UNIVERSITÄT ZU LÜBECK
INSTITUT FÜR MATHEMATIK

Numerische Simulation Iterierter Stochastischer Integrale

Numerical Simulation of Iterated Stochastic Integrals

Masterarbeit

im Rahmen des Studiengangs
Mathematik in Medizin und Lebenswissenschaften
der Universität zu Lübeck

Vorgelegt von
Felix Kastner

Ausgegeben und betreut von
Prof. Dr. Andreas Rößler
Institut für Mathematik

Lübeck, den 18.03.2019

Eidesstattliche Erklärung

Ich versichere an Eides statt, die vorliegende Arbeit selbstständig und nur unter Benutzung der angegebenen Quellen und Hilfsmittel angefertigt zu haben.

Lübeck,
18.03.2019

Felix Kastner

Zusammenfassung

In vielen naturwissenschaftlichen Prozessen hängt die Änderung einer Größe von dem aktuellen Wert dieser Größe ab. Man denke zum Beispiel an den radioaktiven Zerfall, bei dem umso mehr Teilchen zerfallen umso mehr anfangs zur Verfügung standen. Diese Prozesse lassen sich mathematisch durch Differentialgleichungen modellieren. Dies funktioniert jedoch nur für deterministische Vorgänge. Möchte man zufällige Ereignisse berücksichtigen, benötigt man stochastische Differentialgleichungen.

Für die effiziente Simulation von Lösungen stochastischer Differentialgleichungen sind unter anderem iterierte stochastische Integrale notwendig. Diese können nur in seltenen Fällen explizit angegeben werden und müssen numerisch approximiert werden. In dieser Arbeit werden die Ansätze von Kloeden, Platen und Wright sowie Wiktorsson implementiert und auf ihre Eigenschaften untersucht. Dazu wird zunächst eine kurze Einführung in stochastische Differentialgleichungen und deren numerische Lösung gegeben.

Abstract

In many scientific processes the change in one quantity depends on the current value of that quantity. A popular example is radioactive decay where the number of decaying particles is proportional to the number of particles left. These processes are mathematically modelled by differential equations. However this only works for deterministic processes. To account for random events stochastic differential equations are necessary.

To simulate solutions of stochastic differential equations efficiently the use of iterated stochastic integrals is essential. These can only in rare cases be calculated explicitly and thus have to be approximated numerically. In this work the approaches by Kloeden, Platen and Wright as well as Wiktorsson are implemented and their properties studied. To this end first a short introduction to stochastic differential equations and their numerical solution will be given.

Inhaltsverzeichnis

Einleitung

Die *Brownsche Bewegung* wurde erstmals 1828 von ihrem Namensgeber Robert Brown beschrieben, der unter dem Mikroskop beobachtete, wie sich Pollen in Wasser bewegten [Bro28]. Allerdings wurden erst im Jahr 1905 durch Albert Einstein die theoretischen Grundlagen für die Erklärung der Brownschen Bewegung geschaffen [Ein05]. Er demonstrierte, wie die molekularkinetische Theorie der Wärme die von Brown beobachtete Bewegung vorhersagt. Zusätzlich leitete er her, dass die Bewegungen der Teilchen einer Normalverteilung folgen.

Das mathematische Pendant zu der physikalischen Beobachtung der Brownschen Bewegung bildet der *Wiener Prozess* aus der Stochastik. Dieser stochastische Prozess mit normalverteilten, unabhängigen Zuwächsen ist benannt nach Norbert Wiener, welcher 1923 seine wahrscheinlichkeitstheoretische Existenz bewiesen hat und die Theorie der stochastischen *Wiener-Integrale* entwickelte [Wie23]. Diese Ideen wurden von Kiyoshi Itô weitergeführt. Er entwickelte 1944 das nach ihm benannte *Itô-Integral*, mit dem nun auch stochastische Integranden behandelt werden konnten [Itô44].

Seit der Definition des stochastischen Integrals von Itô hat sich die Theorie der stochastischen Differentialgleichungen fortlaufend weiterentwickelt und Einzug in zahlreiche Bereiche der modernen Naturwissenschaften gefunden. Besonders in der Finanzmathematik werden stochastische Differentialgleichungen zur Modellierung von Aktienkursen genutzt. Ein Beispiel ist das berühmte *Black-Scholes-Modell*, welches 1973 von Fischer Black, Myron Scholes und Robert Merton vorgeschlagen wurde und auf der sogenannten geometrischen Brownschen Bewegung beruht [BS73, Mer73].

Stochastische Differentialgleichungen lassen sich nur in wenigen einfachen Fällen analytisch lösen. Um mögliche Pfade einer Lösung zu generieren benötigt man also numerische Methoden, die diese sinnvoll approximieren. Zur effizienten Lösung stochastischer Differentialgleichungen sind gewisse iterierte stochastische Integrale notwendig. Deren effiziente Simulation ist das Thema dieser Arbeit.

In Kapitel 1 werden zunächst die Definition des Wiener Prozesses sowie die Grundlagen der stochastischen Integration gegeben. Nach der Definition stochastischer Differentialgleichungen wird die Existenz und Eindeutigkeit von Lösungen untersucht und die Itô-Formel als wichtiges Werkzeug vorgestellt. Es wird anhand eines Beispiels der Unterschied zwischen gewöhnlichen und stochastischen Differentialgleichungen demonstriert.

Kapitel 2 behandelt die numerische Lösung stochastischer Differentialgleichungen. Es werden die bekannten Verfahren von Maruyama und Milstein vorgestellt und verglichen. Besonderer Fokus wird in diesem Zusammenhang auf die Bedeutung der iterierten stochastischen Integrale gelegt.

In Kapitel 3 werden die iterierten stochastischen Integrale genauer untersucht. Insbesondere geht es um deren numerische Simulation, wozu zunächst der Fourierreihen-Ansatz von Kloeden, Platen und Wright beschrieben wird. Weiterhin wird die Approximation des Restterms nach Wiktorsson vorgestellt. Beide Verfahren wurden in Julia und MATLAB implementiert, diese Implementierungen werden im letzten Abschnitt hinsichtlich ihrer theoretischen und praktischen Eigenschaften verglichen.

Kapitel 1: Einführung in Stochastische Differentialgleichungen

Den wichtigsten Bestandteil stochastischer Differentialgleichungen bildet der Wiener Prozess – ein rigoroses mathematisches Modell der physikalischen Brownschen Bewegung. In diesem Kapitel wird zunächst die Definition des stochastischen Wiener Prozesses gegeben und einige Eigenschaften aufgeführt. Darauf aufbauend werden die Grundlagen für die Theorie der stochastischen Differentialgleichungen vorgestellt und dann anhand der stochastischen Lotka-Volterra-Gleichungen ein Beispiel aus der Biologie gegeben.

1.1 Wiener Prozesse und stochastische Integration

Sei im Folgenden $(\Omega, \mathcal{F}, \mathrm{P})$ ein Wahrscheinlichkeitsraum mit vollständiger σ-Algebra $\mathcal{F}$ und Wahrscheinlichkeitsmaß P. Weiter sei eine vollständige und rechtsstetige Filtration $(\mathcal{F}_t)_{t\geq 0}$ in $\mathcal{F}$ gegeben.

Definition 1 (Wiener Prozess)**.**
Wir nennen den stochastischen Prozess $(W_t)_{t\geq 0}$ einen m-dimensionalen Wiener Prozess*, wenn folgende Eigenschaften gelten:*

i) $W_0 = 0_m$ P-*f.s.,*

ii) die Pfade $t \mapsto W_t$ *sind* P-*f.s. stetig,*

iii) die Inkremente $W_t - W_s$ *folgen einer* m*-dimensionalen Normalverteilung* $\mathcal{N}(0_m, (t-s)\cdot I_m)$ *für alle* $0 \leq s < t$

iv) und die Inkremente $W_t - W_s$ *sind unabhängig von den Inkrementen* $W_v - W_u$ *für* $0 \leq s < t \leq u < v$ *.*

Die Existenz eines stochastischen Prozesses, der die in Definition 1 geforderten Eigenschaften erfüllt, ist nicht von vornherein ersichtlich, doch es existieren verschiedene Wege, diese zu zeigen. Ein möglicher Ansatz nutzt den Erweiterungssatz von Daniell-Kolmogorov und den Stetigkeitssatz von Kolmogorov-Chentsov [KS91, Section 2.2].

Für einen m-dimensionalen Wiener Prozess $(W_t)_{t\geq 0}$ bilden die Komponenten $(W_t^i)_{t\geq 0}$, $1 \leq i \leq m$, stochastisch unabhängige eindimensionale Wiener Prozesse. Weiterhin sind die Pfade eines Wiener Prozesses nirgends differenzierbar und von unbeschränkter Variation auf jedem Intervall. Auf dem Intervall $[0, T]$ gilt für die quadratische Variation $[W]_T = T$. Bezüglich der von $(W_t)_{t\geq 0}$ erzeugten Filtration $(\mathcal{F}_t^W)_{t\geq 0}$ ist der Wiener Prozess ein Martingal, d.h. er erfüllt

$$\mathbb{E}\Big(W_t \,\Big|\, \mathcal{F}_s^W\Big) = W_s \tag{1.1}$$

für alle $0 \leq s \leq t$. Für einen Skalierungsfaktor $c > 0$ gilt, dass der Prozess $(\tilde{W}_t)_{t\geq 0}$ mit $\tilde{W}_t = \frac{1}{\sqrt{c}} W_{c\cdot t}$ wieder ein Wiener Prozess ist. Auch der durch $\tilde{W}_t = W_{t+s} - W_s$ definierte Prozess erfüllt für $s \geq 0$ die Eigenschaften aus Definition 1.

Sei $L^2_{\text{ad}}([0,T]\times\Omega)$ der Raum der reellwertigen stochastischen Prozesse $(f(t))_{t\in[0,T]}$ mit $\|f\|^2_{L^2_{\text{ad}}([0,T]\times\Omega)} = \int_0^T \mathbb{E}\Big(|f(t)|^2\Big)\,\mathrm{d}t < \infty$, die bezüglich $(\mathcal{F}_t)_{t\geq 0}$ adaptiert sind. Für diese Prozesse lässt sich das stochastische Itô-Integral

$$\left(\int_0^T f(t)\,\mathrm{d}W_t\right)(\omega) \coloneqq \int_0^T f(t,\omega)\,\mathrm{d}W_t(\omega) \tag{1.2}$$

bezüglich eines eindimensionalen Wiener Prozesses $(W_t)_{t\geq 0}$ als Grenzwert in $L^2(\Omega)$ von stochastischen Integralen geeigneter Stufenfunktionen definieren. Die Details, wie Existenz und Wohldefiniertheit dieses Grenzwertes, findet man zum Beispiel in [KS91, Section 3.2].

Theorem 2 (Eigenschaften des Itô-Integrals [KP99, Section 3.2])**.**
Für zwei Prozesse $f, g \in L^2_{\text{ad}}([0,T]\times\Omega)$ und $\alpha,\beta\in\mathbb{R}$ besitzt das Itô-Integral folgende Eigenschaften:

i) $\displaystyle\int_0^T \alpha f(t)+\beta g(t)\,\mathrm{d}W_t = \alpha\int_0^T f(t)\,\mathrm{d}W_t + \beta\int_0^T g(t)\,\mathrm{d}W_t,$

ii) $\displaystyle\mathbb{E}\Big(\int_0^T f(t)\,\mathrm{d}W_t\Big) = 0,$

iii) $\displaystyle\Big\|\int_0^T f(t)\,\mathrm{d}W_t\Big\|^2_{L^2(\Omega)} = \mathbb{E}\Big(\Big|\int_0^T f(t)\,\mathrm{d}W_t\Big|^2\Big) = \int_0^T \mathbb{E}\Big(|f(t)|^2\Big)\,\mathrm{d}t = \|f\|^2_{L^2_{\text{ad}}([0,T]\times\Omega)},$

iv) $\displaystyle\Big\langle\int_0^T f(t)\,\mathrm{d}W_t, \int_0^T g(t)\,\mathrm{d}W_t\Big\rangle_{L^2(\Omega)} = \langle f,g\rangle_{L^2_{\text{ad}}([0,T]\times\Omega)},$

v) der durch $\displaystyle X_t = \int_0^t f(s)\mathrm{d}W_s \coloneqq \int_0^T \mathbb{1}_{[0,t]}(s)f(s)\mathrm{d}W_s$ *definierte stochastische Prozess $(X_t)_{t\in[0,T]}$ ist ein stetiges Martingal bezüglich $(\mathcal{F}_t)_{t\geq 0}$.*

Für einen m-dimensionalen Wiener Prozess $(W_t)_{t\geq 0}$ und einen $\mathbb{R}^{d\times m}$-wertigen stochastischen Prozess $(f(t))_{t\in[0,T]}$ mit Komponenten $f^{i,j}\in L^2_{\text{ad}}([0,T]\times\Omega)$ ist das mehrdimensionale Itô-Integral definiert als

$$\int_0^t f(s)\,\mathrm{d}W_s \coloneqq \begin{pmatrix} \sum_{j=1}^m \int_0^t f^{1,j}(s)\,\mathrm{d}W_s^j \\ \vdots \\ \sum_{j=1}^m \int_0^t f^{d,j}(s)\,\mathrm{d}W_s^j \end{pmatrix}. \tag{1.3}$$

Nun lässt sich eine stochastische Differentialgleichung definieren.

Definition 3 (Stochastische Differentialgleichung)**.**
Betrachte das Zeitintervall $[0,T]\subset\mathbb{R}$. Seien $a^i, b^{i,j}\colon [0,T]\times\mathbb{R}^d\to\mathbb{R}$ für $1\leq i\leq d$ und

$1 \leq j \leq m$ *Funktionen, die* $\mathcal{B}\left([0,T] \times \mathbb{R}^d\right)$-$\mathcal{B}(\mathbb{R})$*-messbar sind. Weiterhin bezeichne* $a = (a^i)_{1 \leq i \leq d}$ *den* d*-dimensionalen Drift-Vektor und* $b = \left(b^{i,j}\right)_{\substack{1 \leq i \leq d \\ 1 \leq j \leq m}}$ *die* $d \times m$*-dimensionale Diffusionsmatrix, sodass* $a(t,x) \in \mathbb{R}^d$ *und* $b(t,x) \in \mathbb{R}^{d \times m}$*. Darüber hinaus sei* $\xi \colon \Omega \to \mathbb{R}^d$ *eine Zufallsvariable, die unabhängig von* $\mathcal{F}_\infty^W := \sigma\left(\bigcup_{t \geq 0} \mathcal{F}_t^W\right)$ *ist. Dann wird*

$$X_t = \xi + \int_0^t a(s, X_s)\,\mathrm{d}s + \int_0^t b(s, X_s)\,\mathrm{d}W_s \tag{1.4}$$

eine Stochastische Differentialgleichung *(SDE) mit Komponenten*

$$X_t^i = \xi^i + \int_0^t a^i(s, X_s)\,\mathrm{d}s + \sum_{j=1}^m \int_0^t b^{i,j}(s, X_s)\,\mathrm{d}W_s^j$$

genannt, wobei $(X_t)_{t \geq 0}$ *ein* d*-dimensionaler stochastischer Prozess mit stetigen Pfaden ist.*

1.2 Existenz und Eindeutigkeit der Lösungen von SDEs

In diesem Abschnitt soll untersucht werden, wie man sinnvoll eine Lösung einer SDE definieren kann, unter welchen Voraussetzungen Lösungen existieren und inwieweit diese eindeutig sind. Wie auch für gewöhnliche Differentialgleichungen existieren verschiedene Definitionen, welche Eigenschaften eine Lösung einer SDE haben sollte. Wir beschränken uns hier auf den Begriff der *starken Lösung*.

Da ξ als unabhängig von $\mathcal{F}_\infty^W$ vorausgesetzt wurde, ist $(W_t)_{t \geq 0}$ auch ein Wiener Prozess bzgl. der durch $\mathcal{G}_t := \sigma(\xi, W_s \colon 0 \leq s \leq t)$ definierten Filtration $(\mathcal{G}_t)_{t \geq 0}$. Geht man über zur augmentierten Filtration

$$\mathcal{F}_t := \sigma(\mathcal{G}_t \cup \mathcal{N}) \text{ mit } \mathcal{N} = \{N \subseteq \Omega \colon \exists G \in \mathcal{G}_\infty \text{ mit } N \subseteq G \text{ und } \mathrm{P}(G) = 0\}, \tag{1.5}$$

erhält man eine vollständige, rechtsstetige Filtration. Auch bzgl. $(\mathcal{F}_t)_{t \geq 0}$ ist $(W_t)_{t \geq 0}$ ein Wiener Prozess.

Definition 4 (Starke Lösung)**.**
Ein d*-dimensionaler Prozess* $(X_t)_{t \geq 0}$ *mit stetigen Pfaden wird* starke Lösung *der SDE* (1.4) *bzgl. des Wiener Prozesses* $(W_t)_{t \geq 0}$ *und der Anfangsbedingung* ξ *genannt, wenn die folgenden Bedingungen gelten:*

i) $(X_t)_{t \geq 0}$ *ist adaptiert bzgl.* $(\mathcal{F}_t)_{t \geq 0}$ *aus* (1.5)*,*

ii) $\mathrm{P}(X_0 = \xi) = 1$*,*

iii) für alle $1 \leq i \leq d$*,* $1 \leq j \leq m$ *und* $t \in [0,T]$ *gilt*

$$\mathrm{P}\left(\int_0^t |a^i(s, X_s)| + |b^{i,j}(s, X_s)|^2\,\mathrm{d}s < \infty\right) = 1$$

iv) und für alle $t \in [0,T]$ *gilt* (1.4) P*-f.s..*

Es wird nun die zu diesem Lösungsbegriff „passende“ Definition der Eindeutigkeit gegeben.

Definition 5 (Starke Eindeutigkeit)**.**
Wenn für jeden beliebigen Wiener Prozess $(W_t)_{t\geq 0}$ und jede Anfangsbedingung ξ für zwei starke Lösungen $(X_t)_{t\geq 0}$ und $(\tilde{X}_t)_{t\geq 0}$ der SDE (1.4) bzgl. $(W_t)_{t\geq 0}$ und ξ gilt, dass

$$\mathrm{P}\Big(X_t = \tilde{X}_t \,\forall t \in [0,T]\Big) = 1\,,$$

dann nennen wir die Lösung von (1.4) stark eindeutig.

Ähnlich dem deterministischen Fall kann man die Existenz einer starken Lösung mit Hilfe einer Art *Picard-Lindelöf-Iteration* zeigen. Dass heißt man startet mit $X_t^{(0)} \equiv \xi$ und iteriert

$$X_t^{(n+1)} \coloneqq \xi + \int_0^t a\left(s, X_s^{(n)}\right)\,\mathrm{d}s + \int_0^t b\left(s, X_s^{(n)}\right)\,\mathrm{d}W_s\,.$$

Um die Konvergenz der Iterierten zu gewährleisten, ist es sinnvoll zu fordern, dass die Koeffizientenfunktionen a und b Lipschitz-stetig sind. Zusätzlich fordert man, dass sie im zweiten Argument nur linear wachsen, um zu verhindern, dass die Iterierten $X_t^{(n)}$ „explodieren“. Dieser Ansatz führt zu folgendem Satz.

Theorem 6 (Existenz und Eindeutigkeit [KS91])**.**
Seien Drift $a\colon [0,T]\times\mathbb{R}^d \to \mathbb{R}^d$ und Diffusion $b\colon [0,T]\times\mathbb{R}^d \to \mathbb{R}^{d\times m}$ stetige Funktionen, welche die Lipschitzbedingung

$$\|a(t,x) - a(t,y)\|_{\mathbb{R}^d} + \|b(t,x) - b(t,y)\|_F \leq K\cdot\|x-y\|_{\mathbb{R}^d} \tag{1.6}$$

und die lineare Wachstumsbedingung

$$\|a(t,x)\|_{\mathbb{R}^d}^2 + \|b(t,x)\|_F^2 \leq K^2\cdot\Big(1+\|x\|_{\mathbb{R}^d}^2\Big) \tag{1.7}$$

für alle $t\in[0,T]$, $x,y\in\mathbb{R}^d$ und ein $K>0$ erfüllen. Seien weiterhin ein Wiener Prozess $(W_t)_{t\geq 0}$ und eine Anfangsbedingung $\xi\in L^2(\Omega)$ wie in Definition 3 gegeben.

Dann existiert ein stetiger, bzgl. $(\mathcal{F}_t)_{t\geq 0}$ adaptierter, d-dimensionaler stochastischer Prozess $(X_t)_{t\geq 0}$, welcher eine stark eindeutige Lösung der SDE (1.4) bzgl. $(W_t)_{t\geq 0}$ und ξ ist.

Falls zusätzlich $\xi\in L^{2n}(\Omega)$ für ein $n\in\mathbb{N}$ ist, so gilt

$$\mathbb{E}\Big(\max_{0\leq s\leq t}\|X_s\|_{\mathbb{R}^d}^{2n}\Big) \leq C\cdot\Big(1+\mathbb{E}\Big(\|\xi\|_{\mathbb{R}^d}^{2n}\Big)\Big)\cdot e^{C\cdot t} \tag{1.8}$$

für alle $t\in[0,T]$ mit $C=C(T,K,n,d,m)$ und ein $T>0$.

1.3 Itô-Formel

Ein wichtiges Werkzeug für die Arbeit mit stochastischen Integralen ist die *Itô-Formel*. Sie stellt das stochastische Analogon zur deterministischen Produktregel und partiellen Integration dar. Die Itô-Formel gilt für eine große Klasse stochastischer Prozesse, die zunächst definiert wird.

Definition 7 (Itô-Prozess)**.**
Sei ein m-dimensionaler Wiener Prozess $(W_t)_{t\geq 0}$ gegeben. Wir nennen den stochastischen Prozess $(X_t)_{t\in[0,T]}$ einen d-dimensionalen Itô-Prozess*, wenn es eine $\mathcal{F}_0$-messbare, d-dimensionale Zufallsvariable X_0, einen adaptierten, d-dimensionalen Prozess $(F_t)_{t\in[0,T]}$ und einen adaptierten, $d\times m$-dimensionalen Prozess $(G_t)_{t\in[0,T]}$ gibt, sodass sich $(X_t)_{t\in[0,T]}$ darstellen lässt als*

$$X_t = X_0 + \int_0^t F_s \,\mathrm{d}s + \int_0^t G_s \,\mathrm{d}W_s$$

und die Integrale existieren.

Offensichtlich ist jeder Wiener Prozess auch ein Itô-Prozess mit $F_t \equiv 0_m$, $G_t \equiv I_m$ und Anfangswert $X_0 = W_0 = 0$ P-f.s..

Proposition 8 (Eindimensionale Itô-Formel [KP99, Section 3.3])**.**
Sei $(X_t)_{t\in[0,T]}$ ein adaptierter, eindimensionaler Itô-Prozess bzgl. einem eindimensionalen Wiener Prozess mit

$$\int_0^t |F_s| \,\mathrm{d}s < \infty \quad \textit{und} \quad \int_0^t |G_s|^2 \,\mathrm{d}s < \infty$$

P-*f.s. für alle $t \in [0,T]$. Dann gilt für alle $\varphi \in C^2(\mathbb{R},\mathbb{R})$ und $t \in [0,T]$*

$$\begin{aligned}\varphi(X_t) &= \varphi(X_0) + \int_0^t \varphi'(X_s) \,\mathrm{d}X_s + \frac{1}{2}\int_0^t \varphi''(X_s) \,\mathrm{d}[X]_s \\ &= \varphi(X_0) + \int_0^t \varphi'(X_s)F_s \,\mathrm{d}s + \int_0^t \varphi'(X_s)G_s \,\mathrm{d}W_s + \frac{1}{2}\int_0^t \varphi''(X_s)G_s^2 \,\mathrm{d}s\end{aligned}$$

P-*f.s. und alle vorkommenden Integrale existieren.*

Mit Hilfe der Itô-Formel lassen sich einige einfachere stochastische Integrale direkt lösen. Setzt man zum Beispiel $X_t = W_t$, also $F_t \equiv 0$, $G_t \equiv 1$ und $X_0 = 0$ P-f.s., und $\varphi(x) = \frac{1}{2}x^2$, so erhält man

$$\frac{1}{2}W_t^2 = \int_0^t W_s \,\mathrm{d}W_s + \frac{1}{2}\int_0^t \,\mathrm{d}s\,.$$

Daraus ergibt sich dann die explizite Darstellung des stochastischen Integrals eines eindimensionalen Wiener Prozesses bezüglich sich selber

$$\int_0^t W_s \,\mathrm{d}W_s = \frac{1}{2}W_t^2 - \frac{1}{2}t\,. \tag{1.9}$$

Proposition 9 (Mehrdimensionale Itô-Formel [KP99, Section 3.4])**.**
Sei $(X_t)_{t\in[0,T]}$ ein adaptierter, d-dimensionaler Itô-Prozess bzgl. einem m-dimensionalen Wiener Prozess mit

$$\int_0^t |F_s^i| \,\mathrm{d}s < \infty \quad \textit{und} \quad \int_0^t |G_s^{i,j}|^2 \,\mathrm{d}s < \infty$$

P-*f.s. für alle* $t \in [0,T]$, $i = 1,\dots,d$ *und* $j = 1,\dots,m$. *Dann gilt für alle* $\varphi \in C^{1,2}([0,T]\times\mathbb{R}^d,\mathbb{R})$ *und* $t \in [0,T]$

$$\begin{aligned}\varphi(t,X_t) = \varphi(0,X_0) &+ \int_0^t \nabla_t\varphi(s,X_s)\,\mathrm{d}s + \int_0^t \nabla_x\varphi(s,X_s)F_s\,\mathrm{d}s\\ &+ \int_0^t \nabla_x\varphi(s,X_s)G_s\,\mathrm{d}W_s\\ &+ \frac{1}{2}\int_0^t \operatorname{tr}(G_s^\top\,\nabla_x^2\varphi(s,X_s)\,G_s)\,\mathrm{d}s\end{aligned}$$

P-*f.s., wobei* $\nabla_x\varphi(s,x) \in \mathbb{R}^{1\times d}$ *und* $\nabla_x^2\varphi(s,x) \in \mathbb{R}^{d\times d}$ *der Gradient und die Hessematrix der* x*-Komponenten von* φ *sind.*

1.4 Praktische Anwendungsbeispiele

Viele natürliche Prozesse werden traditionell durch gewöhnliche Differentialgleichungen angenähert, obwohl sich die zugrundeliegenden Akteure nur selten deterministisch verhalten. SDEs bieten in diesen Fällen die Möglichkeit die stochastische Natur der Dinge bei der Modellierung zu berücksichtigen. Deshalb haben SDEs in den letzten Jahren Einzug in alle Bereiche der naturwissenschaftlichen Forschung genommen: von Wettermodellen über die Simulation chemischer Reaktionen bis zur Modellierung komplexer Räuber-Beute-Beziehungen. In diesem Abschnitt soll am Beispiel der klassischen *Lotka-Volterra-Gleichungen* kurz gezeigt werden, wie sich deterministische Modelle durch stochastische Terme erweitern lassen.

Die Lotka-Volterra-Gleichungen für zwei Spezies lassen sich in folgender Form darstellen

$$\begin{aligned}B_t &= \int_0^t \alpha_1 B_s - \beta_1 B_s R_s\,\mathrm{d}s\,,\\ R_t &= \int_0^t -\alpha_2 R_s + \beta_2 B_s R_s\,\mathrm{d}s\,.\end{aligned}$$

Dabei sind B_t und R_t die Populationsgrößen der Beute- und Räuberspezies zum Zeitpunkt $t \geq 0$. Die Koeffizienten $\alpha_i > 0$ beschreiben das isolierte Verhalten ohne die Einwirkung der jeweils anderen Spezies – exponentielles Wachstum der Beute ohne Räuber und exponentielles Aussterben der Räuber ohne Beute – und die $\beta_i > 0$ charakterisieren die Änderung der Populationsgröße beim Aufeinandertreffen der beiden Spezies. In Abbildung 1 links ist ein typischer Verlauf der Lösung dargestellt. Man kann unter anderem die charakteristische Periodizität der Lösung und die Phasenverschiebung der beiden Komponenten erkennen.

Eine mögliche stochastische Erweiterung ergänzt das Modell um einen zweidimensionalen Wiener Prozess zu folgender SDE

$$\begin{aligned}B_t &= \int_0^t \alpha_1 B_s - \beta_1 B_s R_s\,\mathrm{d}s + \int_0^t \gamma_{1,1} B_s\,\mathrm{d}W_s^1 + \int_0^t \gamma_{1,2} R_s\,\mathrm{d}W_s^2\,,\\ R_t &= \int_0^t -\alpha_2 R_s + \beta_2 B_s R_s\,\mathrm{d}s + \int_0^t \gamma_{2,1} B_s\,\mathrm{d}W_s^1 + \int_0^t \gamma_{2,2} R_s\,\mathrm{d}W_s^2\,.\end{aligned}$$

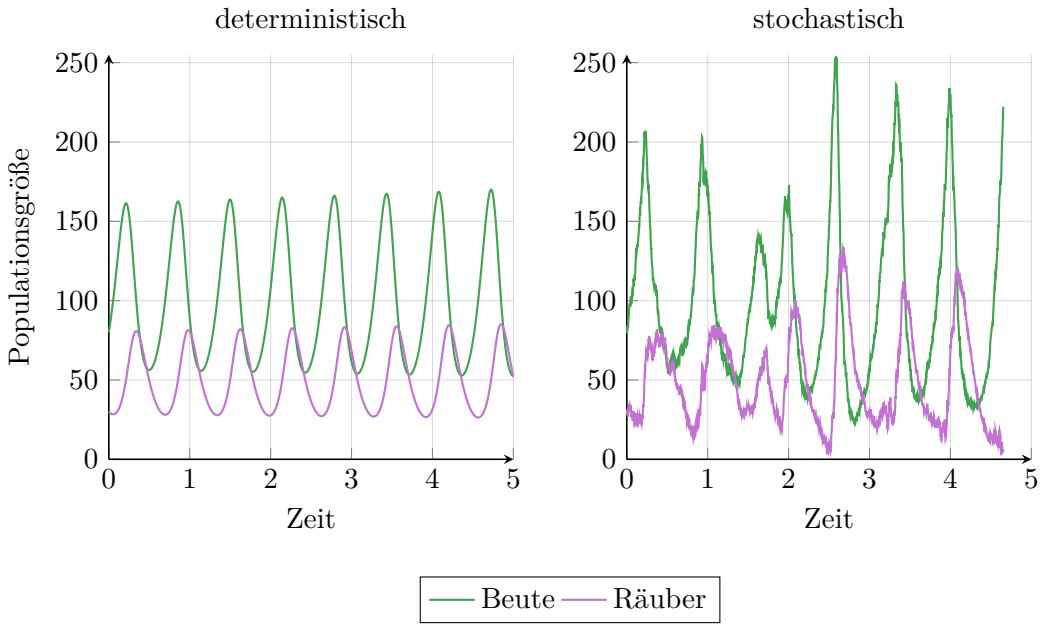

Abbildung 1. Simulation der deterministischen (links) und stochastischen (rechts) Lotka-Volterra-Gleichungen. Die Anfangswerte sind $B_0 = 80$ und $R_0 = 30$. Die weiteren Parameter wurden wie folgt gewählt: $\alpha_1 = \alpha_2 = 10$, $\beta_1 = 0.2$, $\beta_2 = 0.1$ und $\gamma_{1,1} = \gamma_{1,2} = \gamma_{2,1} = \gamma_{2,2} = 0.3$. Im stochastischen Fall sterben die Räuber im Zeitpunkt $t = 4.6$ aus.

Hierbei stehen die beiden Komponenten des Wiener Prozesses für zwei unabhängige, zufällige Umweltfaktoren, die jeweils durch die $\gamma_{i,j}$ gewichtet Einfluss auf die Populationsgrößen der Beute und der Räuber nehmen. Eine mögliche Realisation einer Lösung ist in Abbildung 1 rechts dargestellt. Tatsächlich ist der dargestellte Verlauf kein „typischer“ Verlauf – in vielen Fällen tritt das Aussterben der Räuber schon früher ein und auch ein Aussterben der Beute ist möglich.

Kapitel 2: Numerische Methoden zur Lösung von SDEs

Ein weitverbreitetes Verfahren zur Lösung stochastischer Differentialgleichungen ist eine Erweiterung des Euler-Verfahrens für gewöhnliche Differentialgleichungen, welches zuerst 1955 von G. Maruyama untersucht wurde [Mar55]. Ein zweites, in gewisser Weise besseres aber auch aufwendigeres Verfahren wurde 1974 von G. N. Milstein vorgestellt [Mil74].

In den folgenden Abschnitten soll zuerst geklärt werden, wie man die Konvergenz solcher Näherungsverfahren klassifiziert. Danach wird anhand der Verfahren von Maruyama und Milstein die Bedeutung der stochastischen iterierten Integrale demonstriert.

2.1 Konvergenz numerischer Methoden

Im Folgenden sei $\mathcal{I}^h = \{t_0 = 0, t_1, \ldots, t_N = T\}$ eine äquidistante Diskretisierung des Zeitintervalls $[0, T]$ mit Schrittweite h. Weiterhin sei $\Delta t_n \coloneqq t_{n+1} - t_n = h$ und $\Delta W_n \coloneqq W_{t_{n+1}} - W_{t_n}$.

Analog zur deterministischen Theorie existieren auch hier mehrere Wege, die Konvergenz zu definieren. Da sich die Fehler der Approximation im Allgemeinen über die Zeitpunkte aufsummieren, betrachtet man meist nur den Fehler im Endzeitpunkt. Weiterhin ist es üblich, die Konvergenz im quadratischen Mittel zu untersuchen.

Definition 10 (Konvergenz im quadratischen Mittel)**.**
Eine zeitdiskrete Approximation $(Y_t)_{t\in\mathcal{I}^h}$ mit maximaler Schrittweite h konvergiert im quadratischen Mittel gegen $(X_t)_{t\geq 0}$ zum Zeitpunkt T für $h \to 0$, wenn

$$\lim_{h\to 0} \mathbb{E}\Big(\|X_T - Y_T\|^2\Big) = 0\,.$$

$(Y_t)_{t\in\mathcal{I}^h}$ konvergiert mit Ordnung $\gamma > 0$ im quadratischen Mittel gegen $(X_t)_{t\geq 0}$ im Zeitpunkt T für $h \to 0$, wenn eine von h unabhängige Konstante $C > 0$ und ein $\delta_0 > 0$ existieren, so dass

$$\sqrt{\mathbb{E}(\|X_T - Y_T\|^2)} \leq C \cdot h^\gamma$$

für alle $h \in]0, \delta_0[$ gilt.

2.2 Euler-Maruyama-Verfahren

Das Euler-Maruyama-Verfahren ist eine Erweiterung des expliziten Euler-Verfahrens für gewöhnliche Differentialgleichungen. Sei $(X_t)_{t\geq 0}$ über eine stochastische Differentialgleichung wie in Definition 3 gegeben. Die zeitdiskrete Approximation $(Y_t)_{t\in\mathcal{I}^h}$ von $(X_t)_{t\geq 0}$ erhält man über folgende Iterationsvorschrift [Mar55]:

$$\begin{aligned} Y_0 &\coloneqq X_0\,, \\ Y_{n+1} &\coloneqq Y_n + a(t_n, Y_n)\cdot \Delta t_n + b(t_n, Y_n)\cdot \Delta W_n\,. \end{aligned}$$

Im Vergleich zum Euler-Verfahren wurde hier der stochastische Term $b(t_n, Y_n) \cdot \Delta W_n$ ergänzt. Insbesondere ergibt sich das Euler-Verfahren für $b \equiv 0$, also für deterministische Prozesse $(X_t)_{t\geq 0}$. Für die Simulation eines Pfades muss in jedem Schritt ein Inkrement des Wiener Prozesses $\Delta W_n \sim \mathcal{N}(0_m, hI_m)$ generiert werden. Dieses kann zum Beispiel über die Marsaglia-Polar-Methode aus gleichverteilten Zufallszahlen erzeugt werden. Alternativ bieten viele moderne Programmiersprachen bereits eine vorgefertigte Funktion zur Erzeugung normalverteilter Zufallszahlen.

Aufgrund der Einfachheit dieses Verfahrens ist es weitverbreitet und oft die erste Wahl. Allerdings ist es numerisch nicht das effizienteste Verfahren. Es kann gezeigt werden, dass das Euler-Maruyama-Verfahren unter den Voraussetzungen von Theorem 6 mit der Ordnung $\gamma = 0.5$ konvergiert [KP99].

2.3 Milstein-Verfahren

Das bekannteste Verfahren höherer Ordnung ist das Milstein-Verfahren. Für eine d-dimensionale SDE bzgl. einem m-dimensionalen Wiener Prozess ist die k-te Komponente des Verfahrens gegeben durch [Mil74]

$$\begin{aligned} Y_0^k &\coloneqq X_0^k\,, \\ Y_{n+1}^k &\coloneqq Y_n^k + a^k(t_n, Y_n)\cdot \Delta t_n + \sum_{j=1}^{m} b^{k,j}(t_n, Y_n)\cdot \Delta W_n^j \\ &\quad + \sum_{j_1,j_2=1}^{m}\sum_{l=1}^{d} b^{l,j_1}(t_n, Y_n)\frac{\partial b^{k,j_2}}{\partial x^l}(t_n, Y_n)\cdot \int_{t_n}^{t_{n+1}}\int_{t_n}^{s_1} \mathrm{d}W_s^{j_1}\,\mathrm{d}W_{s_1}^{j_2}\,. \end{aligned}$$

Die ersten drei Summanden entsprechen dem Euler-Maruyama-Verfahren, doch zusätzlich werden die partiellen Ableitungen von b und die iterierten Integrale $\int_{t_n}^{t_{n+1}}\int_{t_n}^{s_1} \mathrm{d}W_s^{j_1}\,\mathrm{d}W_{s_1}^{j_2}$ benötigt. Unter etwas stärkeren Voraussetzungen als für das Euler-Maruyama-Verfahren nötig waren (unter anderem höhere Glattheit von b), kann man zeigen, dass das Milstein-Verfahren mit der Ordnung $\gamma = 1$ im quadratischen Mittel konvergiert [Mil74, KP99].

Es ist allerdings im Allgemeinen schwierig, die benötigten iterierten Integrale zu simulieren, da die Verteilung dieser Zufallsvariablen nicht bekannt ist. Eine Möglichkeit ist, sich auf Fälle zu beschränken, in denen sich diese Integrale vereinfachen. Gilt für die Diffusionskoeffizienten zum Beispiel $b^{i,j}(t,x) = b^{i,j}(t)$, so reduziert sich das Milstein-Verfahren auf das Euler-Maruyama-Verfahren. Man spricht in diesem Fall von *additivem Rauschen*. Eine ähnliche Vereinfachung ist möglich, wenn die Diffusionskoeffizienten linear in x sind, d.h. $b^{i,j}(t,x) = b^{i,j}(t)x^i$. Unter Ausnutzung der Identität

$$\int_{t_n}^{t_{n+1}}\int_{t_n}^{s_1} \mathrm{d}W_s^{j_1}\,\mathrm{d}W_{s_1}^{j_2} + \int_{t_n}^{t_{n+1}}\int_{t_n}^{s_1} \mathrm{d}W_s^{j_2}\,\mathrm{d}W_{s_1}^{j_1} = \Delta W_n^{j_1}\cdot \Delta W_n^{j_2} \tag{2.1}$$

und (1.9) erhält man die folgende Iterationsvorschrift für das Milstein-Verfahren mit

linearem Rauschen:

$$Y_{n+1}^k = Y_n^k + a^k(t_n, Y_n) \cdot \Delta t_n + \sum_{j=1}^{m} b^{k,j}(t_n, Y_n) \cdot \Delta W_n^j + \frac{1}{2} \sum_{j_1,j_2=1}^{m} b^{k,j_1}(t_n) x^k b^{k,j_2}(t_n) \cdot (\Delta W_n^{j_1} \Delta W_n^{j_2} - \mathbb{1}_{\{j_1=j_2\}} \Delta t_n) .$$

Die beiden Fälle von additivem und linearem Rauschen sind Spezialfälle einer allgemeineren Kommutativitätsbedingung

$$\sum_{l=1}^{d} b^{l,j_1}(t,x) \frac{\partial b^{k,j_2}}{\partial x^l}(t,x) = \sum_{l=1}^{d} b^{l,j_2}(t,x) \frac{\partial b^{k,j_1}}{\partial x^l}(t,x)$$

für alle $k = 1, \ldots, d$, $j_1, j_2 = 1, \ldots, m$, $t \in [0,T]$ und $x \in \mathbb{R}^d$. Wenn diese erfüllt ist, lässt sich (2.1) ausnutzen und das Milstein-Verfahren für kommutatives Rauschen formulieren:

$$Y_{n+1}^k = Y_n^k + a^k(t_n, Y_n) \cdot \Delta t_n + \sum_{j=1}^{m} b^{k,j}(t_n, Y_n) \cdot \Delta W_n^j + \frac{1}{2} \sum_{j_1,j_2=1}^{m} \sum_{l=1}^{d} b^{l,j_1}(t_n, Y_n) \frac{\partial b^{k,j_2}}{\partial x^l}(t_n, Y_n) \cdot (\Delta W_n^{j_1} \Delta W_n^{j_2} - \mathbb{1}_{\{j_1=j_2\}} \Delta t_n) .$$

Viele praxisrelevante SDEs erfüllen die Kommutativitätsbedingung jedoch nicht. Ein Beispiel hierfür sind die in Abschnitt 1.4 vorgestellten stochastischen Lotka-Volterra-Gleichungen. Außerdem ist es nicht möglich, Verfahren mit Konvergenzordnung $\gamma > 0.5$ zu entwickeln, ohne dabei die iterierten Integrale zu berücksichtigen. Es ist also notwendig, effiziente Approximationen für die iterierten Integrale zu finden, um die Vorteile des Milstein- und anderer Verfahren höherer Konvergenzordnung ausnutzen zu können.

Kapitel 3: Simulation iterierter Integrale

Im vorherigen Kapitel wurde dargelegt, dass es wichtig ist, die beispielsweise im Milstein-Verfahren vorkommenden iterierten stochastischen Integrale zu simulieren. Für die Anwendung zur Lösung von SDEs ist es entscheidend, diese effizient zu berechnen, da die iterierten Integrale in jedem Iterationsschritt benötigt werden.

Im Folgenden werden zuerst einige Eigenschaften der iterierten Integrale gegeben. Danach wird der Ansatz von Kloeden, Platen und Wright zur numerischen Simulation sowie die Approximation des Restterms von Wiktorsson vorgestellt. Abschließend werden die theoretischen Eigenschaften der vorgestellten Algorithmen verglichen und anhand konkreter Implementierungen untersucht.

3.1 Definition und Eigenschaften iterierter Integrale

Sei ein m-dimensionaler Wiener Prozess $(W_t)_{t\in[0,T]}$ auf dem Zeitintervall $[0,T]$ gegeben, dann definieren wir die doppelt iterierten Integrale als

$$\mathcal{I}_{(i,j)}(h) \coloneqq \int_0^h \int_0^s \mathrm{d}W_r^i \,\mathrm{d}W_s^j = \int_0^h W_s^i \,\mathrm{d}W_s^j \tag{3.1}$$

für $h \in [0,T]$ und zwei Komponenten des Wiener Prozesses W_t^i und W_t^j mit $1 \leq i,j \leq m$. Im Fall $i = j$ lässt sich mit der eindimensionalen Itô-Formel nachrechnen, dass

$$\mathcal{I}_{(i,i)}(h) = \frac{1}{2}\left(W_h^i\right)^2 - \frac{1}{2}h \tag{3.2}$$

gilt (vgl. Abschnitt 1.3). In den folgenden Abschnitten beschäftigen wir uns mit der Simulation der iterierten Integrale im Fall $i \neq j$. Zusätzlich beschränken wir uns auf den Fall $h = T$, da in der Anwendung zur Lösung von SDEs in jedem Schritt nur ein Inkrement $\Delta W = W_{t+h} - W_t$ zur Verfügung steht, welches als Endpunkt des Prozesses $(\tilde{W}_t)_{t\in[0,\tilde{T}]}$ mit $\tilde{W}_{\tilde{T}} = \Delta W$ und $\tilde{T} = h$ aufgefasst werden kann.

Nach [KS91, Prop. 4.8] gilt unter gewissen Voraussetzungen für zwei stochastische Prozesse $(X_t)_{t\geq 0}$ und $(M_t)_{t\geq 0}$ und deren zeitskalierte Versionen $(Y_s)_{s\geq 0}$, $Y_s = X_{T(s)}$ und $(B_s)_{s\geq 0}$, $B_s = M_{T(s)}$ mit $T(s) = \inf\{t \geq 0 \colon [M]_t > s\}$

$$\int_0^t X_s \,\mathrm{d}M_s = \int_0^{[M]_t} Y_s \,\mathrm{d}B_s\,.$$

Setzt man $X_t = W_{h\cdot t}^i$ und $M_t = W_{h\cdot t}^j$ dann folgt $[M]_t = h \cdot t$, $T(s) = \frac{s}{h}$ und

$$\int_0^t W_{h\cdot s}^i \,\mathrm{d}W_{h\cdot s}^j = \int_0^{h\cdot t} W_s^i \,\mathrm{d}W_s^j\,.$$

Insbesondere gilt für $t = 1$

$$\int_0^h \int_0^s \mathrm{d}W_r^i \,\mathrm{d}W_s^j = h \cdot \int_0^1 \int_0^s \mathrm{d}\tilde{W}_r^i \,\mathrm{d}\tilde{W}_s^j\,, \tag{3.3}$$

wobei der skalierte Prozess $(\tilde{W}_t)_{t\in[0,1]}$ mit $\tilde{W}_t = \frac{1}{\sqrt{h}} W_{h\cdot t}$ wieder ein Wiener Prozess ist. Daher reicht es, iterierte Integrale auf dem Intervall $[0,1]$ zu betrachten und gegebenenfalls den Wiener Prozess und das resultierende Integral geeignet zu skalieren. Insbesondere wird in den Algorithmen in den folgenden Abschnitten immer von $T = 1$ ausgegangen.

3.2 Simulation nach Kloeden, Platen und Wright

In ihrem Artikel [KPW92] entwickeln die Autoren eine Reihendarstellung für doppelt iterierte Integrale $\mathcal{I}_{(i,j)}$ sowie für weitere mehrfach iterierte Integrale.

Für einen gegebenen m-dimensionalen Wiener Prozess $(W_t)_{0\le t\le T}$ wird die zugehörige „Brownsche Brücke“

$$\left(W_t - \frac{t}{T} W_T\right)_{0\le t\le T} \tag{3.4}$$

gebildet. Diese lässt sich komponentenweise in eine Fourier-Reihe

$$W_t^i - \frac{t}{T} W_T^i = \frac{1}{2} a_0^i + \sum_{r=1}^{\infty} \left(a_r^i \cos\left(\frac{2\pi r}{T} t\right) + b_r^i \sin\left(\frac{2\pi r}{T} t\right)\right) \tag{3.5}$$

mit normalverteilten Koeffizienten

$$a_r^i = \frac{2}{T} \int_0^T \left(W_s^i - \frac{s}{T} W_T^i\right) \cos\left(\frac{2\pi r}{T} s\right) \,\mathrm{d}s \tag{3.6}$$

und

$$b_r^i = \frac{2}{T} \int_0^T \left(W_s^i - \frac{s}{T} W_T^i\right) \sin\left(\frac{2\pi r}{T} s\right) \,\mathrm{d}s \tag{3.7}$$

entwickeln, wobei $a_0^i \sim \mathcal{N}\left(0, \frac{1}{3}T\right)$, $a_r^i \sim \mathcal{N}\left(0, \frac{T}{2\pi^2 r^2}\right)$ und $b_r^i \sim \mathcal{N}\left(0, \frac{T}{2\pi^2 r^2}\right)$ gelten. Zur Herleitung der Erwartungswerte und Varianzen siehe Anhang A.1. Aus dieser Darstellung ergibt sich die komponentenweise Reihenentwicklung

$$W_t^i = \frac{t}{T} W_T^i + \frac{1}{2} a_0^i + \sum_{r=1}^{\infty} \left(a_r^i \cos\left(\frac{2\pi r}{T} t\right) + b_r^i \sin\left(\frac{2\pi r}{T} t\right)\right) \tag{3.8}$$

für den Wiener Prozess in Abhängigkeit vom Wert im Endzeitpunkt W_T^i.

Setzt man die Reihendarstellung (3.8) in die Definition der iterierten Integrale (3.1) ein, so erhält man

$$\begin{aligned}
\mathcal{I}_{(i,j)}(T) &= \int_0^T W_s^i \,\mathrm{d}W_s^j \\
&= \int_0^T \frac{s}{T} W_T^i + \frac{1}{2} a_0^i + \sum_{r=1}^{\infty} \left(a_r^i \cos\left(\frac{2\pi r}{T} s\right) + b_r^i \sin\left(\frac{2\pi r}{T} s\right)\right) \,\mathrm{d}W_s^j \qquad (3.9) \\
&= \frac{1}{2} W_T^i W_T^j + \pi \sum_{r=1}^{\infty} r \cdot \left(a_r^i \left(b_r^j - \frac{1}{\pi r} W_T^j \right) - \left(b_r^i - \frac{1}{\pi r} W_T^i \right) a_r^j \right) . \qquad (3.10)
\end{aligned}$$

Eine ausführliche Herleitung wird im Anhang A.2 gegeben.

Für einen gegebenen Endpunkt W_T eines m-dimensionalen Wiener Prozesses kann man die iterierten Integrale über dem Intervall $[0,T]$ approximieren, indem man die Summe an einem Index $p > 0$ abschneidet und eine Realisierung der Koeffizienten a_r^i und b_r^i für alle $1 \le i \le m$ und $1 \le r \le p$ entsprechend ihrer Verteilung zufällig wählt. Führt man standardnormalverteilte Zufallsvariablen α_r^i und β_r^i ein, kann man die Koeffizienten darstellen als $a_r^i = \sqrt{\frac{T}{2\pi^2 r^2}}\alpha_r^i$ und $b_r^i = \sqrt{\frac{T}{2\pi^2 r^2}}\beta_r^i$. Die paarweisen iterierten Integrale lassen sich dann approximieren als

$$\begin{aligned}\mathcal{I}_{(i,j)}^p(T) = &\frac{1}{2}W_T^i W_T^j \\ &+ \frac{T}{2\pi}\sum_{r=1}^{p}\frac{1}{r}\cdot\left(\alpha_r^i\left(\beta_r^j - \sqrt{\frac{2}{T}}W_T^j\right) - \left(\beta_r^i - \sqrt{\frac{2}{T}}W_T^i\right)\alpha_r^j\right).\end{aligned} \tag{3.11}$$

Unter Beachtung von (3.2) für die Diagonalelemente ergibt sich die Matrixdarstellung

$$\begin{aligned}\mathcal{I}^p(T) = &\frac{1}{2}W_T W_T^\top - \frac{1}{2}TI_m \\ &+ \frac{T}{2\pi}\sum_{r=1}^{p}\frac{1}{r}\cdot\left(\alpha_r\left(\beta_r - \sqrt{\frac{2}{T}}W_T\right)^\top - \left(\beta_r - \sqrt{\frac{2}{T}}W_T\right)\alpha_r^\top\right),\end{aligned} \tag{3.12}$$

wobei jeweils die Variablen $\alpha_r = (\alpha_r^i)_{1\le i\le m} \in \mathbb{R}^m$, $\beta_r = (\beta_r^i)_{1\le i\le m} \in \mathbb{R}^m$ und $\mathcal{I}^p(T) = \left(\mathcal{I}_{(i,j)}^p(T)\right)_{\substack{1\le i\le m\\ 1\le j\le m}} \in \mathbb{R}^{m\times m}$ zusammengefasst wurden. Weiterhin sei $\tilde{\beta}_r := \frac{1}{r}\left(\beta_r - \sqrt{\frac{2}{T}}W_T\right)$. Für die Implementierung ist es vorteilhaft, die Summe auseinander zu ziehen, damit die dyadischen Produkte $\alpha_r\tilde{\beta}_r^\top$ nicht doppelt ausgewertet werden müssen. Mit der Notation

$$S^p := \sum_{r=1}^{p}\alpha_r\tilde{\beta}_r^\top \tag{3.13}$$

ergibt sich dann die Darstellung in Algorithmus 1a.

Fasst man weiter $\alpha = (\alpha_1 \mid \ldots \mid \alpha_p) \in \mathbb{R}^{m\times p}$ und $\tilde{\beta} = (\tilde{\beta}_1 \mid \ldots \mid \tilde{\beta}_p) \in \mathbb{R}^{m\times p}$ zusammen, so lässt sich die Summe (3.13) vereinfacht als Matrixprodukt schreiben:

$$S^p = \alpha\tilde{\beta}^\top. \tag{3.14}$$

Da moderne wissenschaftliche Programmiersprachen häufig über optimierte Routinen zur Matrixmultiplikation verfügen [BLAS], ist diese Darstellung oft schneller als die explizite Summe in Algorithmus 1a. Dies wird in Algorithmus 1b ausgenutzt.

Für die Verfahren in diesem Abschnitt kann folgender Satz zur Konvergenz im quadratischen Mittel gezeigt werden.

Theorem 11 (Konvergenz nach Kloeden, Platen, Wright [KPW92])**.**
Für $\mathcal{I}_{(i,j)}^p$ *wie in* (3.11) *gilt*

$$\mathbb{E}\left(\left|\mathcal{I}_{(i,j)}(T) - \mathcal{I}_{(i,j)}^p(T)\right|^2\right) \le \frac{1}{2\pi^2}\cdot\frac{T^2}{p}$$

für alle $1 \le i,j \le m$.

1. Simuliere
$$W_1 \in \mathbb{R}^m \colon W_1 \sim \mathcal{N}(0_m,\, I_m)$$
2. Simuliere
$$\begin{aligned} \alpha_r &\in \mathbb{R}^m \colon\ \alpha_r \sim \mathcal{N}(0_m,\, I_m) \\ \beta_r &\in \mathbb{R}^m \colon\ \beta_r \sim \mathcal{N}(0_m,\, I_m) \\ \tilde{\beta}_r &\in \mathbb{R}^m \colon\ \tilde{\beta}_r = \frac{1}{r}\left(\beta_r - \sqrt{2}W_1\right) \end{aligned}$$
und berechne
$$S^p = \sum_{r=1}^{p} \alpha_r \tilde{\beta}_r^\top$$
3. Berechne
$$\mathcal{I}^p = \frac{W_1 W_1^\top - I_m}{2} + \frac{1}{2\pi}(S^p - S^{p\top})$$

Algorithmus 1a. Kloeden, Platen, Wright

1. Simuliere
$$W_1 \in \mathbb{R}^m \colon W_1 \sim \mathcal{N}(0_m,\, I_m)$$
2. Simuliere
$$\begin{aligned} \alpha &\in \mathbb{R}^{m\times p} \colon\ \alpha_{i,j} \sim \mathcal{N}(0,\, 1) \\ \beta &\in \mathbb{R}^{m\times p} \colon\ \beta_{i,j} \sim \mathcal{N}(0,\, 1) \\ \tilde{\beta} &\in \mathbb{R}^{m\times p} \colon\ \tilde{\beta}_r = \frac{1}{r}\left(\beta_r - \sqrt{2}W_1\right) \end{aligned}$$
und berechne
$$S^p = \alpha\tilde{\beta}^\top$$
3. Berechne
$$\mathcal{I}^p = \frac{W_1 W_1^\top - I_m}{2} + \frac{1}{2\pi}(S^p - S^{p\top})$$

Algorithmus 1b. Kloeden, Platen, Wright: Version 2

Weiterhin konnten Kloeden und Platen zeigen, dass man für ein Itô-Taylor-Verfahren (wie die Verfahren aus Kapitel 2) der starken Konvergenzordnung γ und Schrittweite h den Abschneideindex p so wählen muss, dass

$$\mathbb{E}\left(\left|\mathcal{I}_{(i,j)}(h) - \mathcal{I}^p_{(i,j)}(h)\right|^2\right) \leq K \cdot h^{2\gamma+1} \tag{3.15}$$

erfüllt ist [KP99, Korollar 10.6.5]. Nutzt man die in diesem Abschnitt vorgestellte Approximation, so folgt mit Theorem 11, dass

$$p \geq \frac{C_1}{K} \cdot h^{1-2\gamma} \in \mathcal{O}\left(h^{1-2\gamma}\right)$$

mit $C_1 = \frac{1}{2\pi^2}$ gelten muss. Insbesondere muss für das Milstein-Verfahren $p \in \mathcal{O}\left(h^{-1}\right)$ gewählt werden.

3.3 Restglied-Approximation nach Wiktorsson

Im vorherigen Abschnitt wurde der Restterm beim Abschneiden der Summe in (3.10) vernachlässigt. In dem Artikel [Wik01] wird nun die Verteilung dieses Restgliedes analysiert und daraus resultierend ein Korrektur-Term vorgeschlagen.

Der Rest lässt sich in der Notation des vorangegangenen Abschnitts darstellen als

$$R^p = \frac{T}{2\pi} \sum_{r=p+1}^{\infty} \left(\alpha_r \tilde{\beta}_r^\top - \tilde{\beta}_r \alpha_r^\top\right).$$

Mit dem Vektorisierungsoperator $\mathrm{vec}(\,\cdot\,)$, welcher die Spalten einer Matrix untereinander in einen Vektor schreibt, und dem Kroneckerprodukt $\otimes\colon \mathbb{R}^{m\times n} \times \mathbb{R}^{p\times q} \to \mathbb{R}^{mp\times nq}$ ergibt sich für $u, v \in \mathbb{R}^m$

$$\mathrm{vec}(uv^\top) = v \otimes u\,.$$

Zusätzlich sei die Matrix $P_m \in \mathbb{R}^{m^2\times m^2}$ so gewählt, dass $P_m(u \otimes v) = v \otimes u$ für alle Vektoren $u, v \in \mathbb{R}^m$ gilt. Dann folgt für das Restglied

$$\mathrm{vec}(R^p) = \frac{T}{2\pi} \sum_{r=p+1}^{\infty} (P_m - I_{m^2}) \left(\alpha_r \otimes \tilde{\beta}_r\right) \in \mathbb{R}^{m^2}.$$

Da R^p antisymmetrisch ist, dass heißt $R^{p\top} = -R^p$, reicht es, die $M \coloneqq \frac{m(m-1)}{2}$ Einträge unterhalb der Diagonalen zu betrachten. Zu diesem Zweck führt Wiktorsson die Matrix $K_m \in \mathbb{R}^{M\times m^2}$ ein, welche aus der vektorisierten Matrix $\mathrm{vec}(R^p)$ genau die Elemente unterhalb der Diagonalen von R^p auswählt. Diese Matrix hat die folgende Struktur:

$$K_m = \begin{pmatrix} 0_{m-1\times 1} & I_{m-1} & 0_{m-1\times m(m-1)} \\ 0_{m-2\times m+2} & I_{m-2} & 0_{m-2\times m(m-2)} \\ \vdots & \vdots & \vdots \\ 0_{m-k\times (k-1)m+k} & I_{m-k} & 0_{m-k\times m(m-k)} \\ \vdots & \vdots & \vdots \\ 0_{1\times (m-2)m+m-1} & 1 & 0_{1\times m} \end{pmatrix}.$$

Sei $\varepsilon^p \coloneqq K_m \operatorname{vec}(R^p) \in \mathbb{R}^M$ und $\operatorname{mat}(\,\cdot\,)\colon \mathbb{R}^{m^2} \to \mathbb{R}^{m\times m}$ der inverse Operator zu $\operatorname{vec}(\,\cdot\,)$ für quadratische Matrizen. In einigen Programmiersprachen wird diese Operation durch den Befehl `reshape(v,m,m)` für Vektoren $v \in \mathbb{R}^{m^2}$ ausgedrückt. Dann kann das Restglied R^p wie folgt rekonstruiert werden:

$$R^p = \operatorname{mat}\left((I_{m^2} - P_m)K_m^\top \varepsilon^p\right).$$

Wiktorsson konnte zeigen, dass der Zufallsvektor $\frac{2\pi}{T\sqrt{a^p}}\varepsilon^p$ für $p \to \infty$ in Verteilung gegen einen normalverteilten Vektor ε_∞ mit Erwartungswert $\mathbb{E}(\varepsilon_\infty) = 0$ und Varianz $\mathbb{E}(\varepsilon_\infty^2) = \Sigma_\infty$ konvergiert [Wik01]. Dabei ist $a^p \coloneqq \sum_{r=p+1}^{\infty} \frac{1}{r^2} = \psi_1(p+1)$ die Trigamma-Funktion an der Stelle $p+1$ und

$$\Sigma_\infty = 2I_M + \frac{2}{T}K_m(I_{m^2} - P_m)\left(W_T W_T^\top \otimes I_m\right)(I_{m^2} - P_m)K_m^\top. \tag{3.16}$$

Führt man einen standardnormalverteilten Zufallsvektor $G^p \in \mathbb{R}^M$ ein, kann der Rest mit $\varepsilon_\infty = \sqrt{\Sigma_\infty}G^p$ approximiert werden als

$$R^p \approx \tilde{R}^p \coloneqq \operatorname{mat}\left((I_{m^2} - P_m)K_m^\top \frac{T}{2\pi}\sqrt{a^p}\sqrt{\Sigma_\infty}G^p\right). \tag{3.17}$$

Für die Matrixwurzel gilt die explizite Darstellung [Wik01]

$$\sqrt{\Sigma_\infty} = \frac{\Sigma_\infty + 2\sqrt{1 + \frac{\|W_T\|^2}{T}}I_M}{\sqrt{2}\left(1 + \sqrt{1 + \frac{\|W_T\|^2}{T}}\right)}. \tag{3.18}$$

Die Approximation der iterierten Integrale nach Wiktorsson ergibt sich damit zu

$$\tilde{\mathcal{I}}^p_{(i,j)}(T) = \frac{1}{2}W_T^i W_T^j + \frac{T}{2\pi}(S^p - S^{p\top}) + \tilde{R}^p \tag{3.19}$$

Diese Darstellung führt aufbauend auf Algorithmus 1b zu Algorithmus 2a.

Algorithmus 2a enthält mehrere Matrixmultiplikationen mit Matrizen in der Größenordnung $m^2 \times m^2$ und ein teures Kronecker-Produkt mit einer Einheitsmatrix, welches auf algorithmischer Ebene lediglich Werte dupliziert. Um den Rechenaufwand zu verringern setzt man zunächst (3.16) in (3.18) ein:

$$\begin{aligned}\sqrt{\Sigma_\infty} &= \frac{2I_M + \frac{2}{T}K_m(I_{m^2} - P_m)\left(W_T W_T^\top \otimes I_m\right)(I_{m^2} - P_m)K_m^\top + 2\sqrt{1 + \frac{\|W_T\|^2}{T}}I_M}{\sqrt{2}\left(1 + \sqrt{1 + \frac{\|W_T\|^2}{T}}\right)} \\ &= \frac{\sqrt{2}K_m(I_{m^2} - P_m)\left(W_T W_T^\top \otimes I_m\right)(I_{m^2} - P_m)K_m^\top}{T\left(1 + \sqrt{1 + \frac{\|W_T\|^2}{T}}\right)} + \sqrt{2}I_M. \end{aligned} \tag{3.20}$$

1. Simuliere
$$W_1 \in \mathbb{R}^m \colon W_1 \sim \mathcal{N}(0_m,\, I_m)$$
2. Simuliere
$$\begin{aligned} \alpha &\in \mathbb{R}^{m\times p} \colon\ \alpha_{i,j} \sim \mathcal{N}(0,\, 1) \\ \beta &\in \mathbb{R}^{m\times p} \colon\ \beta_{i,j} \sim \mathcal{N}(0,\, 1) \\ \tilde{\beta} &\in \mathbb{R}^{m\times p} \colon\ \tilde{\beta}_r = \frac{1}{r}\left(\beta_r - \sqrt{2}W_1\right) \end{aligned}$$
und berechne
$$S^p = \alpha\tilde{\beta}^\top$$
3. Simuliere
$$G^p \in \mathbb{R}^M \colon G^p \sim \mathcal{N}(0_M,\, I_M)$$
und berechne
$$\tilde{R}^p = \operatorname{mat}\left((I_{m^2} - P_m)K_m^\top\sqrt{a^p}\sqrt{\Sigma_\infty}G^p\right)$$
4. Berechne
$$\tilde{\mathcal{I}}^p = \frac{W_1W_1^\top - I_m}{2} + \frac{1}{2\pi}(S^p - S^{p\top}) + \frac{1}{2\pi}\tilde{R}^p$$

Algorithmus 2a. Wiktorsson

Weiterhin benötigt man folgende Identitäten:

$$\begin{aligned}(I_{m^2} - P_m)K_m^\top K_m(I_{m^2} - P_m) &= I_{m^2} - P_m\,,\\ (I_{m^2} - P_m)\operatorname{vec}(A) &= \operatorname{vec}(A - A^\top)\,,\\ \operatorname{mat}((I_{m^2} - P_m)\operatorname{vec}(A)) &= A - A^\top,\\ (A \otimes I_m)\operatorname{vec}(B) &= \operatorname{vec}(A \cdot B)\,,\end{aligned}$$

für Matrizen $A, B \in \mathbb{R}^{m\times m}$. Mit diesen und (3.20) ergibt sich für (3.17)

$$\tilde{R}^p = \tfrac{1}{2\pi}\operatorname{mat}\left((I_{m^2} - P_m)\left(\frac{(W_TW_T^\top\otimes I_m)(I_{m^2}-P_m)\sqrt{2a^p}K^\top G^p}{1+\sqrt{1+\frac{\|W_T\|^2}{T}}} + T\sqrt{2a^p}K^\top G^p\right)\right)\,. \tag{3.21}$$

Führt man die Matrix $\tilde{G}^p = \operatorname{mat}\left(\sqrt{2a^p}K^\top G^p\right)$ ein, so gilt

$$\tilde{G}^p \in \mathbb{R}^{m\times m}\colon\ \begin{cases}\tilde{G}^p_{i,j} \sim \mathcal{N}(0,\,2a^p), & i<j\\ \tilde{G}^p_{i,j} = 0\,, & i\geq j\end{cases}\,.$$

Gleichung (3.21) lässt sich damit vereinfachen zu

$$\begin{aligned}\tilde{R}^p &= \frac{1}{2\pi}\operatorname{mat}\left((I_{m^2} - P_m)\operatorname{vec}\left(\frac{W_TW_T^\top(\tilde{G}^p - \tilde{G}^{p\top})}{1+\sqrt{1+\frac{\|W_T\|^2}{T}}} + T\tilde{G}^p\right)\right)\\ &= \frac{1}{2\pi}\left(V^p - V^{p\top}\right)\end{aligned} \tag{3.22}$$

mit $V^p \coloneqq \frac{1}{1+\sqrt{1+\frac{\|W_T\|^2}{T}}}W_TW_T^\top(\tilde{G}^p - \tilde{G}^{p\top}) + T\tilde{G}^p$. Mit dieser Darstellung ergibt sich der verbesserte Algorithmus 2b.

Die Konvergenzrate der erweiterten Verfahren aus diesem Abschnitt wird im folgenden Satz geklärt.

Theorem 12 (Konvergenz nach Wiktorsson [Wik01])**.**
Mit der zusätzlichen Approximation des Restterms wie in (3.19) *gilt*

$$\mathbb{E}\left(\left|\mathcal{I}_{(i,j)}(T) - \tilde{\mathcal{I}}^p_{(i,j)}(T)\right|^2 \,\middle|\, W_T\right) \leq \frac{m(m-1)(m+4\frac{\|W_T\|^2}{T})}{24\pi^2}\cdot\frac{T^2}{p^2}$$

und

$$\mathbb{E}\left(\left|\mathcal{I}_{(i,j)}(T) - \tilde{\mathcal{I}}^p_{(i,j)}(T)\right|^2\right) \leq \frac{5m^2(m-1)}{24\pi^2}\cdot\frac{T^2}{p^2}$$

für alle $1 \leq i, j \leq m$.

Zusammen mit (3.15) folgt aus Theorem 12, dass für ein Verfahren der starken Konvergenzordnung γ und Schrittweite h mit der Approximation $\tilde{\mathcal{I}}^p$ der Abschneideindex p so gewählt werden muss, dass

$$p \geq \sqrt{\frac{C_2}{K}}\cdot h^{\frac{1}{2}-\gamma} \in \mathcal{O}\left(h^{\frac{1}{2}-\gamma}\right)$$

1. Simuliere
$$W_1 \in \mathbb{R}^m \colon W_1 \sim \mathcal{N}(0_m,\, I_m)$$

2. Simuliere
$$\begin{aligned} \alpha &\in \mathbb{R}^{m\times p} \colon\ \alpha_{i,j} \sim \mathcal{N}(0,\, 1) \\ \beta &\in \mathbb{R}^{m\times p} \colon\ \beta_{i,j} \sim \mathcal{N}(0,\, 1) \\ \tilde{\beta} &\in \mathbb{R}^{m\times p} \colon\ \tilde{\beta}_r = \frac{1}{r}\left(\beta_r - \sqrt{2}W_1\right) \end{aligned}$$
und berechne
$$S^p = \alpha\tilde{\beta}^\top$$

3. Simuliere
$$\tilde{G}^p \in \mathbb{R}^{m\times m} \colon\ \begin{cases} \tilde{G}^p_{i,j} \sim \mathcal{N}(0,\, 2a_n), & i < j \\ \tilde{G}^p_{i,j} = 0\,, & i \geq j \end{cases}$$
und berechne
$$\begin{aligned} V^p &= \frac{1}{1+\sqrt{1+\|W_1\|^2}} W_1 W_1^\top (\tilde{G}^p - \tilde{G}^{p\top}) + \tilde{G}^p \\ \tilde{S}^p &= S^p + V^p \end{aligned}$$

4. Berechne
$$\tilde{\mathcal{I}}^p = \frac{W_1 W_1^\top - I_m}{2} + \frac{1}{2\pi}(\tilde{S}^p - \tilde{S}^{p\top})$$

Algorithmus 2b. Wiktorsson: Version 2

	#Zufallszahlen	Konvergenz in h	Größenordnung p
KPW	$2pm$	$\mathcal{O}\left(h^2 p^{-1}\right)$	$\mathcal{O}\left(h^{1-2\gamma}\right)$
Wik	$2pm + \frac{1}{2}m(m-1)$	$\mathcal{O}\left(m^3 h^2 p^{-2}\right)$	$\mathcal{O}\left(m^{\frac{3}{2}} h^{\frac{1}{2}-\gamma}\right)$

Tabelle 1. Vergleich der theoretischen Eigenschaften der Algorithmen mit und ohne Restglied-Approximation. Die Spalte Größenordnung p gibt dabei an, wie der Abschneideindex p gewählt werden muss, um eine Konvergenzrate von $\mathcal{O}\left(h^{2\gamma+1}\right)$ in h zu erreichen.

mit $C_2 \coloneqq \frac{m(m-1)(m+4\frac{\|W_T\|^2}{T})}{24\pi^2}$ erfüllt ist. Insbesondere gilt für das Milstein-Verfahren, dass $p \in \mathcal{O}\left(h^{-\frac{1}{2}}\right)$ gewählt werden muss.

3.4 Vergleich der Algorithmen

In diesem Abschnitt werden die vorgestellten Algorithmen in Bezug auf ihre theoretischen Eigenschaften und insbesondere hinsichtlich ihrer Laufzeiten untersucht. Dazu wurden alle vier Algorithmen in der Programmiersprache Julia [BEK+17] implementiert. Die Quelltexte sind im Anhang B aufgeführt. Dort sind zusätzlich auch MATLAB-Implementierungen zu finden [MAT18]. Die Laufzeiten wurden gemessen auf einem Rechner mit einer AMD Ryzen 5 2600 CPU mit sechs Prozessorkernen, 16 GB Arbeitsspeicher und der aktuellen Julia LTS Version 1.0.3 auf einem 64-Bit Linux-Betriebssystem. Im Folgenden seien die beiden Versionen des Algorithmus von Kloeden, Platen und Wright mit KPW1 und KPW2 bezeichnet, die beiden Versionen, welche die Restgliedapproximation von Wiktorsson beinhalten mit Wik1 und Wik2.

Theoretische Eigenschaften

Die folgenden Eigenschaften gelten jeweils für beide Versionen der vorgestellten Algorithmen.

Sowohl der Algorithmus von Kloeden, Platen und Wright als auch der von Wiktorsson benötigen $2pm$ normalverteilte Zufallszahlen für die Summe (3.13). Der Algorithmus von Wiktorsson verwendet zusätzlich $\frac{1}{2}m(m-1)$ Zufallszahlen, um den Restterm der Summe zu approximieren.

Sei h die Schrittweite eines Verfahrens zur numerischen Lösung von SDEs. Dann werden, zum Beispiel für das Milstein-Verfahren, die iterierten Integrale eines Inkrements des Wiener Prozesses mit dieser Schrittweite benötigt. Um die Konvergenzrate des gewählten Verfahrens zu erhalten, muss die Genauigkeit der Approximation der Integrale und damit der Abschneideindex p in Abhängigkeit von h gewählt werden. Genauer braucht man für ein Verfahren der Ordnung γ, eine Konvergenzrate von $\mathcal{O}\left(h^{2\gamma+1}\right)$. So steigt der Aufwand für den Algorithmus von Wiktorsson in der Anwendung nur proportional zu $h^{\frac{1-2\gamma}{2}}$, während der Aufwand für KPW proportional zu $h^{1-2\gamma}$ ist. Die Ergebnisse aus den Theoremen 11 und 12 sowie die resultierenden Größenordnungen von p sind in Tabelle 1 gegenübergestellt.

Algorithmus	Speicherplatz (#Floats)
KPW1	$2m^2 + 2m$
KPW2	$2m^2 + 2pm$
Wik1	$4m^3 + \frac{5}{2}m^2 + \left(\frac{3}{2} + 2p\right) m + 15$
Wik2	$2m^2 + 2pm + m$

Tabelle 2. Der notwendige Speicherplatz für die in Anhang B.1 gegebenen Julia-Implementierungen. Angegeben ist die Anzahl der Gleitkommazahlen (Floats), die für einen Aufruf benötigt werden.

Speicherplatz

Die beiden Implementierungen des Algorithmus von Kloeden, Platen und Wright unterscheiden sich nur darin, dass die Summe einmal mit Hilfe einer Schleife und einmal mittels eines Matrixproduktes umgesetzt wurde. Dies ändert nichts an der Laufzeitkomplexität des Algorithmus, profitiert aber von spezialisierten Routinen zur Matrixmultiplikation. In den vorkompilierten Versionen von Julia, die im Internet zur Verfügung gestellt werden, ist openBLAS bereits enthalten [BLAS]. Dies ist eine Software-Bibliothek, die optimierte Funktionen zur Berechnung von z.B. Vektor-Matrix- und Matrix-Matrix-Produkten gemäß der *Basic Linear Algebra Subprograms* (BLAS)-Spezifikation anbietet. Was sich aber ändert ist der notwendige Speicher, da in KPW1 jeweils nur zwei Vektoren der Länge m im Speicher gehalten werden müssen, während in KPW2 zwei Matrizen der Größe $2pm$ benötigt werden. Der notwendige Speicherplatz gemessen an den erstellten Julia-Implementierungen für alle vier Algorithmen ist in Tabelle 2 aufgelistet.

Für Wik2 entspricht der Speicherplatzbedarf fast dem Algorithmus KPW2, da der Speicher für die dort verwendeten Matrizen effizient wiederverwertet werden kann. Zusätzlich benötigt man allerdings einen Vektor der Länge m für die Matrixmultiplikation in Zeile 18 des Algorithmus. Der benötigte Speicherplatz für Wik1 ist sehr hoch, da die Matrizen explizit gespeichert werden und der Speicher nicht gut wiederverwertet werden kann. Diese Matrizen sind allerdings sehr dünn besetzt und durch die Speicherung im *Compressed Sparse Column* (CSC) Format benötigt man für eine $k \times l$-Matrix mit N Nicht-Null-Einträgen nur $2N + l + 3$ Floats (zwei davon zur Speicherung von k und l). Die Dimensionen der vorkommenden Matrizen und der entsprechend benötigte Speicherplatz ist in Tabelle 3 aufgeführt.

Laufzeitkomplexität

Die Laufzeitkomplexität wird für KPW und Wik2 dominiert durch die Summe (3.13). Dies führt zu einer Laufzeitkomplexität von $\mathcal{O}\left(pm^2\right)$. Aufgrund der expliziten Berechnung der benötigten Matrizen in Wik1 ergibt sich ein zusätzlicher Term, der die vorherige Summe dominieren kann. Durch die Nutzung des CSC-Formats für dünn besetzte Matrizen ist dieser Aufwand ungefähr proportional zu m^4. Wik1 hat damit eine Laufzeitkomplexität von $\mathcal{O}\left(pm^2 + m^4\right)$.

	k	l	N	Speicherplatz: $2N + l + 3$
P	m^2	m^2	m^2	$3m^2 + 3$
K	$\frac{m(m-1)}{2}$	m^2	$\frac{m(m-1)}{2}$	$2m^2 - m + 3$
$K(I-P)$	$\frac{m(m-1)}{2}$	m^2	$m(m-1)$	$3m^2 - 2m + 3$
Σ	$\frac{m(m-1)}{2}$	$\frac{m(m-1)}{2}$	$(2m-3)\frac{m(m-1)}{2}$	$2m^3 - \frac{9}{2}m^2 + \frac{5}{2}m + 3$
$\sqrt{\Sigma}$	$\frac{m(m-1)}{2}$	$\frac{m(m-1)}{2}$	$(2m-3)\frac{m(m-1)}{2}$	$2m^3 - \frac{9}{2}m^2 + \frac{5}{2}m + 3$

Tabelle 3. Dimensionen der $k \times l$ Matrizen, die im Algorithmus Wik1 auftreten, sowie die Anzahl der Nicht-Null-Einträge N und die Anzahl der Floats, die für die Speicherung im Compressed Sparse Column (CSC) Format benötigt werden. m steht hier wie oben für die Dimension des Wiener Prozesses.

Laufzeiten

In Abbildung 2 sind die Laufzeiten der vier Algorithmen abhängig von der Dimension des Wiener Prozesses und für feste Werte von p zu sehen. Man kann gleich erkennen, dass die Implementierung Wik1 durch die Abhängigkeit von m^4 im Vergleich sehr ineffizient ist. Zur besseren Übersicht sind in Abbildung 3 nur die Laufzeiten für KPW1, KPW2 und Wik2 aufgeführt. Man erkennt insbesondere die quadratische Abhängigkeit von der Dimension des Wiener Prozesses.

Um das asymptotische Verhalten besser zu sehen, sind in Abbildung 4 die Laufzeiten für KPW2 und Wik2 für hochdimensionale Wiener Prozesse in einem Loglog-Plot dargestellt. Hier nähern sich die Graphen langsam der erwarteten Steigung von 2 an. Darüber hinaus kann man für großes m den zusätzlichen Aufwand für die Restapproximation erkennen. Bei $p = 128$ und $m = 1024$ ergibt sich ein Laufzeitunterschied von im Schnitt 6 ms zwischen KPW2 und Wik2.

In den Abbildungen 5 und 6 sind die Laufzeiten der Algorithmen in Abhängigkeit von dem Abschneideindex p für einige feste Werte von m zu sehen. Insbesondere in Abbildung 6 ist die lineare Abhängigkeit von p gut zu erkennen. Man sieht auch, wie die Implementierung mittels der Matrixmultiplikation in Wik2 eine deutliche Verbesserung gegenüber Wik1 bringt.

In Abbildung 7 ist das asymptotische Verhalten der Algorithmen KPW2 und Wik2 für hohe Werte von p dargestellt. Auch hier kann man die Annäherung an die erwartete Steigung von 1 für lineare Abhängigkeiten erkennen. Insbesondere sieht man, dass sich die Laufzeiten der beiden Algorithmen aneinander annähern sobald die Laufzeit durch die Summe (3.13) dominiert wird. Der zusätzliche Aufwand für die Restapproximation ist nur von der Dimension des Wiener Prozesses abhängig. Für $p \gg m$ ist dieser also vernachlässigbar und die Konvergenzverbesserung, welche die Restapproximation bringt, überwiegt die zusätzlichen Kosten.

Der Vorteil der Restapproximation von Wiktorsson liegt darin, dass der Abschneideindex p für ein Ordnung 1-Verfahren nur proportional zu $h^{-\frac{1}{2}}$ gewählt werden muss, anstatt proportional zu h^{-1} ohne die Restapproximation. Allerdings entsteht zusätzlich eine Abhängigkeit von der Dimension m des Wiener Prozesses. In Abbildung 8 sind die

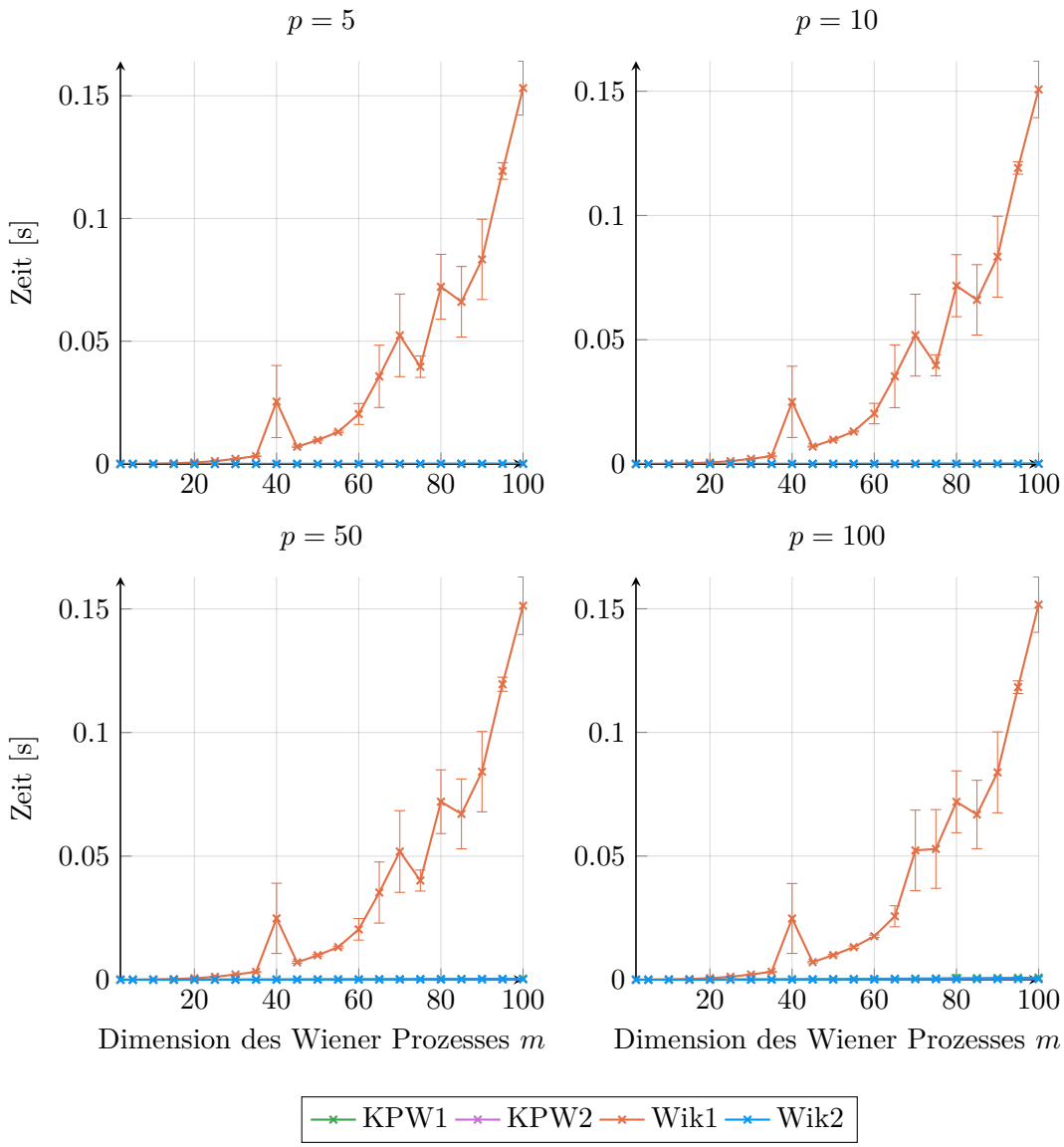

Abbildung 2. Laufzeiten aller vier Algorithmen abhängig von m für verschiedene Werte von p. Die Zeiten wurden gemittelt über 500 Durchläufe.

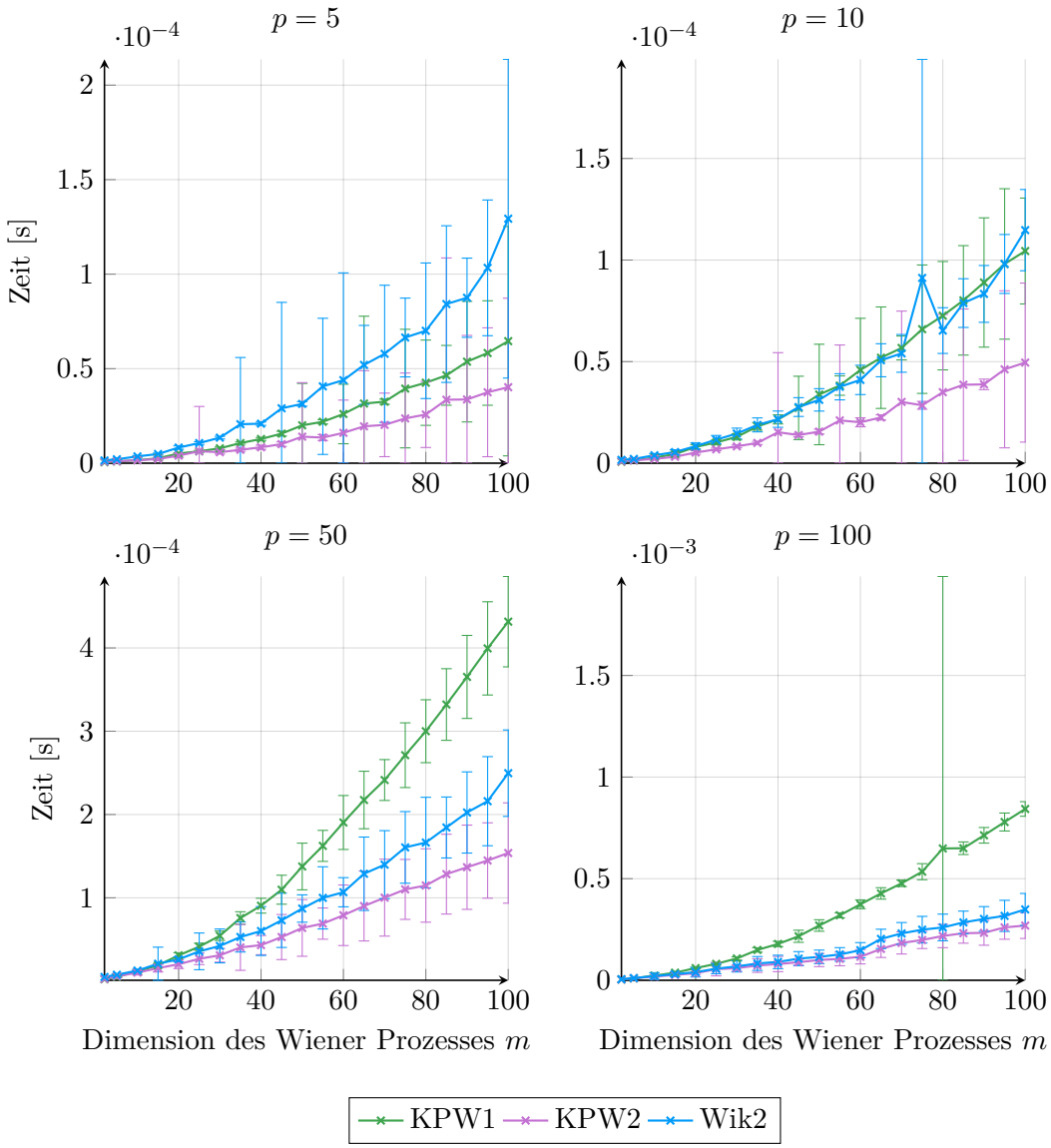

Abbildung 3. Laufzeiten der drei Algorithmen KPW1, KPW2 und Wik2 abhängig von m für verschiedene Werte von p. Die Zeiten wurden gemittelt über 500 Durchläufe.

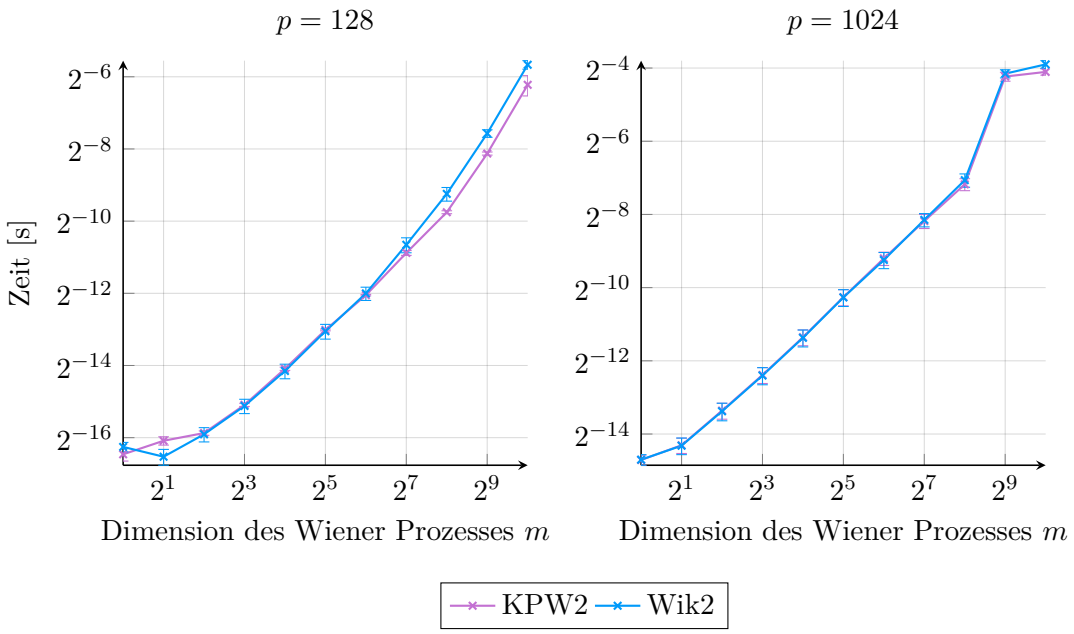

Abbildung 4. Loglog-Plot der Laufzeiten der Algorithmen KPW2 und Wik2 für hochdimensionale Wiener Prozesse. Die Zeiten wurden gemittelt über 500 Durchläufe.

Funktion

$$p_{\texttt{KPW}}(h) = \left\lceil \frac{1}{2\pi^2 h} \right\rceil \tag{3.23}$$

sowie die Funktion

$$p_{\texttt{Wik}}(h) = \left\lceil \sqrt{\frac{5m^2(m-1)}{24\pi^2 h}} \right\rceil \tag{3.24}$$

für verschiedene m eingezeichnet. Diese geben den Abschneideindex für die beiden Algorithmen entsprechend Theorem 11 und Theorem 12 an. Diese Darstellung lässt erkennen, dass der Algorithmus von Wiktorsson nicht von vornherein immer die bessere Wahl ist. Vielmehr ist die Entscheidung abhängig von der Dimension des Wiener Prozesses und der notwendigen Schrittweite des Verfahrens. Die Auswirkungen auf die realen Laufzeiten für KPW2 und Wik2 sind in Abbildung 9 dargestellt.

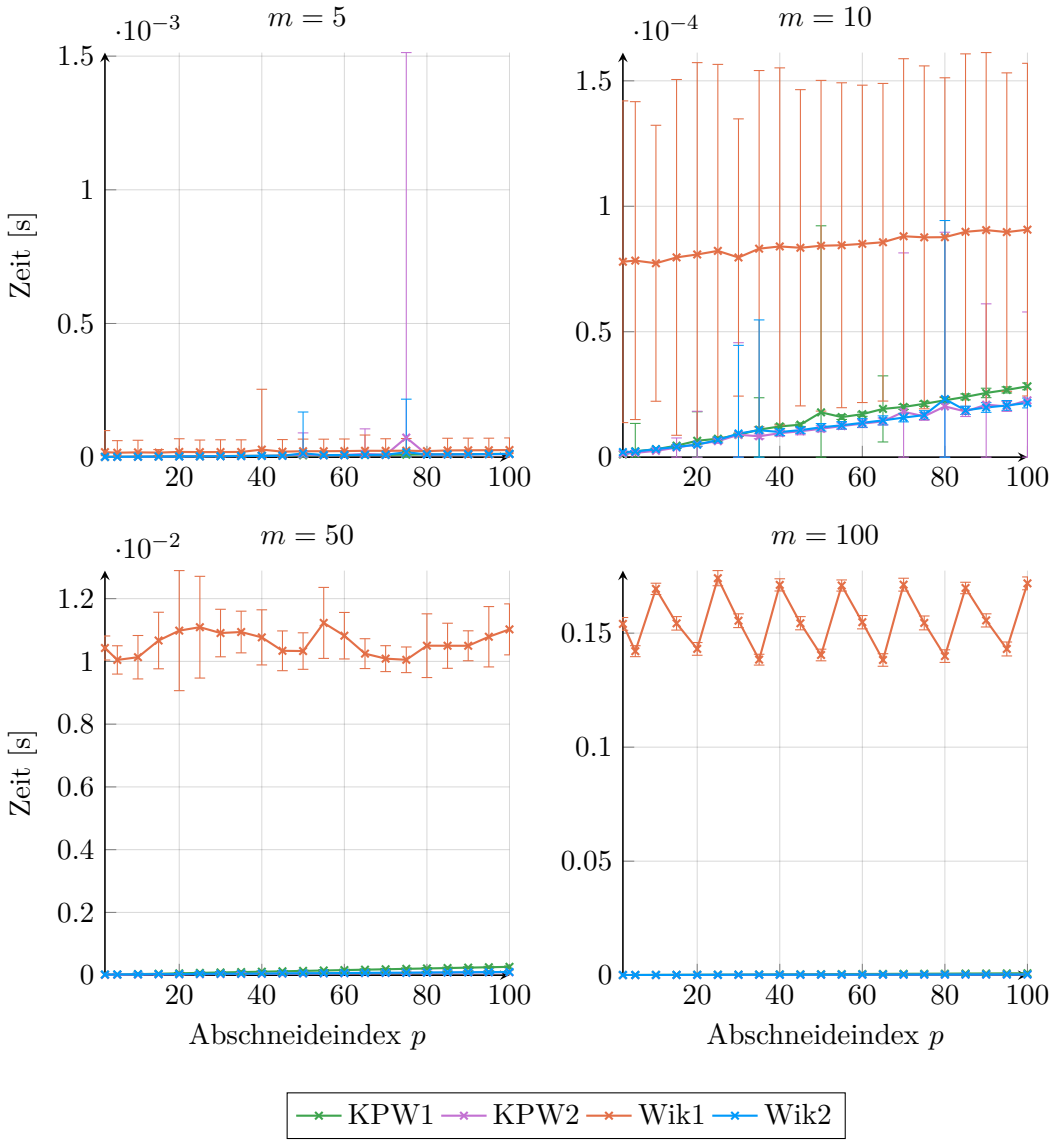

Abbildung 5. Laufzeiten aller vier Algorithmen abhängig von p für verschiedene Werte von m. Die Zeiten wurden gemittelt über 500 Durchläufe.

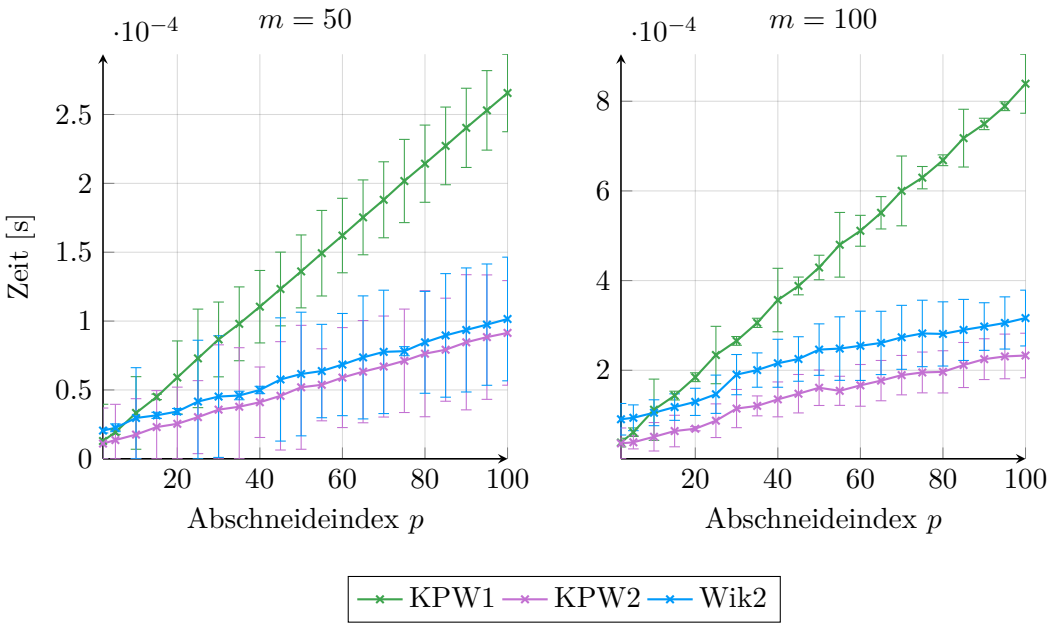

Abbildung 6. Laufzeiten der drei Algorithmen KPW1, KPW2 und Wik2 abhängig von p für verschiedene Werte von m. Die Zeiten wurden gemittelt über 500 Durchläufe.

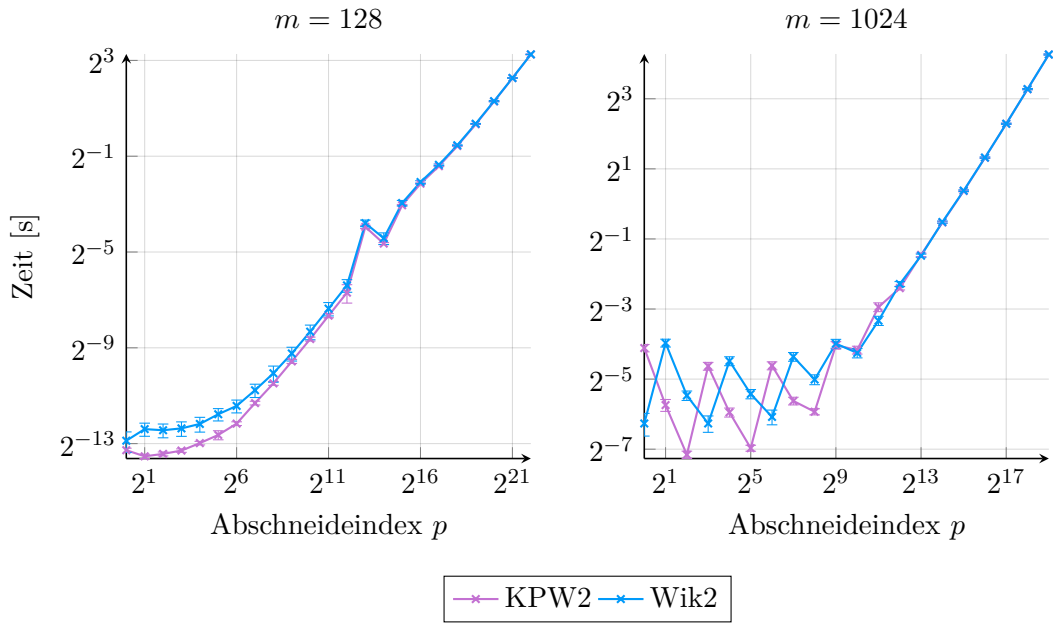

Abbildung 7. Loglog-Plot der Laufzeiten der Algorithmen KPW2 und Wik2 für hohe Werte des Abschneideindex p. Die Zeiten wurden gemittelt über 500 Durchläufe.

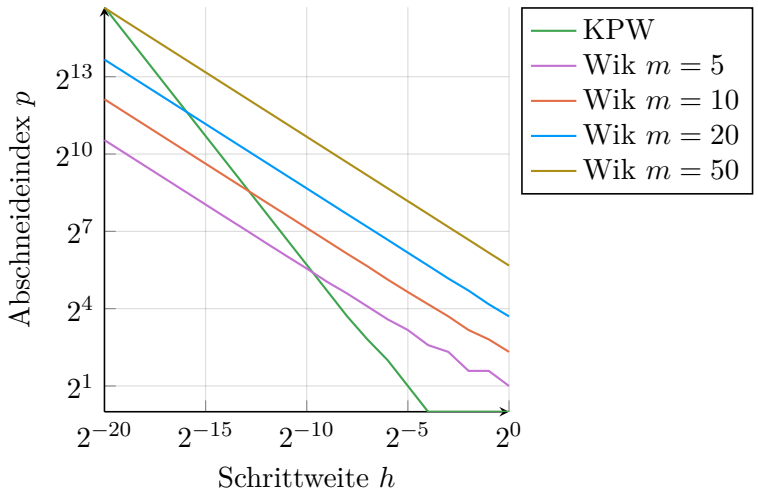

Abbildung 8. Notwendige Anzahl p an Termen in der Approximation in Abhängigkeit von der Schrittweite h (für ein Verfahren der Ordnung $\gamma = 1$). Für den Algorithmus von Wiktorsson ist diese zusätzlich von der Dimension m des Wiener Prozesses abhängig.

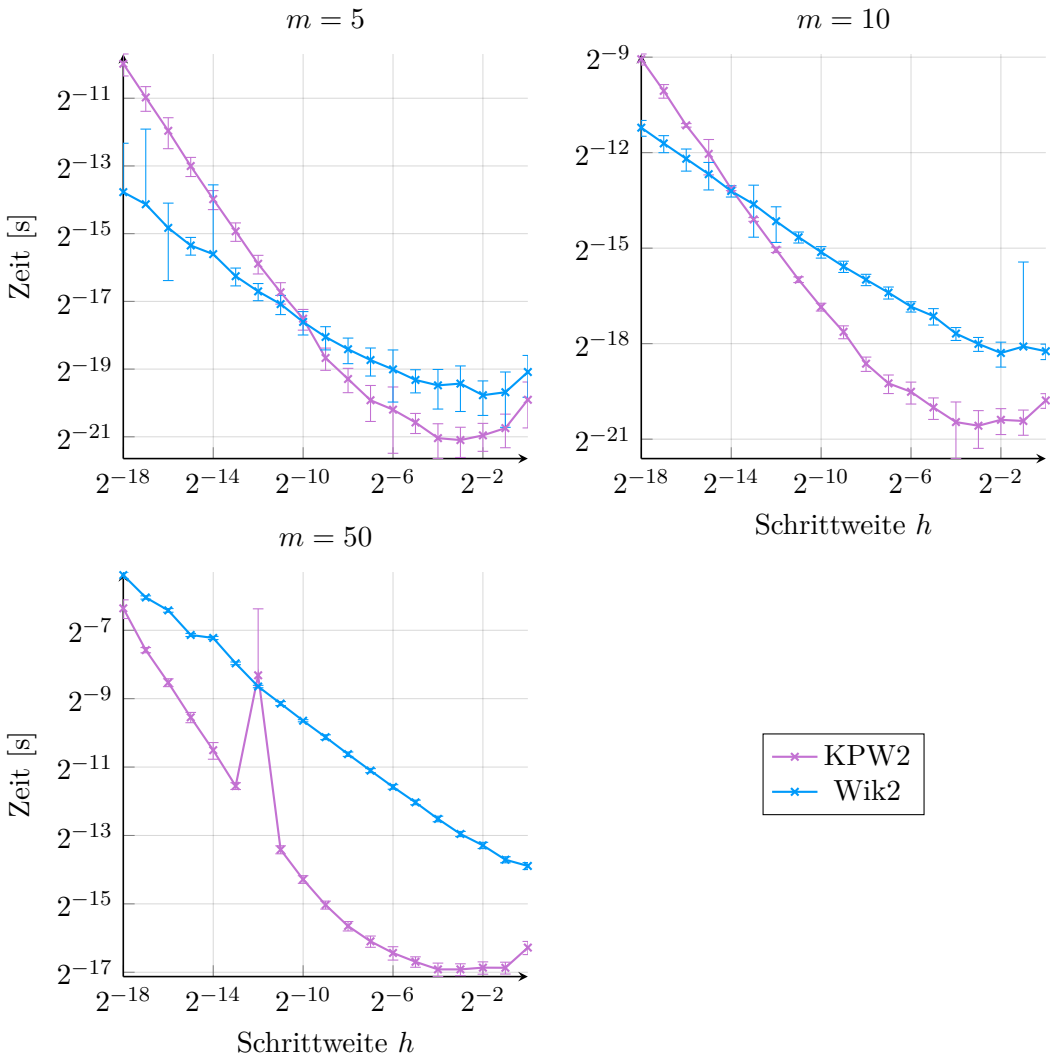

Abbildung 9. Die Laufzeiten der Algorithmen KPW2 und Wik2 in Abhängigkeit von der Schrittweite h. Der Abschneideindex p wurde entsprechend (3.23) bzw. (3.24) gewählt. Die Zeiten wurden gemittelt über 500 Durchläufe.

Kapitel 4: Fazit und Ausblick

In dieser Arbeit wurden doppelt iterierte stochastische Integrale untersucht. Es wurden die Ansätze von Kloeden, Platen und Wright sowie von Wiktorsson zur numerischen Approximation der iterierten Integrale über eine stochastische Fourierreihen-Entwicklung der Brownschen Brücke vorgestellt und in der Programmiersprache Julia effizient implementiert.

Die effiziente Simulation iterierter stochastischer Integrale ist von großer Bedeutung für die numerische Lösung stochastischer Differentialgleichungen. Ein wichtiges Anwendungsfeld ist auch die Lösung stochastischer partieller Differentialgleichungen (SPDEs). SPDEs beinhalten unendlichdimensionale Q-Wiener Prozesse, die zur numerischen Lösung durch hochdimensionale SDEs approximiert werden. Die Effizienzsteigerung in diesen hochdimensionalen Fällen ist enorm. Eine existierende MATLAB-Implementierung des Algorithmus von Wiktorsson benötigt für einen 100-dimensionalen Wiener Prozess mit $p = 10$ ca. 36 s. Im Vergleich dazu benötigt die hier vorgestellte Implementierung Wik2 für dieselben Parameter nur 0.1 ms – eine Verbesserung um einen Faktor 360 000.

Weiterhin sollte man untersuchen, ob sich die in dieser Arbeit vorgestellten Optimierungen übertragen lassen auf die Simulation drei- und mehrfach iterierter Integrale. Diese würden den Weg ebnen, numerische Verfahren höherer Ordnungen zur Lösung von SDEs zu verwenden.

Eine weitere Idee ist, die Brownsche Brücke nicht auf dem Intervall des aktuellen Inkrements des Wiener Prozesses zu simulieren, sondern auf dem gesamten Zeitintervall der betrachteten SDE.

Anhang A: Herleitungen

Wichtig für die Herleitungen in diesem Abschnitt ist folgende Version der partiellen Integration für stetig differenzierbare $f\colon [0,T] \to \mathbb{R}$ und einen eindimensionalen Wiener Prozess $(W_t)_{t\geq 0}$ [KP99, Exercise 3.2.7]:

$$\int_0^T f(t)\,\mathrm{d}W_t = f(T)W_T - \int_0^T f'(t)W_t\,\mathrm{d}t\,. \tag{A.1}$$

A.1 Verteilung der Fourier-Koeffizienten

Die Koeffizienten a_r^i und b_r^i sind lineare Transformationen normalverteilter Zufallsvariablen und damit auch normalverteilt. Mit (A.1) gilt

$$\begin{aligned} a_r^i &= \frac{2}{T}\int_0^T \left(W_s^i - \frac{s}{T}W_T^i\right)\cos\left(\frac{2\pi r}{T}s\right)\,\mathrm{d}s \\ &= \begin{cases} -\dfrac{1}{\pi r}\displaystyle\int_0^T \sin\left(\frac{2\pi r}{T}s\right)\,\mathrm{d}W_s^i\,, & r>0 \\ W_T^i - \dfrac{2}{T}\displaystyle\int_0^T s\,\mathrm{d}W_s^i\,, & r=0\,. \end{cases} \end{aligned} \tag{A.2}$$

Da Itô-Integrale Erwartungswert null haben gilt also $\mathbb{E}(a_r^i)=0$ für alle $r\geq 0$. Für die Varianz im Fall $r>0$ folgt aus den Verteilungseigenschaften des Itô-Integrals

$$\begin{aligned} \mathbb{E}\left(\left(\frac{1}{\pi r}\int_0^T \sin\left(\frac{2\pi r}{T}s\right)\,\mathrm{d}W_s^i\right)^2\right) &= \frac{1}{\pi^2 r^2}\int_0^T \sin^2\left(\frac{2\pi r}{T}s\right)\,\mathrm{d}s \\ &= \frac{T}{2\pi^2 r^2}\,. \end{aligned} \tag{A.3}$$

Im Fall $r=0$ gilt

$$\begin{aligned} \mathbb{E}\left(\left(W_T^i - \frac{2}{T}\int_0^T s\,\mathrm{d}W_s^i\right)^2\right) &= \mathbb{E}\left(\left(W_T^i\right)^2\right) + \frac{4}{T^2}\mathbb{E}\left(\left(\int_0^T s\,\mathrm{d}W_s^i\right)^2\right) \\ &\quad - \frac{4}{T}\mathbb{E}\left(W_T^i\int_0^T s\,\mathrm{d}W_s^i\right) \\ &= T + \frac{4}{T^2}\int_0^T s^2\,\mathrm{d}s - \frac{4}{T}\mathbb{E}\left(\int_0^T \mathrm{d}W_s^i\int_0^T s\,\mathrm{d}W_s^i\right) \\ &= \frac{7}{3}T - \frac{4}{T}\int_0^T s\,\mathrm{d}s \\ &= \frac{1}{3}T\,. \end{aligned} \tag{A.4}$$

Die Verteilung der b_r^i ergibt sich analog zum Fall $r > 0$, sodass insgesamt

$$a_0^i \sim \mathcal{N}(0, \frac{1}{3}T)\,, \tag{A.5}$$

$$a_r^i \sim \mathcal{N}(0, \frac{T}{2\pi^2 r^2}) \quad \text{und} \tag{A.6}$$

$$b_r^i \sim \mathcal{N}(0, \frac{T}{2\pi^2 r^2}) \quad \text{gelten.} \tag{A.7}$$

A.2 Reihendarstellung nach Kloeden, Platen und Wright

Die anschließende Herleitung von (3.10) folgt den Ausführungen in [KPW92]. An (3.9) anknüpfend erhält man

$$\begin{aligned}
\mathcal{I}_{(i,j)}(T) &= \int_0^T W_s^i \,\mathrm{d}W_s^j \\
&= \int_0^T \frac{s}{T} W_T^i + \frac{1}{2}a_0^i + \sum_{r=1}^{\infty}\left(a_r^i \cos\left(\frac{2\pi r}{T}s\right) + b_r^i \sin\left(\frac{2\pi r}{T}s\right)\right) \mathrm{d}W_s^j \\
&= \frac{1}{T}W_T^i \int_0^T s \,\mathrm{d}W_s^j && \text{(A.8a)} \\
&\quad + \frac{1}{2}a_0^i \int_0^T \mathrm{d}W_s^j && \text{(A.8b)} \\
&\quad + \sum_{r=1}^{\infty}\left[a_r^i \int_0^T \cos\left(\frac{2\pi r}{T}s\right) \mathrm{d}W_s^j \right. && \text{(A.8c)} \\
&\qquad \left. + b_r^i \int_0^T \sin\left(\frac{2\pi r}{T}s\right) \mathrm{d}W_s^j\right]. && \text{(A.8d)}
\end{aligned}$$

Für das Integral (A.8b) gilt $\int_0^T \mathrm{d}W_s^j = W_T^j$. Die anderen drei Itô-Integrale werden in den folgenden Abschnitten behandelt.

Zu Ausdruck A.8a

Aus der Definition der Fourier-Koeffizienten (3.6) folgt

$$\begin{aligned}
a_0^i &= \frac{2}{T}\int_0^T W_s^i - \frac{s}{T}W_T^i \,\mathrm{d}s \\
&= \frac{2}{T}\int_0^T W_s^i \,\mathrm{d}s - W_T^i\,.
\end{aligned} \tag{A.9}$$

Für das Integral (A.8a) gilt dann mit (A.1)

$$\begin{aligned}
\int_0^T s \,\mathrm{d}W_s^j &= T \cdot W_T^j - \int_0^T W_s^j \,\mathrm{d}s \\
&= \frac{1}{2}T \cdot (2W_T^j - \frac{2}{T}\int_0^T W_s^j \,\mathrm{d}s) \\
&= \frac{1}{2}T \cdot (W_T^j - a_0^j)\,.
\end{aligned} \tag{A.10}$$

Zu Ausdruck A.8c

Mit (A.1) folgt aus (A.8c) zunächst

$$\int_0^T \cos\left(\frac{2\pi r}{T}s\right)\,\mathrm{d}W_s^j = W_T^j + \frac{2\pi r}{T}\int_0^T \sin\left(\frac{2\pi r}{T}s\right) W_s^j\,\mathrm{d}s\,. \tag{A.11}$$

Durch Einsetzen der Reihendarstellung (3.8) von W_s^j erhält man

$$\begin{aligned}&\int_0^T \sin\left(\frac{2\pi r}{T}s\right) W_s^j\,\mathrm{d}s\\ =&\int_0^T \sin\left(\frac{2\pi r}{T}s\right)\left[\frac{s}{T}W_T^j + \frac{1}{2}a_0^j + \sum_{n=1}^{\infty}\left(a_n^j\cos\left(\frac{2\pi n}{T}s\right) + b_n^j\sin\left(\frac{2\pi n}{T}s\right)\right)\right]\mathrm{d}s\,.\end{aligned} \tag{A.12}$$

Mit den Resultaten

$$\begin{aligned}\int_0^T s\cdot\sin\left(\frac{2\pi r}{T}s\right)\,\mathrm{d}s &= -\frac{T^2}{2\pi r}\,,\\ \int_0^T \sin\left(\frac{2\pi r}{T}s\right)\,\mathrm{d}s &= 0\,,\\ \int_0^T \sin\left(\frac{2\pi r}{T}s\right)\cos\left(\frac{2\pi n}{T}s\right)\,\mathrm{d}s &= 0 \quad \text{und}\\ \int_0^T \sin\left(\frac{2\pi r}{T}s\right)\sin\left(\frac{2\pi n}{T}s\right)\,\mathrm{d}s &= \begin{cases}0 & r\neq n\\ \frac{1}{2}T & r=n\end{cases}\end{aligned}$$

folgt schließlich aus (A.11)

$$\begin{aligned}\int_0^T \cos\left(\frac{2\pi r}{T}s\right)\,\mathrm{d}W_s^j &= W_T^j + \frac{2\pi r}{T}\left[-\frac{T}{2\pi r}W_T^j + \frac{1}{2}Tb_r^j\right]\\ &= \pi r b_r^j\,.\end{aligned} \tag{A.13}$$

Zu Ausdruck A.8d

Zusammen mit

$$\int_0^T s\cdot\cos\left(\frac{2\pi r}{T}s\right)\,\mathrm{d}s = 0 \tag{A.14}$$

ergibt sich für das Integral in (A.8d) analog zum vorherigen Abschnitt

$$\begin{aligned}\int_0^T \sin\left(\frac{2\pi r}{T}s\right)\,\mathrm{d}W_s^j &= -\frac{2\pi r}{T}\int_0^T \cos\left(\frac{2\pi r}{T}s\right) W_s^j\,\mathrm{d}s\\ &= -\frac{2\pi r}{T}\left[\frac{1}{2}Ta_r^j\right]\\ &= -\pi r a_r^j\,.\end{aligned} \tag{A.15}$$

Ergebnis

Einsetzen von (A.10), (A.13) und (A.15) in (A.8) ergibt dann

$$\begin{aligned}\mathcal{I}_{(i,j)}(T) &= \frac{1}{T}W_T^i\left[\frac{1}{2}T\cdot\left(W_T^j - a_0^j\right)\right] + \frac{1}{2}a_0^i\left[W_T^j\right] \\ &\quad + \sum_{r=1}^{\infty}\left(a_r^i\left[\pi r b_r^j\right] + b_r^i\left[-\pi r a_r^j\right]\right) \\ &= \frac{1}{2}W_T^iW_T^j + \frac{1}{2}a_0^iW_T^j - \frac{1}{2}a_0^jW_T^i + \pi\sum_{r=1}^{\infty} r\cdot\left(a_r^ib_r^j - b_r^ia_r^j\right). \end{aligned} \tag{A.16}$$

Setzt man in der Reihendarstellung (3.8) $t = 0$ oder $t = T$, so erhält man $a_0^i = -2\sum_{r=1}^{\infty} a_r^i$ und kann den zweiten und dritten Summanden weiter umformen zu

$$\begin{aligned}\frac{1}{2}a_0^iW_T^j &= -\sum_{r=1}^{\infty} a_r^iW_T^j \\ &= \pi\sum_{r=1}^{\infty} r\cdot a_r^i\left(-\frac{1}{\pi r}W_T^j\right). \end{aligned} \tag{A.17}$$

Schließlich gilt

$$\mathcal{I}_{(i,j)}(T) = \frac{1}{2}W_T^iW_T^j + \pi\sum_{r=1}^{\infty} r\cdot\left(a_r^i\left(b_r^j - \frac{1}{\pi r}W_T^j\right) - \left(b_r^i - \frac{1}{\pi r}W_T^i\right)a_r^j\right). \tag{A.18}$$

Anhang B: Quelltext

Im Folgenden werden die Implementierungen der vier vorgestellten Algorithmen gegeben. Sie wurden in der Programmiersprache Julia [BEK+17] erstellt und getestet. Zusätzlich sind Implementierungen in MATLAB [MAT18] gegeben.

B.1 Julia Code

Es werden die Pakete `LinearAlgebra`, `Random` und `SpecialFunctions` benötigt. Wik1 benötgt zusätzlich `SparseArrays`.

KPW1

```
function kpw1(W, p)
    m = length(W)
    S = zeros(m,m)
    alpha_r = similar(W,m)
    beta_r = similar(W,m)
    for r = 1:p
        randn!(alpha_r)
        randn!(beta_r)
        beta_r .= (beta_r .- sqrt(2).*W) ./ r
        S .+= alpha_r .* beta_r'
    end
    Ip = 0.5.*W.*W' .+ inv(2pi).*(S.-S')
    @inbounds for i=1:m
        Ip[i,i] -= 0.5
    end
    Ip
end
```

KPW2

```
function kpw2(W, p)
    m = length(W)
    beta = randn(m,p)
    beta .= (beta.-sqrt(2).*W) ./ (1:p)'
    S = beta*randn(p,m)
    Ip = 0.5.*W.*W' .+ inv(2pi).*(S.-S')
    @inbounds for i=1:m
        Ip[i,i] -= 0.5
    end
    Ip
end
```

Wik1

```
function wik1(W, p)
    m = length(W)
    M = div(m*(m-1),2)
    beta = randn(m,p)
    beta .= (beta .- sqrt(2) .* W) ./ (1:p)'
    S = beta*randn(p,m)
    P = sparse(1:m^2, [i+j*m for i=1:m for j=0:m-1], 1)
    K = sparse(1:M, [i+(j-1)*m for i=1:m, j=1:m if j<i], 1, M, m^2)
    KIP = K-K*P
    Sigma = 2.0I + 2.0 * KIP * kron(W*W',sparse(1.0I,m,m)) * KIP'
    sqrtSigma = (Sigma + (2*sqrt(1+W'*W))*I) / (sqrt(2)+sqrt(2+2*W'*W))
    R = reshape(sqrt(trigamma(p+1)) * KIP' * sqrtSigma * randn(M), m, m)
    Ip = 0.5 .* W .* W' .+ inv(2pi) .* (S .- S' .+ R)
    @inbounds for i=1:m
        Ip[i,i] -= 0.5
    end
    Ip
end
```

Wik2

```
function wik2(W, p)
    m = length(W)
    S = similar(W,m,m)
    G = similar(W,m,m)
    beta = randn(m,p)
    beta .= (beta.-sqrt(2).*W) ./ (1:p)'
    mul!(S, beta, randn(p,m))
    a = sqrt(2*trigamma(p+1))
    for j = 1:m
        @inbounds G[j,j] = 0.0
        for i = j+1:m
            g = a * randn()
            @inbounds G[i,j] = g
            @inbounds G[j,i] = -g
            @inbounds S[i,j] += g
        end
    end
    S .+= inv(1+sqrt(1+W'*W)) .* W .* (W'*G)
    G .= 0.5.*W.*W' .+ inv(2pi).*(S .- S')
    @inbounds for i=1:m
        G[i,i] -= 0.5
    end
    G
end
```

B.2 Matlab Code

KPW1

```
function Ip = kpw1(W, p)
    m = length(W);
    S = zeros(m,m);
    for r = 1:p
        alpha_r = randn(m,1);
        beta_r = (randn(m,1) - sqrt(2).*W) ./ r;
        S = S + alpha_r.*beta_r';
    end
    Ip = 0.5.*W.*W' + (S-S')/(2*pi) - 0.5*eye(m);
end
```

KPW2

```
function Ip = kpw2(W, p)
    m = length(W);
    alpha = randn(p,m);
    beta = (randn(m,p) - sqrt(2).*W) ./ (1:p);
    S = beta * alpha;
    Ip = 0.5.*W.*W' + (S-S')/(2*pi) - 0.5*eye(m);
end
```

Wik1

```
function Ip = wik1(W,p)
    m = length(W);
    M = m*(m-1)/2;
    I_M = eye(M);
    alpha = randn(p,m);
    beta = (randn(m,p) - sqrt(2).*W) ./ (1:p);
    S = beta * alpha;
    P = sparse(1:m^2,reshape((1:m)+(0:m-1)'*m,m^2,1), 1);
    K = sparse(1:M, find(tril(ones(m),-1)), 1, M, m^2);
    KIP = K-K*P;
    Sigma = 2*I_M + 2*KIP*kron(W*W',speye(m))*KIP';
    sqrtSigma = (Sigma + (2*sqrt(1+norm(W)^2))*I_M) / (sqrt(2)+sqrt(2+2*norm(W)^2));
    R = reshape(sqrt(psi(1,p+1))*KIP'*sqrtSigma*randn(M,1), m, m);
    Ip = 0.5.*W.*W' + (S-S'+R)/(2*pi) - 0.5*eye(m);
end
```

Wik2

```
function G = wik2(W,p)
    m = length(W);
    alpha = randn(p,m);
    beta = (randn(m,p) - sqrt(2).*W) ./ (1:p);
    S = beta * alpha;
    G = zeros(m);
    G(tril(true(m),-1)) = sqrt(2*psi(1,p+1)) .* randn(m*(m-1)/2,1);
    S = S + W.*(W'*(G-G')).*(1+sqrt(1+norm(W)^2)) + G;
    G = 0.5.*W.*W' + (S-S')/(2*pi) - 0.5*eye(m);
end
```

Referenzen

[BEK+17] J. Bezanson, A. Edelman, S. Karpinski und V. B. Shah. „Julia: A Fresh Approach to Numerical Computing“. In: *SIAM review* 59.1 (2017), S. 65–98.

[BLAS] *OpenBLAS*. URL: `https://www.openblas.net`.

[Bro28] R. Brown. „A Brief Account of Microscopical Observations Made in the Months of June, July and August 1827, on the Particles Contained in the Pollen of Plants; and on the General Existence of Active Molecules in Organic and Inorganic Bodies“. In: *The Philosophical Magazine* 4.21 (1828), S. 161–173.

[BS73] F. Black und M. Scholes. „The Pricing of Options and Corporate Liabilities“. In: *Journal of Political Economy* 81.3 (1973), S. 637–654.

[Ein05] A. Einstein. „Über die von der molekularkinetischen Theorie der Wärme geforderte Bewegung von in ruhenden Flüssigkeiten suspendierten Teilchen“. In: *Annalen der Physik* 322.8 (1905), S. 549–560.

[Itô44] K. Itô. „Stochastic Integral“. In: *Proceedings of the Imperial Academy* 20.8 (1944), S. 519–524.

[KP99] P. E. Kloeden und E. Platen. *Numerical Solution of Stochastic Differential Equations*. 3rd. Springer-Verlag, 1999.

[KPW92] P. E. Kloeden, E. Platen und I. W. Wright. „The Approximation of Multiple Stochastic Integrals“. In: *Stochastic Analysis and Applications* 10.4 (1992), S. 431–441.

[KS91] I. Karatzas und S. E. Shreve. *Brownian Motion and Stochastic Calculus*. Bd. 113. Graduate Texts in Mathematics. Springer-Verlag, 1991.

[Mar55] G. Maruyama. „Continuous Markov Processes and Stochastic Equations“. In: *Rendiconti del Circolo Matematico di Palermo* 4.1 (1955), S. 48.

[MAT18] *MATLAB*. Version 9.4.0.813654 (R2018a). The MathWorks Inc. 23. Feb. 2018. URL: `https://www.mathworks.com/products/matlab.html`.

[Mer73] R. C. Merton. „Theory of Rational Option Pricing“. In: *The Bell Journal of Economics and Management Science* 4.1 (1973), S. 141–183.

[Mil74] G. N. Milstein. „Approximate Integration of Stochastic Differential Equations“. In: *Theory of Probability and its Applications* 19.3 (1974), S. 583–588.

[Wie23] N. Wiener. „Differential-Space“. In: *Journal of Mathematics and Physics* 2.1-4 (1923), S. 131–174.

[Wik01] M. Wiktorsson. „Joint Characteristic Function and Simultaneous Simulation of Iterated Itô Integrals for Multiple Independent Brownian Motions“. In: *The Annals of Applied Probability* 11.2 (2001), S. 470–487.