightarrow \mathbb{R}^1$ und definiere $Y(t) = U(X, t)$. Dann lautet die Ito-Formel

$$dY(t) = L^0 U(X, t)dt + \sum_{k=1}^m L^k U(X, t)dW_t^k$$

oder in der Integralversion

$$U(X(t)) = U(X, 0) + \int_{t_0}^t L^0 U(X, s) ds + \sum_{k=1}^m \int_{t_0}^t L^k U(X, s) dW_s^k$$

mit den Operatoren

$$L^0 U = \frac{\partial U}{\partial t} + \sum_{i=1}^d a_i \frac{\partial U}{\partial x_i} + \frac{1}{2} \sum_{i,j=1}^d \sum_{k=1}^m b_{ik} b_{jk} \frac{\partial^2 U}{\partial x_i \partial x_j}$$

und

$$L^k = \sum_{i=1}^d b_{ik} \frac{\partial U}{\partial x_i}, \quad k = 1, \dots, m$$

Wir approximieren mit denselben Schritten wie im skalaren Fall. Daraus erhalten wir folgende Diskretisierung

$$\begin{aligned} X_i(t_{n+1}) &= X_i(t_n) + a_i(X, t_n) \Delta_n + \sum_{k=1}^m b_{ik}(X, t_n) \Delta W_k \\ &\quad + \frac{1}{2} L^0 a_i(X, t_n) I_{(k,0)} \\ &\quad + \sum_{k=1}^m \left(L^0 b_{ik}(X, t_n) I_{(0,k)} + \sum_{j=1}^m L^j b_{ik}(X, t_n) I_{(j,k)} \right) \end{aligned}$$

für jedes $i = 1, \dots, d$

3.9. Kommutativitätsbedingung

Oft können wir die Struktur einer SDGL benutzen, um das Verfahren zu vereinfachen und die Auswertung der Multi-Integrale zu vermeiden. Eine stochastische Differentialgleichung

$$dX(t) = a(X, t) dt + \sum_{j=1}^m b_j(X, t) dW_t^j$$

besitzt kommutatives Rauschen, wenn

$$L^i b_j(x) \equiv L^j b_i(x), \quad \forall t, \quad i, j \in \{1, \dots, m\}.$$

In diesem Fall benutzen wir die Identität

$$I_{(i,j)} + I_{(j,i)} = I_{(i)} I_{(j)} = \Delta W_n^i \Delta_n^j$$

um die Summe

$$L^i b_j(X(n)) I_{(i,j)} + L^j b_i(X(n)) I_{(j,i)}$$

zu vereinfachen. Wir erhalten

$$\begin{aligned}
 &= L^i b_j(X(n)) \{I_{(i,j)} + I_{(j,i)}\} \\
 &= L^i b_j(X(n)) I_{(i)} I_{(j)} \\
 &= L^i b_j(X(n)) \Delta W_n^i \Delta W_n^j
 \end{aligned}$$

Wenn die Kommutativitätsbedingung erfüllt ist, kann man die Auswertung der stochastischen Multi-Integrale vermeiden.

Allgemeines schwaches Taylor-Verfahren

Das allgemeine schwache Taylor-Verfahren der Ordnung $\beta = 1, 2, \dots$ hat die Form

$$X(t_{n+1}) \approx X(t) + \sum_{n=1}^{\beta} \sum_{j_1, \dots, j_n} L^{j_1} \dots L^{j_{n-1}} b_{j_n} I_{(j_1, \dots, j_n)}$$

Wenn $\beta = 1$ erhalten wir so das Euler-Verfahren.

Allgemeines starkes Taylor-Verfahren

Das allgemeine starke Taylor-Verfahren der Ordnung $\beta = \frac{1}{2}, 1, \frac{3}{2}, \dots$ hat die Form

$$X(t_{n+1}) \approx X(t) + \sum_{(j_1, \dots, j_n) \in A_\beta} L^{j_1} \dots L^{j_{n-1}} b_{j_n} I_{(j_1, \dots, j_n)}$$

Die Menge A_β ist wie folgt definiert: Ein Vektor A mit Indizes (j_1, \dots, j_n) liegt in A_β , wenn entweder (i) die Anzahl der Indizes n plus die Anzahl der Indizes, die gleich 0 sind kleiner oder gleich 2β sind, oder (ii) $n = \beta + \frac{1}{2}$ und alle n Indizes 0 sind.

4. Wurzel-Diffusionsprozess

Im Kapitel 2.3 wurden bereits Wurzel-Prozesse vorgestellt. In diesem Teil der Arbeit beschäftigen wir uns mit Wurzel-Diffusionsprozessen (square root diffusion) der Form

$$dX(t) = \kappa(\theta - X(t))dt + \sigma\sqrt{X(t)}dW(t). \quad (4.0.1)$$

Die amerikanischen Mathematiker Cox, Ingersoll und Ross schlugen in ihrer Arbeit "A theory of the term structure of interest rates" 1985 den Wurzel-Diffusionsprozess erstmalig als Modell für kurzfristige Zinssätze vor (short rate model). Daher wird der Prozess auch CIR-Modell genannt. Die Herangehensweise dieses Kapitels folgt der von [3].

4.1. Definition [CIR-Modell]

Das CIR-Modell ist durch die stochastische Differentialgleichung

$$dX(t) = \kappa(\theta - X(t))dt + \sigma\sqrt{X(t)}dW(t) \quad (4.1.1)$$

gegeben. Die Parameter $\kappa \geq 0, \theta > 0, \sigma$ seien reell und $W(t), t \geq 0$ sei ein Wiener Prozess. Bei diesem Prozess findet im Driftterm die Idee der Mean-Reversion Anwendung. Der beschriebene Prozess $X(t)$ wird mit der Mean-Reversion-Stärke κ zum Mean-Reversion-Niveau θ gezogen. Das CIR-Modell ist eine Zusammensetzung aus einem Mean-Reversion-Prozess sowie einem Wurzel-Prozess. Vorausgesetzt es gelten die folgenden Bedingungen

$$X(0) > 0 \quad \text{sowie} \quad 2\kappa\theta \geq \sigma^2,$$

so kann es beim CIR-Modell auf Grund der Proportionalität der Volatilität σ zu $\sqrt{X(t)}$ fast sicher zu keinen negativen Werten kommen. Wenn der Prozess die Nulllinie erreicht, das heißt $X(t) = 0$, wird auch der gesamte Diffusionsterm $\sigma\sqrt{X(t)}$ auf Null gesetzt. Dann kommt das Mean-Reversion-Prinzip zum Tragen und es können nur noch Werte auftreten, die sich näher am Mean-Reversion-Niveau befinden.

Es existiert keine geschlossene Lösung für diesen Prozess, was die analytische Umgänglichkeit des Prozess erschwert. Allerdings kann man mit Hilfe der Übergangsdichte, welche bekannt ist, Pfade exakt simulieren.

4.2. Übergangsdichte eines stochastischen Prozesses

Mit der Übergangsdichte $\rho(x)$ eines Prozesses kann die Wahrscheinlichkeit, dass der Prozess zum Zeitpunkt t_{n+1} einen Wert zwischen a und b annimmt, vorausgesetzt, dass der Prozess zum Zeitpunkt t_n gleich X war, angegeben werden.

$\rho(X', t_{n+1}|X, t_n)$ bezeichnet die Übergangsdichte, die die Wahrscheinlichkeit beschreibt, dass der Prozess zur Zeit t_n gleich X' , unter der eben genannten Bedingung, so gilt:

$$P(a \leq X(t_{n+1}) \leq b | X(t_n) = X) = \int_a^b \rho(X', t_{n+1}|X, t_n). \quad (4.2.1)$$

Alle betreffenden Informationen des Prozesses sind in der Übergangsdichte enthalten.

Die Übergangsdichte des CIR-Modells ist durch eine nichtzentrale Chi-Quadratverteilung gegeben.

4.3. Parameterbeschreibung

- θ ist das Gleichgewichtsniveau des Prozesses, auch Mean-Reversion-Level genannt.
- κ gibt die Stärke der Regulierungsfunktion an, man sagt auch Mean-Reversion-Speed.
- σ ist die Volatilität des Prozesses.

4.4. Proposition

Sei $X(0) \geq 0$. Wenn $2\kappa\theta \geq \sigma^2$ (Stabilitätsbedingung) gilt, so ist der Prozess fast sicher strikt positiv für alle $t \geq 0$.

Begründung

Der Grund für die in (4.4) gegebene Bedingung ist, wenn $X(t)$ gegen 0 konvergiert, dass auch der Diffusionsterm gegen 0 konvergiert. Dann wird der Prozess vom positiven Drift nach oben gezogen.

4.5. Chi-Quadratverteilung / nichtzentrale Chi-Quadratverteilung

[vgl. [5], pp.122-124]

Wie bereits erwähnt, hat das CIR-Modell als Übergangswahrscheinlichkeit die nichtzentrale Chi-Quadratverteilung. Diese wird in diesem Teil der Arbeit definiert und erläutert. Wenn ν eine positive ganze Zahl ist und Z_1, \dots, Z_ν unabhängige standardnormalverteilte Zufallsvariablen sind, so heißt die Verteilung von

$$Z_1^2 + \dots + Z_\nu^2$$

Chi-Quadratverteilung mit ν Freiheitsgraden. Im Zeichen \mathcal{X}_ν^2 . Die \mathcal{X}_ν^2 -Verteilung ist gegeben durch

$$P(\mathcal{X}_\nu^2 \leq y) = \frac{1}{2^{\frac{\nu}{2}} \Gamma(\frac{\nu}{2})} \int_0^y e^{-\frac{z}{2}} z^{\frac{\nu}{2}-1} dz \quad (4.5.1)$$

mit einer Gamma-Verteilung Γ und es gilt $\Gamma(n) = (n-1)!$, wenn n eine positive ganze Zahl ist. Dieser Ausdruck definiert eine gültige Wahrscheinlichkeitsverteilung für $\nu > 0$ und ist damit auch die Definition für nicht-ganzzahlige ν .

Für $n \in \mathbf{N}$ und a_1, \dots, a_ν konstant ist die Verteilung von

$$\sum_{i=1}^{\nu} (Z_i + a_i)^2 \quad (4.5.2)$$

die nichtzentrale Chi-Quadratverteilung ($\mathcal{X}'_\nu{}^2$) mit ν Freiheitsgraden und $\lambda = \sum_{i=1}^{\nu} a_i^2$ dem Nichtzentralitätsparameter. Es gilt:

$$\begin{aligned} P(\mathcal{X}'_\nu{}^2(\lambda) \leq y) &= F_{\mathcal{X}'_\nu{}^2(\lambda)}(y) \\ &= e^{-\frac{\lambda}{2}} \sum_{j=0}^{\infty} \frac{(\frac{1}{2}\lambda)^j / j!}{2^{(\frac{\nu}{2})+j} \Gamma(\frac{\nu}{2} + j)} \int_0^y z^{(\frac{\nu}{2})+j-1} e^{-\frac{z}{2}} dz. \end{aligned} \quad (4.5.3)$$

für $y > 0$. Aus (4.5.3) folgt, dass für $\nu > 1$ und ν ganzzahlig gilt

$$\mathcal{X}'_\nu{}^2(\lambda) = \mathcal{X}'_{\nu-1}{}^2(\lambda) + \mathcal{X}_{\nu-1}^2. \quad (4.5.4)$$

Glasserman [5] sagt in seinem Buch, dass nach Johnson diese Darstellung auch für nicht-ganzzahlige $\nu > 0$ gilt. Für $\nu > 0$ zeigt Gleichung (4.5.4), dass eine nichtzentrale Chi-Quadratverteilte Zufallsvariable wie eine gewöhnliche Chi-Quadratverteilte Zufallsvariable mit zufälligem Parameter des Freiheitsgrades dargestellt werden kann.

Wenn N eine Poissonverteilte Zufallsvariable mit Mittelwert $\frac{\lambda}{2}$ ist, dann gilt

$$P(N = j) = e^{-\frac{\lambda}{2}} \frac{(\frac{\lambda}{2})^j}{j!}, \quad j = 0, 1, 2, \dots$$

Betrachtet man nun eine Zufallsvariable $\mathcal{X}_{\nu+2N}^2$ mit N Poisson-verteilt¹. Bedingt auf $N = j$ hat die Zufallsvariable die gewöhnliche \mathcal{X}^2 -Verteilung mit $\nu + 2j$ Freiheitsgraden. Dann gilt:

$$P(\mathcal{X}_{\nu+2N}^2 \leq y | N = j) = \frac{1}{2^{(\frac{\nu}{2})+j} \Gamma((\frac{\nu}{2}) + j)} \int_0^y e^{-\frac{z}{2}} z^{(\frac{\nu}{2})+j-1} dz \quad (4.5.5)$$

und die unbedingte Verteilung ist gegeben durch

$$\sum_{j=0}^{\infty} P(N = j) P(\mathcal{X}_{\nu+2N}^2 \leq y | N = j) = \sum_{j=0}^{\infty} e^{-\frac{\lambda}{2}} \frac{(\frac{\lambda}{2})^j}{j!} P(\mathcal{X}_{\nu+2j}^2 \leq y). \quad (4.5.6)$$

Um einen Pfad des CIR-Modells exakt zu simulieren, kann man auf diese Eigenschaften zurückgreifen. Es gilt für die nichtzentrale Chi-Quadratverteilte Übergangswahrscheinlichkeit die Verteilung (4.5.3). Auf unseren Prozess bezogen gilt:

$$X(t) = \frac{\sigma^2(1 - e^{-\kappa(t-u)})}{4\kappa} \mathcal{X}_d'^2 \left(\frac{4\kappa e^{-\kappa(t-u)}}{\sigma^2(1 - e^{-\kappa(t-u)})} X(u) \right), \quad t > u \quad (4.5.7)$$

mit

$$d = \frac{4\theta\kappa}{\sigma^2}. \quad (4.5.8)$$

Das heißt man kann sagen, dass wenn $X(u)$ bekannt ist, ist $X(t)$ eine $\frac{(\sigma^2(1 - e^{-\kappa(t-u)})}{4\kappa}$ -mal nichtzentrale Chi-Quadratverteilte Zufallsvariable mit d Freiheitsgraden und Nichtzentralitätsparameter

$$\lambda = \frac{4\kappa e^{-\kappa(t-u)}}{\sigma^2(1 - e^{-\kappa(t-u)})} X(u). \quad (4.5.9)$$

Äquivalent dazu ist die Aussage

$$P(X(t) \leq y | X(u)) = F_{\mathcal{X}_d'^2}(\lambda) \left(\frac{4\kappa y}{\sigma^2(1 - e^{-\kappa(t-u)})} \right) \quad (4.5.10)$$

¹Definition Poissionverteilung: Sei $\lambda \in \mathbb{R}_+$. Eine Zufallsvariable X , deren Zielbereich alle natürlichen Zahlen $0, 1, \dots$ umfasst, heißt Poissionverteilt mit Parameter λ , kurz $\text{Pois}(\lambda)$ -verteilt, wenn

$$P(X = k) = e^{-\lambda} \frac{\lambda^k}{k!},$$

mit $k = 0, 1, 2, \dots$ [vgl. [8], p.27].

mit d wie in (4.5.8), λ wie in (4.5.9) und $F_{\mathcal{X}'_d}(\lambda)$ wie in (4.5.3). Die Chi-Quadratverteilung ist ein Spezialfall der Gammaverteilung². Mit diesem Wissen kann man nun die Pfade des CIR-Modells exakt simulieren.

4.6. Simulation eines Pfades des CIR-Modells

[vgl. [5], p.124]

| |
|--|
| <p>1. Fall: $d > 1$ für $i = 0, \dots, n - 1$ definiere</p> $c = \sigma^2(1 - e^{-\kappa(t_{i+1}-t_i)})/4\kappa$ $\lambda = X(t_i)(e^{-\kappa(t_{i+1}-t_i)})/c$ <p>generiere</p> $Z \sim N(0, 1)$ $Y \sim \mathcal{X}_{d-1}^2$ <p>setze</p> $X(t_{i+1}) = c[(Z + \sqrt{\lambda})^2 + Y]$ <p>ende</p> |
| <p>2. Fall: $d \leq 1$ für $i = 0, \dots, n - 1$ definiere</p> $c = \sigma^2(1 - e^{-\kappa(t_{i+1}-t_i)})/4\kappa$ $\lambda = X(t_i)(e^{-\kappa(t_{i+1}-t_i)})/c$ <p>generiere</p> $N \sim Pois\left(\frac{\lambda}{2}\right)$ $Y \sim \mathcal{X}_{d+2N}^2$ <p>setze</p> $X(t_{i+1}) = cY$ <p>ende</p> |

²Definition Gammaverteilung: Die Gammaverteilung ist durch die folgende Wahrscheinlichkeitsdichte gegeben:

$$f(y) = \frac{1}{\Gamma(a)\beta^a} y^{a-1} e^{-\frac{y}{\beta}},$$

mit $y \geq 0$ und reellen Parametern a und β [vgl. [5]].

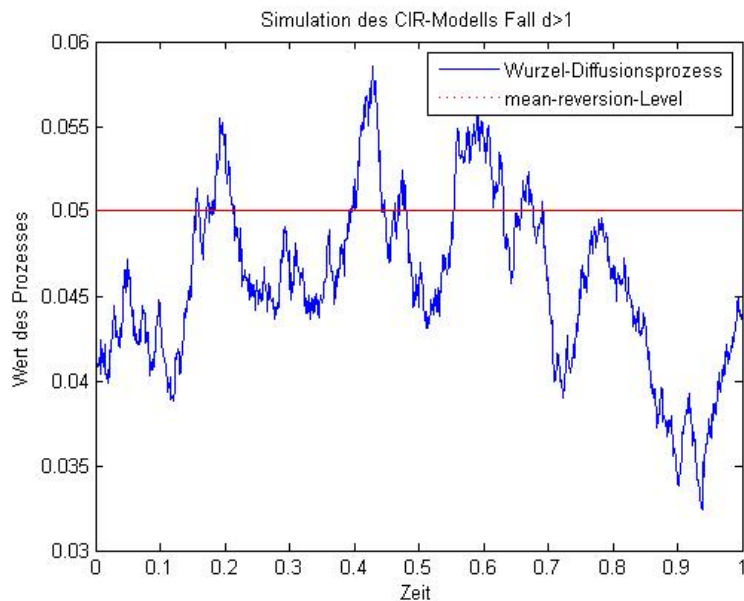


Abbildung 7: Pfad des CIR-Modells für $d > 1$

Abbildung (7) zeigt einen Pfad des CIR-Modells für den Fall $d > 1$. Es wurde mittels der in (4.6) beschriebenen Simulation (links) erstellt. Abbildung (8) zeigt dementsprechend einen Pfad, der mittels der in (4.6) beschriebenen Simulation (rechts) erstellt wurde. In beiden Abbildungen ist zusätzlich noch das Mean-reversion-Level mit eingezeichnet.

4.7. Diskretisierungsverfahren für das CIR-Modell

Die eben erläuterte Methode zur exakten Simulation des CIR-Modells ist numerisch sehr aufwendig, da man für jeden Schritt eine normalverteilte und eine gammaverteilte Zufallsvariable ziehen muss. Dies ist für zum Beispiel Monte-Carlo-Simulationen³ aus diesem Grund eher ungeeignet. Eine andere Möglichkeit, zwar nicht exakt, jedoch schneller ist die Anwendung von den in Kapitel 3 kennengelernten Diskretisierungsmethoden, wie beispielsweise das einfache Euler-Maruyama-Verfahren. Hier ist jedoch zu beachten, dass jedes $X(t_n)$, welches durch beispielsweise das Euler-Maruyama-Verfahren erzeugt wurde, mit positiver Wahrscheinlichkeit negative Werte annehmen kann. Diskretisierungsverfahren, die die Struktur des CIR-Modells erhalten, also die die Positivität des Prozesses garantieren, werden in diesem Kapitel der Arbeit vorgestellt. Die vorgestellten Diskretisierungsverfahren wurden aus [1], [2], [3], [7] und [12] entnommen.

³Die Monte Carlo Simulation ist heutzutage ein wichtiges Instrument bei der Preisgestaltung von Wertpapieren [vgl. [5]]. Für die Anwendung der Monte Carlo Simulation wird ein Integrand an zufällig gewählten Stützstellen ausgewertet. Der Integralwert wird als Mittel der Funktionswerte an diesen Stützstellen berechnet [vgl. [4]].

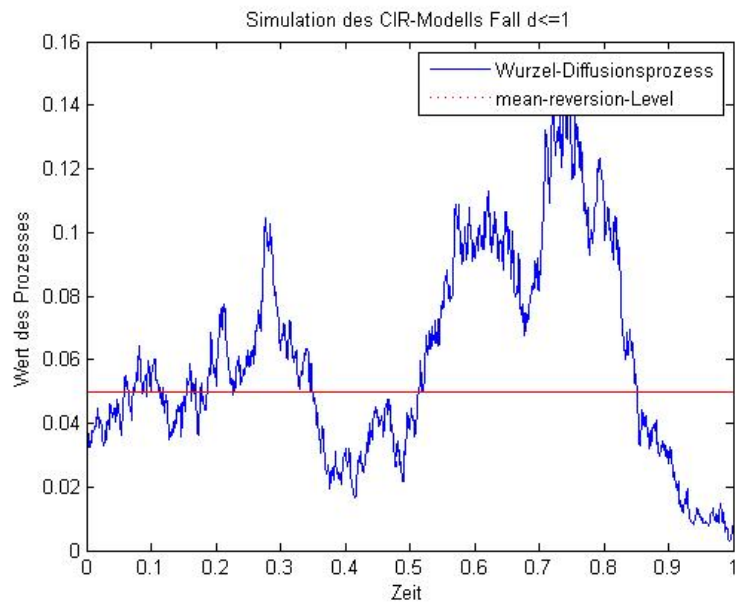


Abbildung 8: Pfad des CIR-Modells für $d \leq 1$

4.7.1. Positivität

Definition

[vgl. [6]]

Es sei X_t ein positiver stochastischer Prozess, das heißt es gilt für alle $t > 0$,

$$P(\{X_t > 0\}) = 1.$$

Unter dieser Bedingung nennen wir das numerische Diskretisierungsverfahren mit der Approximation $X(n)$ für $X(t_n)$ positivitätserhaltend genau dann, wenn

$$P(\{X(n+1) > 0 | X(n) > 0\}) = 1$$

gilt für alle $n \in \mathbb{N}$.

4.7.2. Euler-Maruyama-Verfahren

Wir beginnen wieder mit dem einfachsten Verfahren, dem Euler-Maruyama-Verfahren. Das Euler-Maruyama-Verfahren haben wir folgendermaßen definiert (siehe Kapitel 3)

$$X(n+1) = X(n) + a(X(n))\Delta_n + b(X(n))\Delta W(n),$$

$$n = 0, 1, 2, \dots, N-1$$

mit $\Delta_n = t_{n+1} - t_n = \int_{t_n}^{t_{n+1}} 1 \cdot ds$ und $\Delta W_n = W(t_{n+1}) - W(t_n) = \int_{t_n}^{t_{n+1}} 1 \cdot dW(s)$. Daraus ergibt sich für das CIR-Modell

$$X(n+1) = X(n) + \kappa(\theta - X(n))\Delta_n + \sigma\sqrt{X(n)}\Delta W(n),$$

$$n = 0, 1, 2, \dots, N-1.$$

Dieses Verfahren liefert an den Zeitpunkten $t_{n+1} = t_0 + (n+1)\Delta_n$, die für

$$\Delta W < -\frac{X_{t_n} + \kappa(\theta - X_{t_n})\Delta_n}{\sigma\sqrt{X_{t_n}}}$$

trotz $X_{t_n} > 0$ eine negative Approximation für $X_{t_{n+1}}$ [vgl. [6], p.237]. In diesem Fall würde das Verfahren scheitern. Das Euler-Maruyama-Verfahren ist also nicht positivitaterhaltend.

Wir wollen uns nun Verfahren anschauen, mit denen man das CIR-Modell approximieren kann, ohne dass das Verfahren auf Grund von negativen Werten scheitert.

Euler-Verfahren nach Higham

Higham hat folgende Losung fur das Problem der Strukturhaltung vorgeschlagen. Er hat $\sqrt{X(t)}$ durch $\sqrt{|X(t)|}$ ersetzt. Daraus erhalt man folgendes Verfahren

$$X(n+1) = X(n) + \kappa(\theta - X(n))\Delta_n + \sigma\sqrt{|X(n)|}\Delta W(n). \quad (4.7.1)$$

Dieses Verfahren ist nicht positivitaterhaltend fur Wurzel-Diffusionsgleichungen, wie das CIR-Modell [vgl. [12]].

Euler-Verfahren nach Deelstra und Delbaen

Deelstra und Delbaen haben eine andere Moglichkeit vorgeschlagen, die Probleme zu verhindern. Sie haben folgendes Verfahren untersucht:

$$X(n+1) = X(n) + \kappa(\theta - X(n)^+)\Delta_n + \sigma\sqrt{X(n)^+}\Delta W(n), \quad (4.7.2)$$

mit $X^+ = \max(x, 0)$. Die Besonderheit dieses Prozesses ist, dass der Prozess kleiner Null werden kann, an den Punkten, an denen das Verfahren deterministisch wird mit einer Aufwartsbewegung $\kappa\theta$. Das bedeutet, man kann die Positivitat der Diskretisierung nicht garantieren.

Euler-Maruyama-Verfahren nach Diop

Diop betrachtete die folgende Losung fur das Problem:

$$X(n+1) = |X(n) + \kappa(\theta - X(n))\Delta_n + \sigma\sqrt{X(n)}\Delta W(n)|. \quad (4.7.3)$$

Dieses Verfahren spiegelt die Losung an der Achse $X = 0$. Das Euler-Verfahren nach Diop ist positivitaterhaltend fur CIR-Prozesse.

Diese Methoden liefern zwar positive Prozesse, jedoch ist es zweifelhaft, ob diese Verfahren zulässige Realisierungen eines Pfades von (4.0.1) erzeugen [vgl. [6]]. Wir können durch implizite Verfahren die Positivität eines Prozesses erhalten. In den nächsten Schritten wollen wir diese impliziten Verfahren vorstellen.

4.7.3. Milstein-Verfahren

Beim Milstein-Verfahren gilt analog zum Euler-Maruyama-Verfahren, dass für

$$\Delta W < \min \left\{ -\frac{X_{t_n} + \kappa(\theta - X_{t_n})}{\sigma\sqrt{X_{t_n}}}, \sqrt{\Delta_n} \right\}$$

das Verfahren trotz positivem Anfangswert $X_{t_n} > 0$ eine negative Approximation für $X_{t_{n+1}}$ liefert [vgl. [6], p.238]. Man muss also wieder das Verfahren so ändern, dass man die Positivität des CIR-Modells gewährleisten kann. Diese Positivität kann man mit dem drift-impliziten Milstein-Verfahren garantieren.

Drift-implizites Milstein-Verfahren

Das drift-implizite Milstein-Verfahren hat die folgende Form

$$X(n+1) = X(n) + a(X(n+1))\Delta_n + b(X(n))\Delta W + \frac{1}{4}b'(X(n))b(X(n))(\Delta W^2 - \Delta_n). \quad (4.7.4)$$

Anhand dieser Gleichung ist zu erkennen, dass sich die drift-implizite Variante nur in der impliziten Behandlung des Driftterms a von dem normalen Milstein-Verfahren unterscheidet.

4.7.4. Satz [Drift-implizites Milstein-Verfahren & Positivität]

[vgl [6], pp.237-238]

Das drift-implizite Milstein-Verfahren (4.7.4), angewandt auf das CIR-Modell (4.0.1), ist positivitätserhaltend für beliebige Schrittweiten Δ_n .

Beweis

Für das CIR-Modell (4.0.1) ist ein Diskretisierungsschritt mittels dem drift-impliziten Milstein-Verfahren (4.7.4) gegeben durch

$$X(n+1) = X(n) + \kappa(\theta - X(n+1))\Delta_n + \sigma\Delta W + \frac{1}{4}\sigma^2(\Delta W^2 - \Delta_n). \quad (4.7.5)$$

Wenn wir diese Gleichung nach $X(n+1)$ auflösen erhalten wir

$$X(n+1) = \frac{X(n) + \kappa\theta\Delta_n + \sigma\sqrt{X(n)}\Delta W + \frac{1}{4}\sigma^2((\Delta W)^2 - \Delta_n)}{1 + \kappa\Delta_n}. \quad (4.7.6)$$

Es muss also nur gezeigt werden, dass der Zähler stets positiv ist. Wir fassen alle Zufallsterme in eine Funktion zusammen

$$g(\Delta W) = \sigma\sqrt{X(n)}\Delta W + \frac{1}{4}\sigma^2\Delta W^2 \quad (4.7.7)$$

daraus erhalten wir mit

$$g'(\Delta W) = \sigma\sqrt{X(n)} + \frac{1}{2}\sigma^2\Delta W \quad (4.7.8)$$

sofort, dass

$$\min_{\Delta W \in \mathbb{R}} g(\Delta W) = g\left(-\frac{\sigma\sqrt{X(n)}}{\sigma}\right) = -X(n). \quad (4.7.9)$$

Jetzt können wir damit den Zähler wie folgt abschätzen

$$\begin{aligned} & X(n) + (\kappa\theta - \frac{1}{4}\sigma^2)\Delta_n + g(\Delta W) \\ & \geq X(n) + (\kappa\theta - \frac{1}{4}\sigma^2)\Delta_n + \min_{\Delta W \in \mathbb{R}} g(\Delta W) \\ & = X(n) + (\kappa\theta - \frac{1}{4}\sigma^2)\Delta_n - X(n) \\ & = (\kappa\theta - \frac{1}{4}\sigma^2)\Delta_n > 0. \end{aligned}$$

Die letzte Ungleichung folgt aus der Stabilitätsbedingung $2\kappa\theta > \sigma^2$. Gilt die Stabilitätsbedingung nicht, ist es sinnvoll an den Stellen, an denen $X(t)$ negativ wird, auf das Verfahren (4.7.2) zurückzugreifen.

4.7.5. Taylor Verfahren schwacher Ordnung 2

[vgl. [5], pp.355-357]

Um ein Verfahren mit schwacher Ordnung 2 für das CIR-Modell zu verwenden, benutzt Glasserman das vereinfachte Taylor-Verfahren

$$\begin{aligned} X(t_{n+1}) &= (t_n) + a(x, t)\Delta_n + b(X, t)\Delta W_n \\ &+ \frac{1}{2}(a'b + ab' + \frac{1}{2}b^2b'')\Delta W_n\Delta_n + \frac{1}{2}bb'[\Delta W_n^2 - \Delta_n] \\ &+ (aa' + \frac{1}{2}b^2a'')\frac{1}{2}\Delta_n^2. \end{aligned} \quad (4.7.10)$$

Um zu vermeiden, dass man die Wurzel aus einer negativen Zahl ziehen oder durch Null teilen muss, ersetzt er $X(t_n)$ durch $\hat{X}(t_n) = |X(t_n)|$. Daraus ergibt sich das folgende vereinfachte Taylor Verfahren 2. Ordnung für Wurzel-Diffusionsprozesse

$$X(t_{n+1}) = \hat{X}(t_n) + \kappa\theta\Delta_n - \kappa\hat{X}(t_n)\Delta_n + \sigma\sqrt{\hat{X}(t_n)}\Delta W_n$$

$$\begin{aligned}
& + \left(\left[\frac{\kappa\theta}{4} - \frac{\sigma^2}{16} \right] \frac{\sigma}{\sqrt{\hat{X}(t_n)}} - \frac{3\kappa\sigma}{2} \sqrt{\hat{X}(t_n)} \right) \Delta W_n \Delta_n \\
& - \frac{1}{2} \kappa^2 (\theta - \hat{X}(t_n)) \Delta_n^2. \tag{4.7.11}
\end{aligned}$$

Dieses Verfahren hat nach Glasserman eine schwache Konvergenzordnung 2. Diese wird im nächsten Kapitel nachgeprüft.

5. Numerische Ergebnisse

5.1. Starke Konvergenz

In diesem Teil der Bachelorarbeit wollen wir uns mit der starken Konvergenz der Verfahren nach Deelstra und Delbaen, nach Diop und dem Drift-impliziten Milstein-Verfahren beschäftigen. Dies wollen wir in numerischer Weise machen. Wir bedienen uns hierfür der in Kapitel 3.5 vorgestellten Methode der Ausgleichsrechnung (Regressionsanalyse) und zusätzlich dem Verfahren der kleinsten Quadrate. Da wir keine geschlossene Lösungsformel für das CIR-Modell zur Verfügung haben, mit der wir den Fehler der Verfahren bestimmen können, müssen wir auf folgendes Lemma zurückgreifen.

Lemma

[vgl. [1], p. 374]

Betrachten wir ein Verfahren X_n , das gegen den kontinuierlichen Prozess X_{t_n} folgendermaßen konvergiert

$$E[\max_n \|X_n^{(\Delta)} - X_{t_n}\|].$$

Dann gilt:

$$E[\max_n \|X_n^{(\Delta)} - X_{t_n}\|] \Leftrightarrow E[\max_n \|X_n^{(\Delta)} - X_n^{(2\Delta)}\|]$$

Beweis

Für den Beweis wird auf [1], p.374 verwiesen.

Aus diesem Lemma können wir folgern, dass wir unser Verfahren auf starke Konvergenz testen können, indem wir unser Verfahren erst mit einer kleinen Schrittweite implementieren und danach noch einmal dasselbe Verfahren mit einer größeren Schrittweite nutzen und daraus unseren Fehler bestimmen. Deelstra & Delbaen sowie Diop haben ihre Verfahren bereits auf starke Konvergenz getestet. Auch von dem Drift-impliziten Milstein-Verfahren wissen wir, dass es im starken Sinne konvergent ist [vgl. [1]]. Das bedeutet, wir können das Lemma auf unsere Verfahren ohne Probleme anwenden. Für die numerischen Tests wählen wir das Standardzeitintervall $[0, 1]$, das heißt wir setzen $T = 1$ und berechnen damit unseren Fehler

$$\epsilon(\Delta) = E[\max_n \|X_n^{(\Delta)} - X_n^{(\Sigma)}\|]$$

, mit $\Sigma > \Delta$, für verschiedene Parameter κ, θ und σ . Im letzten Kapitel haben wir das Verfahren nach Deelstra und Delbaen vorgestellt. Dieses Verfahren ist nicht positivitaterhaltend. Diese Eigenschaft kann man in Abbildung (9) sehen. Man kann beim Vergleichen der beiden empirischen Verteilungen erkennen, dass der CIR-Prozess simuliert mittels dem Euler-Verfahren nach Deelstra und Delbaen zu viele Werte nahe der Null oder sogar kleiner Null annimmt [vgl.[5], pp.124-125]. Abbildung (10) und Abbildung (11) zeigen die Konvergenz des Fehlers $\epsilon(\Delta)$ in einer Funktion mit Schrittweite h für verschiedene Parameter. Man kann erkennen, dass das Drift-implizite Milstein-Verfahren einen kleineren Fehler gibt, als die beiden anderen Verfahren. Das gilt für alle Parameter, die getestet wurden. Man kann erkennen, dass die Konvergenz des Drift-impliziten Milstein-Verfahrens ohne die Bedingung der Stabilitatsbedingung langsamer wird.

Um eine Idee von der Konvergenzgeschwindigkeit zu bekommen, wenden wir nun die lineare Ausgleichsrechnung an. Dazu plotten wir unseren Fehler jetzt in einer doppelt-logarithmischen Skala.

Wir vermuten eine starke Konvergenzordnung von $\frac{1}{2}$ für das Euler-Verfahren nach Deelstra und Delbaen, sowie für das Euler-Verfahren nach Diop. Für das Drift-implizite Milstein-Verfahren erwarten wir eine starke Konvergenzordnung von 1, falls die Stabilitatsbedingung erfüllt ist und sonst eine starke Konvergenz von $\frac{1}{2}$. In Abbildung (12) se-

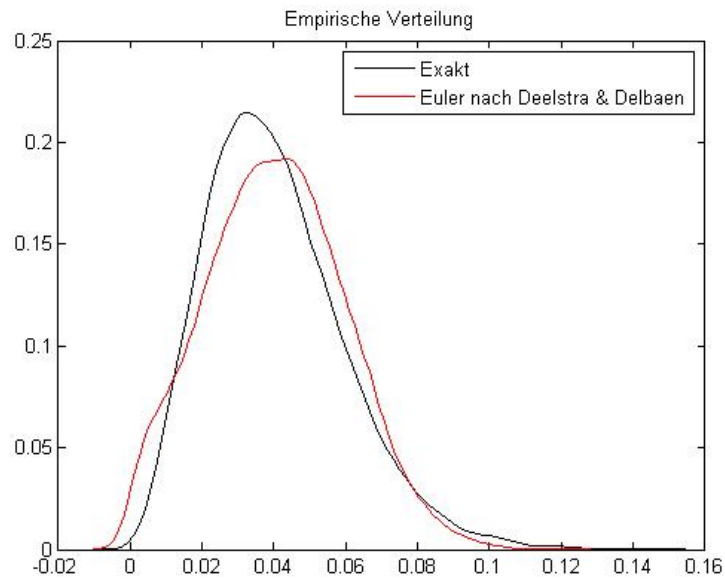


Abbildung 9: Empirische Verteilung des CIR-Prozesses mittels der exakten Simulation und der Simulation durch das Euler-Verfahren nach Deelstra und Delbaen (Euler-DD) mit $\kappa = 0.3$, $\theta = 0.04$, $\sigma = 0.2$, $X_0 = 0.04$ und $T = 0.25$

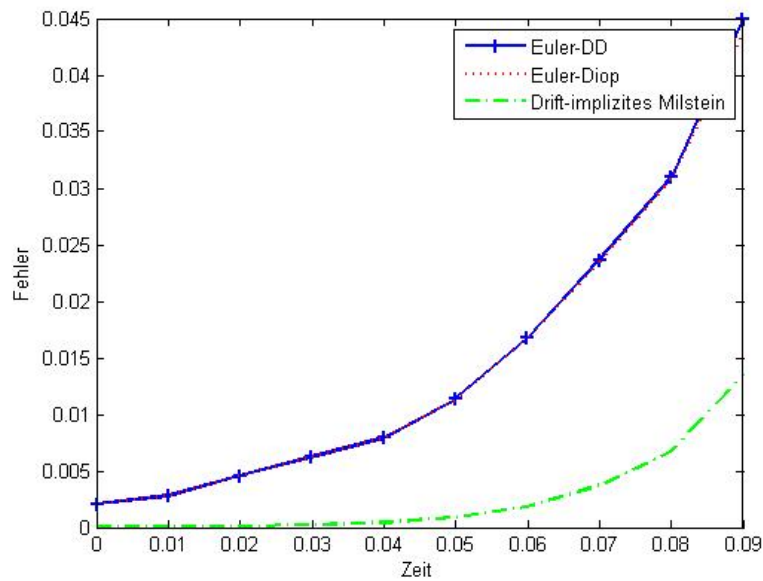


Abbildung 10: Fehler $\epsilon(\Delta)$ als eine Funktion mit $\kappa = 1$, $\theta = 1$, $X_0 = 0.01$ und $\sigma = 1$

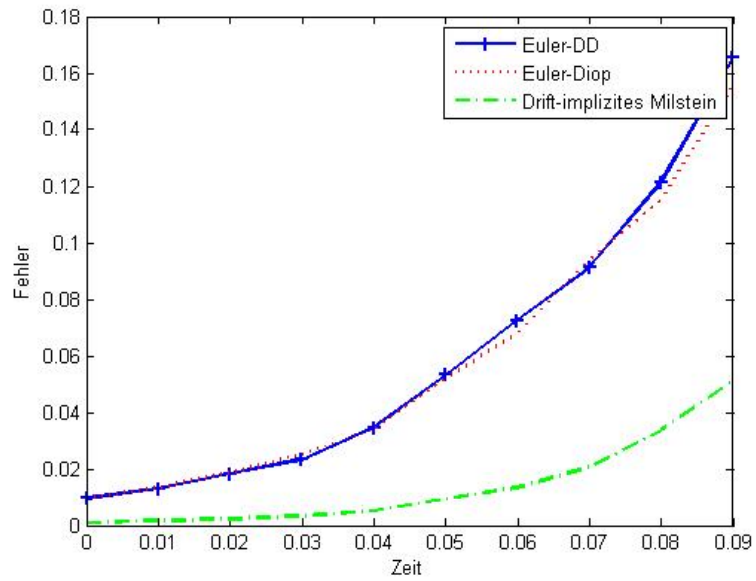


Abbildung 11: Fehler $\epsilon(\Delta)$ als eine Funktion mit $\kappa = 1$, $\theta = 1$, $X_0 = 0.01$ und $\sigma = \sqrt{3}$

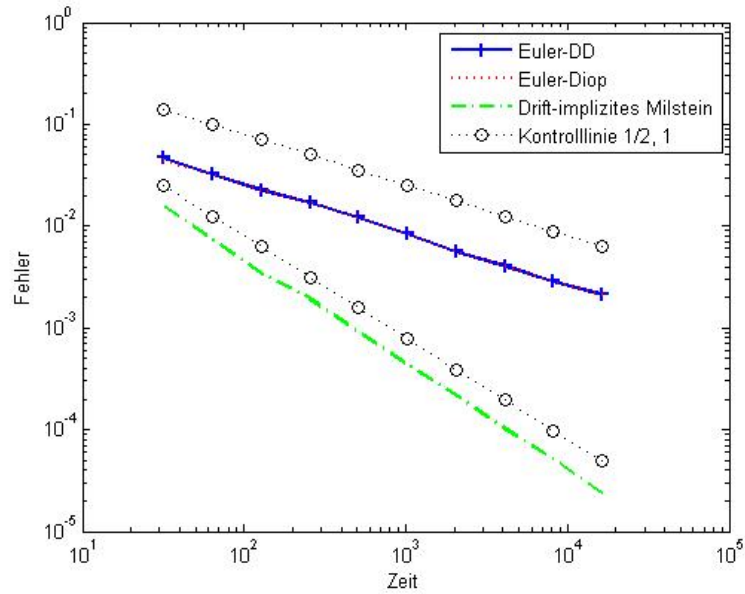


Abbildung 12: Regressionsgeraden für die starke Konvergenz mit $\kappa = 1$, $\theta = 1$, $X_0 = 0.01$ und $\sigma = 1$

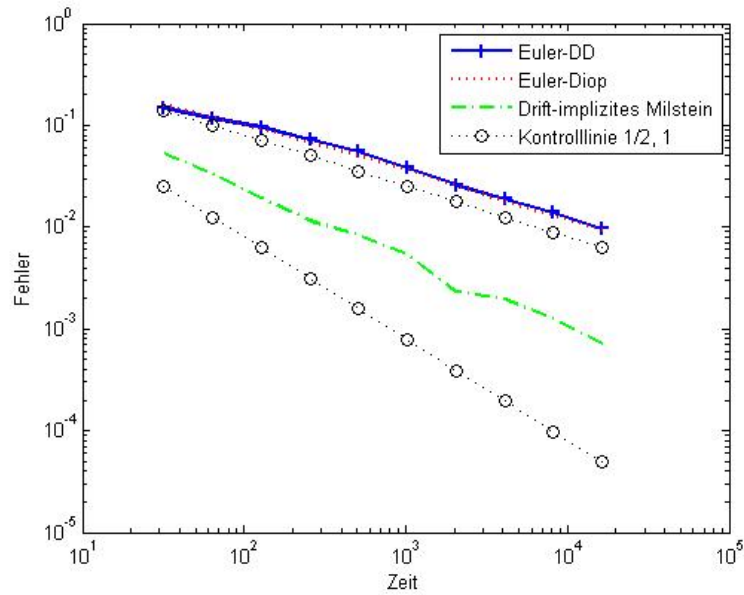


Abbildung 13: Regressionsgeraden für die starke Konvergenz mit $\kappa = 1$, $\theta = 1$, $X_0 = 0.01$ und $\sigma = \sqrt{3}$

hen wir den doppelt-logarithmischen Plot zu Abbildung (10) für 200 Runden. Man kann erkennen, dass die Regressionsgerade der Euler-Methode nach Deelstra und Delbaen eine Steigung von $\approx \frac{1}{2}$ hat. Dasselbe gilt für die Regressionsgerade des Euler-Verfahrens nach Diop. Das Drift-implizite Milstein-Verfahren zeigt eine Regressionsgerade mit Steigung ≈ 1 . Tatsächlich ergab die lineare Ausgleichsrechnung für $\kappa = 1$, $\theta = 1$, $\sigma = 1$ bei 200 Runden:

| Verfahren | γ | Residuum |
|---------------------------|----------|----------|
| Euler-Verfahren DD | 0.4981 | 0.0973 |
| Euler-Verfahren Diop | 0.4953 | 0.1006 |
| drift-implizites Milstein | 1.0323 | 0.1138 |

Für die zweite Simulation ist die Stabilitätsbedingung nicht erfüllt. Es gilt:

$$2\kappa\theta - \sigma^2 = 2 * 1 * 1 - \sqrt{3}^2 = -1.$$

Wir erwarten, dass die Konvergenz des Drift-impliziten Milstein-Verfahrens deutlich langsamer wird. Wir sehen in Abbildung (13) die zugehörigen Regressionsgeraden für 200 Runden. Die lineare Ausgleichsrechnung ergab für $\kappa = 1$, $\theta = 1$ und $\sigma = \sqrt{3}$ bei 200 Runden:

| Verfahren | γ | Residuum |
|---------------------------|----------|----------|
| Euler-Verfahren DD | 0.4489 | 0.2342 |
| Euler-Verfahren Diop | 0.4578 | 0.1280 |
| drift-implizites Milstein | 0.6842 | 0.3195 |

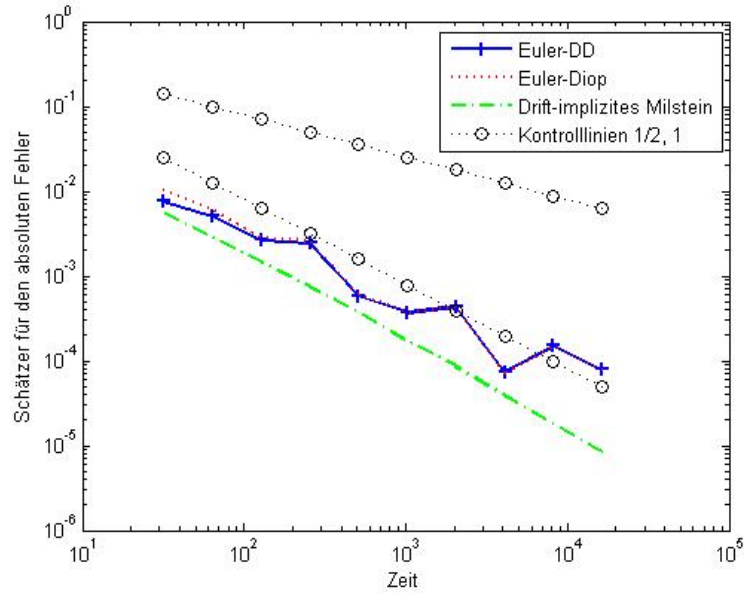


Abbildung 14: Regressionsgeraden für die schwache Konvergenz mit $\kappa = 1$, $\theta = 1$, $X_0 = 0.01$ und $\sigma = 1$ für 500 Runden

Wir lagen also mit unserer Vermutung richtig. Das Drift-implizite Milstein-Verfahren konvergiert ohne Stabilitätsbedingung deutlich langsamer.

5.2. Schwache Konvergenz

Als Nächstes werden die Verfahren auf schwache Konvergenz getestet. Wir wissen aus [1], dass unsere Verfahren, unter der Stabilitätsbedingung, im schwachen Sinne konvergent sind. Wir wenden wieder die lineare Ausgleichsrechnung an. Es ist nur das Fehlerkriterium in unserer Implementierung zu ändern. Für unseren Fehler haben wir nun

$$\max_n \|E(P(X_n^{(\Delta)})) - E(P(X_{t_n}))\| \leq K_{P,T} \Delta^\beta$$

wobei die Konstante hier von T und von P abhängt. Sie gilt für alle Polynome P (oder von einer anderen geeigneten Klasse von Testfunktionen). Als Testfunktion wählen wir die Identität, das heißt $P(X_n) = X_n$. Ansonsten gehen wir analog zur starken Konvergenz vor. In Abbildung (14) sehen wir die Regressionsgeraden der Verfahren für die schwache Konvergenz bei 500 Runden. Man kann sehen, dass alle drei Verfahren näherungsweise eine schwache Konvergenz von 1 haben. Die lineare Ausgleichsrechnung ergibt für $\kappa = 1$, $\theta = 1$ und $\sigma = 1$ bei 500 Runden:

| Verfahren | γ | Residuum |
|---------------------------|----------|----------|
| Euler-Verfahren DD | 0.7810 | 1.2708 |
| Euler-Verfahren Diop | 0.8242 | 1.3137 |
| drift-implizites Milstein | 1.0424 | 0.2118 |

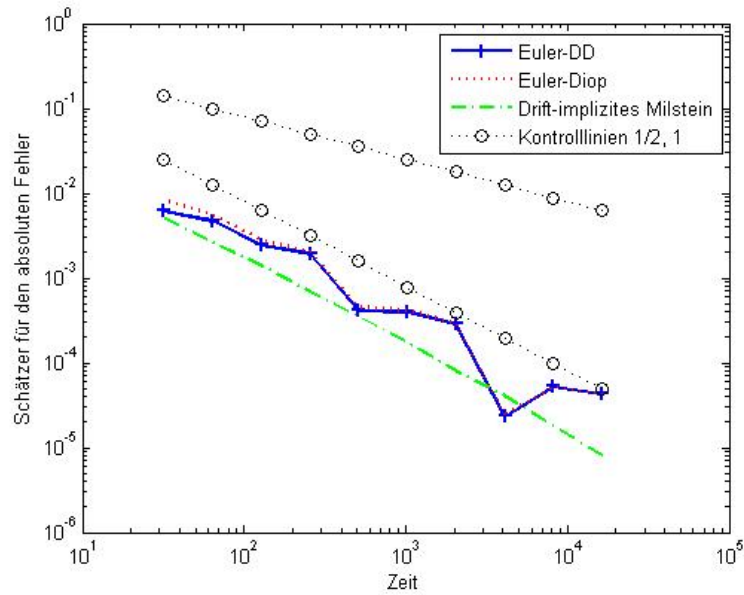


Abbildung 15: Regressionsgeraden für die schwache Konvergenz mit $\kappa = 1$, $\theta = 1$, $X_0 = 0.01$ und $\sigma = 1$ für 1000 Runden

In Abbildung (15) sehen wir den selben Plot wie in Abbildung (14) jedoch mit der doppelten Anzahl an Runden. Die lineare Ausgleichsrechnung liefert hier:

| Verfahren | γ | Residuum |
|---------------------------|----------|----------|
| Euler-Verfahren DD | 0.9198 | 1.7242 |
| Euler-Verfahren Diop | 0.9595 | 1.6551 |
| drift-implizites Milstein | 1.0293 | 0.2132 |

Das Euler-Verfahren nach Diop hat eine schnellere schwache Konvergenz als das Verfahren nach Deelstra und Delbaen. Die schnellste schwache Konvergenz hat jedoch das Drift-implizite Milstein-Verfahren.

Wir fassen nun unsere Ergebnisse in einer Tabelle zusammen:

| Verfahren | starke Konv. | schwache Konv. | Positivität |
|---------------------------|-----------------------|----------------|--|
| Euler-Verfahren DD | $\approx \frac{1}{2}$ | ≈ 1 | Nein |
| Euler-Verfahren Diop | $\approx \frac{1}{2}$ | ≈ 1 | Ja, für alle $\kappa, \theta, \sigma \geq 0$ |
| drift-implizites Milstein | ≈ 1 | ≈ 1 | Ja, für $2\kappa\theta \geq \sigma^2$ |
| drift-implizites Milstein | $\approx \frac{1}{2}$ | ≈ 1 | Nein, für $2\kappa\theta < \sigma^2$ |

Im nächsten Schritt wollen wir numerisch zeigen, dass unsere Standard-Diskretisierungs-

verfahren nicht strukturerhaltend für Wurzel-Diffusionsgleichungen sind. Dazu verwenden wir das Drift-implizite Milstein-Verfahren als Referenzlösung. Wir haben ja bereits die starke Konvergenzordnung dieses Verfahrens gesehen, aus diesem Grund können wir das Verfahren ohne Probleme als Referenzlösung verwenden.

Um zu zeigen, dass unsere Standard-Diskretisierungsverfahren für Wurzel-Diffusionsgleichungen ungeeignet sind verwenden wir den Matlabcode (10). Wir erhalten dabei die folgende Fehlermeldung:

“negativer Wert für $X(n)$, Verfahren scheitert”.

Damit folgt, dass durch die Simulation der Pfade mittels der Standard-Diskretisierungsverfahren negative Werte für das CIR-Modell geliefert werden und diese die Struktur von Wurzel-Diffusionsgleichungen nicht erhalten.

6. Schlussbetrachtung

In dieser Arbeit wurden strukturerhaltende stochastische Approximationen für Wurzel-Diffusionsgleichungen vorgestellt, ihre Konvergenzordnung numerisch bestimmt und untereinander verglichen.

Um diese Diskretisierungsverfahren zu bekommen, wurden erst die Standard-Diskretisierungsverfahren, wie das Euler-Maruyama-Verfahren und das Milstein-Verfahren, hergeleitet.

Es wurde die lineare Ausgleichsrechnung erläutert, um numerisch die Konvergenzordnung bestimmen zu können. Um eine Vorstellung von der Konvergenzordnung zu bekommen, wurden die Standard-Diskretisierungsverfahren getestet. Dies ergibt für das Euler-Maruyama-Verfahren eine starke Konvergenz von $\approx \frac{1}{2}$ und eine schwache Konvergenz von ≈ 1 . Das Milstein-Verfahren hat eine starke sowie eine schwache Konvergenzordnung von ≈ 1 .

Das CIR-Modell wurde betrachtet und festgestellt, dass es keine geschlossene Lösungsformel für dieses Modell gibt. Durch die nichtzentrale Chi-Quadrat-verteilte Übergangswahrscheinlichkeit kann man dieses Modell jedoch exakt simulieren. Dies ist jedoch numerisch sehr aufwendig, da man in jedem Schritt eine normalverteilte und eine gammaverteilte Zufallsvariable simulieren muss. Aus diesem Grund ist es sinnvoll, sich Diskretisierungsverfahren für Wurzel-Diffusionsgleichungen anzuschauen. Diese sind zwar nicht exakt, jedoch schneller zu simulieren.

Für Wurzel-Diffusionsprozesse sind stochastische Approximationen, die die Positivität des Prozesses garantieren, strukturerhaltend. Grund dafür ist die Wurzel im Diffusionsterm.

Bezogen auf das CIR-Modell wird gezeigt, dass unsere Standard-Diskretisierungsverfahren nicht positivitätserhaltend sind. Eine Implementierung dieser Verfahren, angewendet auf das CIR-Modell, scheitert. Um dies zu verhindern, gibt es verschiedene Abänderungen der Standard-Diskretisierungsverfahren für Wurzel-Diffusionsgleichungen.

Das Euler-Verfahren nach Deelstra und Delbaen verhindert das Ziehen einer Wurzel aus

einer negativen Zahl durch das Ersetzen des Diffusionsterms des CIR-Modells durch $\sigma\sqrt{X(n)^+}$ mit $X^+ = \max(x, 0)$. Dieses Verfahren ist jedoch nicht positivitätserhaltend für das CIR-Modell, da der Prozess an den Punkten, an denen das Verfahren deterministisch wird, kleiner Null werden kann. Das Verfahren liefert eine starke Konvergenzordnung von $\approx \frac{1}{2}$, sowie eine schwache Konvergenzordnung von ≈ 1 . Eine Verbesserung zu diesem Verfahren ist das Euler-Verfahren nach Diop. Diop wählt für sein Verfahren den Absolutbetrag des Standard Euler-Verfahrens. Das heißt, das Verfahren spiegelt die negativen Werte des Prozesses an der x-Achse. Dieses Verfahren liefert auch eine starke Konvergenzordnung von $\approx \frac{1}{2}$ und eine schwache Konvergenzordnung von ≈ 1 . Um die starke Konvergenz zu verbessern, kann man das Drift-implizite Milstein-Verfahren verwenden. Dieses Verfahren ist unter der Stabilitätsbedingung positivitätserhaltend für das CIR-Modell. Dann hat das Verfahren eine starke und schwache Konvergenzordnung von ≈ 1 . Gilt jedoch die Stabilitätsbedingung nicht, so ist das Drift-implizite Milstein-Verfahren nicht mehr positivitätserhaltend und die starke Konvergenz wird deutlich langsamer. Die starke Konvergenzordnung verringert sich von ≈ 1 auf $\approx \frac{1}{2}$. In diesem Fall ist es sinnvoll, auf das Euler-Verfahren nach Diop zurückzugreifen.

Im numerischen Teil dieser Arbeit wurde die Konvergenzordnung speziell für das CIR-Modell untersucht. Alle Implementierungen wurden mit Matlab erstellt. Die Programmcodes befinden sich im Anhang.

A. Programmcodes

Listing 1: Code zur Simulation eines Wiener-Prozesses

```
1 %Programm zur Simulation eines Pfades des Wiener Prozesses
2 %Dieser Code wurde verwendet, um die Figure 1 zu erzeugen
3 randn('state',5)
4 N=2^16;
5 t = (0:1:N)'/N;
6 W = [0; cumsum(randn(N,1))]/sqrt(N);
7 plot(t,W);
8 hold on
9 plot(t,0*t,':r')
10 axis([0 1 -2 2])
11 title('Pfad eines Wiener-Prozesses mit dem Mittelwert')
12 legend('Wiener-Prozess', 'Mittelwert')
13 xlabel('Zeit')
14 ylabel('Wert des Wiener-Prozesses')
15 hold off
```

Die nächsten beiden Codes sind die wichtigsten dieser Arbeit. Sie sind universell einsetzbar. Man kann mit ihnen beliebige stochastische Differentialgleichungen auf ihre Konvergenzordnungen überprüfen. Man muss lediglich die Funktionen f und g auf die zu betrachtende stochastische Differentialgleichung anpassen.

Listing 2: Code Test auf starke Konvergenz

```
1 function taylor_sde( X_0, f, g, dg, rounds )
2 %Mit Hilfe dieses Programmes kann man
3 %stochastische Differentialgleichungen
4 %auf starke Konvergenz testen
5 function taylor_sde( X_0, f, g, dg, rounds )
6     %  $dX_t = f(t, X_t) dt + g(t, X_t) dW_t$ 
7     maxM = 2^16;
8     M = fliplr(2.^(5:14));%Anzahl der Schritte
9     error_euler = zeros(numel(M),1);
10    error_milstein = zeros(numel(M),1);
11    for r = 1:rounds
12        %Referenzloesung
13        h = 1/maxM; %Schrittweite der Referenzloesung
14        U = randn(maxM,2);
15        dW = U(:,1)*sqrt(h); %Inkrement des Wiener-Prozesses
16        W = [0; cumsum(dW)];%Wiener-Prozess
17        X_ref = zeros(maxM + 1,1);
18        X_ref(1) = X_0;
19        %Berechnung der Referenzloesung
20        for i=2:maxM+1
21            X_ref(i) = milstein(X_ref(i-1), f, g, dg, dW(i-1), h);
22        end
23        %Berechnung der Loesung mit groeberer Schrittweite
24        for m=1:numel(M)
25            h=1/M(m);
26            factor = maxM/M(m);
```

```

27     X_euler = zeros(M(m) + 1,1);
28     X_euler(1) = X_0;
29     X_milstein = zeros(M(m) + 1,1);
30     X_milstein(1) = X_0;
31     for i=2:M(m)+1
32         dW_n = W( factor*(i-1) + 1 ) - W( factor*(i-2) + 1 );
33         X_euler(i) = euler(X_euler(i-1), f, g, dW_n, h);
34         X_milstein(i) = milstein(X_milstein(i-1), f, g, dg, dW_n, h);
35     end
36     %Berechnung des Fehlers
37     error_euler(m) = error_euler(m) + (X_ref(end)-X_euler(end))^2;
38     error_milstein(m) = error_milstein(m) + (X_ref(end)-X_milstein(end))^2;
39 end
40 end
41 error_euler = sqrt(error_euler./rounds);
42 error_milstein = sqrt(error_milstein./rounds);
43 %Plotten der Ausgleichsgerade mit Kontrolllinien
44 loglog( M, error_euler, '- b', 'Linewidth', 4);
45 hold on
46 loglog( M, error_milstein, '- g', 'Linewidth', 4);
47 loglog( M, 0.8.*M.^-(1/2), ':ok', 'Linewidth', 2);
48 loglog( M, 0.8.*M.^-(1), ':ok', 'Linewidth', 2);
49 legend('Euler', 'Milstein', 'Kontrolllinien 1/2, 1')
50 xlabel('Zeit')
51 ylabel('Schaezter f r den absoluten Fehler')
52 grid
53 %Methode der kleinsten Quadrate
54 dtlist=h*(2.^(0:numel(M)-1));
55 A=[ones(numel(M),1),log(dtlist)'];
56 b=log(error_euler);
57 x=A\b;
58 gamma=x(2)
59 residuum=norm(A*x-b)
60 c=log(error_milstein);
61 y=A\c;
62 sigma=y(2)
63 residuum2=norm(A*y-c)
64 end

```

Listing 3: Code Test auf schwache Konvergenz

```

1 function taylor_sde_weak( X_0, f, df, ddf, g, dg, ddg, rounds )
2 % dX_t = f(t, X_t) dt + g(t, X_t) dW_t
3 %Anzahl der Schritte der Referenzloesung
4 maxM = 2^16;
5 %Anzahl der Schritte
6 M = fliplr(2.^(5:14));
7 error_exact = 0;
8 error_euler = zeros(numel(M),1);
9 error_milstein = zeros(numel(M),1);
10 for r = 1:rounds
11 %Referenzloesung
12 %Schrittweite der Referenzloesung
13 h = 1/maxM;

```

```

14 %Inkremente des Wiener-Prozesses
15     U = randn(maxM,2);
16     dW = U(:,1)*sqrt(h);
17 %Wiener-Prozess
18     W = [0; cumsum(dW)];
19     X_ref = zeros(maxM + 1,1);
20     X_ref(1) = X_0;
21 %Berechnung der Referenzloesung
22     for i=2:maxM+1
23         X_ref(i) = milstein(X_ref(i-1), f, g, dg, dW(i-1), h);
24     end
25     error_exact = error_exact + X_ref(end);
26 %Berechnung der Loesung mit groeberer Schrittweite
27     for m=1:numel(M)
28         h=1/M(m);
29         factor = maxM/M(m);
30         X_euler = zeros(M(m) + 1,1);
31         X_euler(1) = X_0;
32         X_milstein = zeros(M(m) + 1,1);
33         X_milstein(1) = X_0;
34         for i=2:M(m)+1
35             dW_n = W( factor*(i-1) + 1 ) - W( factor*(i-2) + 1 );
36             X_euler(i) = euler(X_euler(i-1), f, g, dW_n, h);
37             X_milstein(i) = milstein(X_milstein(i-1), f, g, dg, dW_n, h);
38         end
39         error_euler(m) = error_euler(m) + X_euler(end);
40         error_milstein(m) = error_milstein(m) + X_milstein(end);
41     end
42 end
43 error_exact = error_exact/rounds;
44 error_euler = abs(error_euler./rounds - error_exact);
45 error_milstein = abs(error_milstein./rounds - error_exact);
46 %Plot
47 loglog( M, error_euler, '- b' , 'Linewidth', 4);
48 hold on
49 loglog( M, error_milstein, '- g' , 'Linewidth', 4);
50 loglog( M, 0.8.*M.^-(1), ':ok' , 'Linewidth', 2);
51 legend( 'Euler', 'Milstein', 'Kontrolllinie 1' )
52 xlabel( 'Zeit' )
53 ylabel( 'Schaezter f r den absoluten Fehler' )
54 grid
55 %Methode der kleinsten Quadrate
56 dtlist = 2.^([1:numel(M)] - 10);
57 A = [ones(numel(M), 1), log(dtlist)'];
58 b = log(error_euler);
59 x = A\b;
60 gamma = x(2)
61 residuum = norm(A*x - b)
62 c = log(error_milstein);
63 y = A\c;
64 sigma = y(2)
65 residuum2 = norm(A*y - c)
66 end

```

Für diese Bachelorarbeit wurden verschiedene Verfahren verwendet. Diese wurden in verschiedene Funktionen implementiert und sollen nun vorgestellt werden:

Listing 4: Code Standard Euler-Verfahren

```

1 function X_n1 = euler( X_n, f, g, dW, h )
3     X_n1 = X_n + f(X_n)*h + g(X_n)*dW;
5 end

```

Listing 5: Code Milstein-Verfahren

```

1 function X_n1 = milstein( X_n, f, g, dg, dW, h )
3     X_n1 = X_n + f(X_n)*h + g(X_n)*dW + 0.5*g(X_n)*dg(X_n)*(dW^2 - h);
5 end

```

Listing 6: Code Euler-Verfahren nach Deelstra und Delbaen

```

1 function X_n1 = euler_dd( X_n, f, g, dW, h )
3     X_n1 = X_n + f_dd(X_n)*h + g_dd(X_n)*dW;
5 end

```

Listing 7: Code Euler-Verfahren nach Diop

```

1 function X_n1 = euler_diop( X_n, f, g, dW, h )
3     X_n1 = abs(X_n + f(X_n)*h + g(X_n)*dW);
5 end

```

Listing 8: Code drift-implizites Milstein-Verfahren

```

1 function X_n1 = milstein_implzit( X_n, g, kappa, theta, sigma, dW, h )
3     X_n1 = (X_n + kappa*theta*h + g(X_n)*dW + (1/4)*sigma^2*(dW^2-h))/(1+kappa*h);
4 end

```

Listing 9: Code exakte Simulation eines CIR-Pfades

```

1 function [X] = cirpath_normalchi(t, kappa, theta, sigma, X0)
2 %Programm zur exakten Simulation eines Pfades
3 %des CIR-Modells mittels einer normalen
4 %Chi-Quadrat-Verteilung
6 dt = diff(t(:));
7 n = length(t);
8 X = [X0; nan*dt];
9 v = sigma^2;

```

```

10 d = 4*kappa*theta/v;
11 e = exp(-kappa*dt);
12 c = v.*(1-e)/(4*kappa);
13 for i = 1:(n-1)
14     l = X(i)*e(i)/c(i);
15     Z= randn;
16     Y=chi2rnd(d-1);
17     X(i+1) = c(i)*[(Z+sqrt(l)).^2+Y];
18 end

20 function [X] = cirpath_pois(t,kappa,theta,sigma,X0)
21 %Programm zur exakten Simulation eines Pfades
22 %des CIR-Modells mittels einer Poisson-Verteilung
23 dt = diff(t(:));
24 n = length(t);
25 X = [X0; nan*dt];
26 v = sigma^2;
27 d = 4*kappa*theta/v;
28 e = exp(-kappa*dt);
29 c = v.*(1-e)/(4*kappa);
30 for i = 1:(n-1)
31     l = X(i)*e(i)/c(i);
32     N= poissrnd(1/2);
33     Y=chi2rnd(d+2*N);
34     X(i+1) = c(i)*Y;
35 end
36 end

```

Listing 10: Code Test der Standard-Diskretisierungsverfahren für das CIR-Programm

```

1 \label{code}
2 function taylor_sde_euler_milstein_wurz( X_0, f, g, dg, rounds )
3     %Anzahl der Schritte der Referenzloesung
4     maxM = 2^16;
5     %Anzahl der Schritte
6     M = fliplr(2.^(5:14));
7     error_euler = zeros(numel(M),1);
8     error_milstein = zeros(numel(M),1);
9     %error_taylor15 = zeros(numel(M),1);
10    for r = 1:rounds
11        %Referenzloesung
12        %Schrittweite der Referenzloesung
13        h = 1/maxM;
14        %Inkrement des Wiener-Prozesses
15        U = randn(maxM,1);
16        dW = U(:,1)*sqrt(h);
17        %Wiener-Prozess
18        W = [0; cumsum(dW)];
19        X_ref = zeros(maxM + 1,1);
20        X_ref(1) = X_0;
21        %Berechnung der Referenzloesung

22        for i=2:maxM+1
23            X_ref(i) = milstein_implzlit(X_ref(i-1), f, g, dW(i-1), h);

```

```

25     end
26
27     %Berechnung der Loesung mit groeoberer Schrittweite
28     for m=1:numel(M)
29         h=1/M(m);
30         factor = maxM/M(m);
31         X_euler = zeros(M(m) + 1,1);
32         X_euler(1) = X_0;
33         X_milstein = zeros(M(m) + 1,1);
34         X_milstein(1) = X_0;
35
36         for i=2:M(m)+1
37             if any(X_euler < 0)
38                 error('negativer Wert fuer X_n, Verfahren scheitert')
39             end
40             if any(X_milstein < 0)
41                 error('negativer Wert fuer X_n, Verfahren scheitert')
42             end
43             dW_n = W( factor*(i-1) + 1 ) - W( factor*(i-2) + 1 );
44             X_euler(i) = euler(X_euler(i-1), f, g, dW_n, h);
45             X_milstein(i) = milstein(X_milstein(i-1), f, g, dg, dW_n, h);
46         end
47
48         error_euler(m) = error_euler(m) + (X_ref(end)-X_euler(end))^2;
49         error_milstein(m) = error_milstein(m) + (X_ref(end)-X_milstein(end))^2;
50     end
51 end
52
53 error_euler = sqrt(error_euler./rounds);
54 error_milstein = sqrt(error_milstein./rounds);
55 %Plot
56 loglog( M, error_euler , '- b' , 'Linewidth', 2);
57 hold on
58 loglog( M, error_milstein , '- g' , 'Linewidth', 2);
59 %loglog( M, error_taylor15 , '- r' , 'Linewidth', 4);
60 loglog( M, 0.8.*M.^-(1/2), ':ok' , 'Linewidth', 1);
61 loglog( M, 0.8.*M.^-(1), ':ok' , 'Linewidth', 1);
62 %loglog( M, 0.8.*M.^-(3/2), ':ok' , 'Linewidth', 2);
63 legend('Euler', 'Milstein', 'Kontrolllinien 1/2, 1')
64 xlabel('Zeit')
65 ylabel('Fehler')
66 grid
67
68 end

```


Literatur

- [1] ALFONSI, AURLIEN: *On the discretization schemes for the CIR (and Bessel squared) processes*. Monte Carlo Methods and Appl., Vol. 11, No.4, pp.355-384 (2005)
- [2] ANDERSEN, LEIF: *Efficient Simulation of the Heston Stochastic Volatility Model*. Working paper series, 2006
- [3] GERING, VERA: *Parameterschätzung stochastischer Prozesse aus der Finanzwelt mittels (Dünnmitter-) Histogramm-Matching-Verfahren*. Diplomarbeit, Köln, 2006
- [4] GERSTNER, THOMAS: *Skript zur Vorlesung: Finanzmathematik (Computational Finance)*. Vorlesungsskript, Frankfurt, 2010
- [5] GLASSERMAN, PAUL: *Monte Carlo Methods in Financial Engineering*. Springer, New York, 2004
- [6] GÜNTHER, MICHAEL: *Finanzderivate mit Matlab: Mathematische Modellierung und numerische Simulation*. Online-Ausgabe Wiesbaden: Vieweg+ Teubner Verlag/Springer Fachmedien Wiesbaden GmbH, Wiesbaden, 2010
- [7] KAHL, C.; GÜNTHER, M.; ROSSBERG, T.: *Structure preserving stochastic integration schemes in interest rate derivate modeling*. Applied Numerical Mathematics 58 (2008) 284-295
- [8] KERSTING, G.; WALKOLBINGER, A.: *Elementare Stochastik*. Birkhäuser, 2007
- [9] KLOEDEN, PETER E.; PLATEN, ECKHARD *Numerical Solution of Stochastic Differential Equations*. Springer, 1995
- [10] KLOEDEN, PETER E.: *Skript zur Vorlesung: Numerik stochastischer Differentialgleichungen*. Vorlesungsskript, Frankfurt, 2010
- [11] LÜTKE SCHELHOWE, BORIS: *Der Ritt auf der Zinskurve. Einsatz eines Zinsstrukturmodells in der barwertigen Zinsbuchsteuerung*. Diplomarbeit, Münster, 2009
- [12] NEUENKIRCH, A.: *Approximation of SDEs under Non-Standard Assumption*. Frankfurt