

YUIMA チュートリアル

「確率微分方程式のデータサイエンス入門」資料

R と YUIMA による確率微分方程式入門

小池 祐太

東京大学 数理・情報教育研究センター
大学院数理科学研究科

目次

第 1 章	ブラウン運動とその諸性質	2
1.1	確率過程	2
1.2	ブラウン運動	3
1.3	ブラウン運動の諸性質	4
1.3.1	二次変動の収束	4
1.3.2	最大値の確率分布	7
1.3.3	逆正弦法則	8
1.4	参考: 確率変数列の収束概念	10
第 2 章	確率積分と伊藤の公式	12
2.1	確率積分	12
2.2	後向き伊藤積分・ストラトノヴィッチ積分	15
2.3	伊藤の公式 (ブラウン運動の場合)	17
2.4	確率積分の二次変動と一般の場合の伊藤の公式	21
第 3 章	確率微分方程式とそのシミュレーション	24
3.1	オイラー・丸山法	24
3.2	代表的なモデル	27
3.2.1	幾何ブラウン運動	27
3.2.2	オルンシュタイン・ウーレンベック過程	30
3.2.3	コックス・インガーソル・ロスモデル	34
3.2.4	ヤコビ過程	36
3.2.5	ベッセル過程	37
3.3	確率ボラティリティモデル	39
3.3.1	ヘストンモデル	40
3.3.2	SABR モデル	41
3.4	不変分布とエルゴード性	43
3.5	参考: 確率微分方程式の解の存在と一意性	48

第1章 ブラウン運動とその諸性質

1.1 確率過程

確率変数とは値がランダムに決定される数値のことであった。**確率過程**とは、値がランダムに決定される時刻 t の関数のことである。時刻 t の動く範囲としては閉区間 $[0, T]$ ($T > 0$) もしくは 0 以上の実数全体 $[0, \infty)$ を考える場合が多い。本稿では時刻 t の動く範囲としては $[0, T]$ を考えることにする。

例 1.1. ϵ を確率変数とする。各時刻 $t \in [0, T]$ について

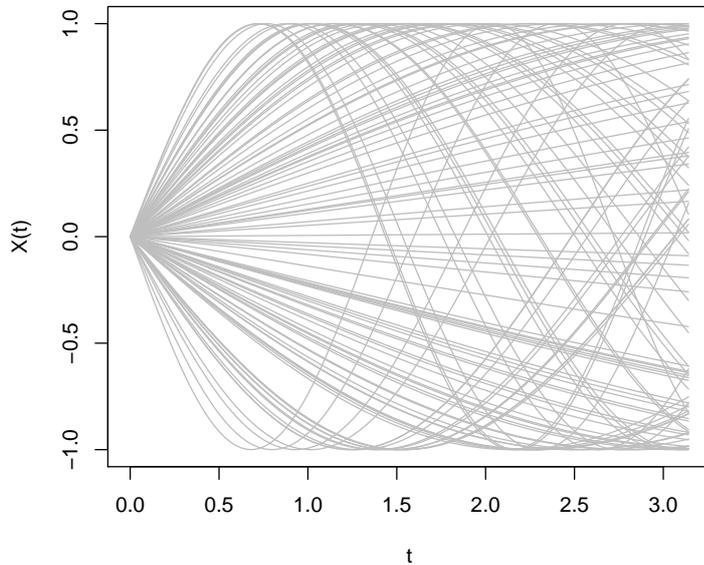
$$X(t) := \sin(\epsilon t)$$

とおくと、値がランダムに決定される時刻 t の関数 $(X(t))_{t \in [0, T]}$ が得られる。このような $(X(t))_{t \in [0, T]}$ が確率過程の一例である。

上の例で示したように、「確率過程 $(X(t))_{t \in [0, T]}$ 」という場合、 $X(t)$ は時点 t で評価した確率過程の関数値を表す。この値はランダムに決定されるから、確率変数である。また、ランダムな決定の結果としてこの確率過程から得られる個々の関数のことを、この確率過程の**サンプルパス**と呼ぶ。確率過程を計算機でシミュレーションする場合、実際には時刻 t を連続的に動かすことは不可能なため、離散的なサンプル時点 $t_0 < t_1 < \dots < t_n$ を時間区間 $[0, T]$ を尽くすよう十分細かく定めて、それらの上で評価したサンプルパスをシミュレーションすることになる。通常、特に理由がない限りサンプル時点は等間隔に取るため、以下ではその場合のみ考える。従って、サンプル時点は $t_i = Ti/n$ ($i = 0, 1, \dots, n$) という形で与えられるとして、区間を分割する個数 n をシミュレーターのパラメーターとして扱う。

例として、以下に例 1.1 で考えた確率過程 $(X(t))_{t \in [0, T]}$ のシミュレーション例を示す ($T = \pi$, $n = 100$ の場合を考えている)。

```
> ### 上の例の確率過程のシミュレーション
> ## シミュレーターの作成
> simulateXt <- function(n){ # nは区間を分割する個数
+   epsilon <- rnorm(1) # 標準正規乱数から生成
+   Time <- seq(0, pi, by = pi/n) # サンプル時点
+   Xt <- sin(epsilon * Time) # XtのサンプルパスをTimeの各点で評価
+   return(Xt)
+ }
> ## モンテカルロ・シミュレーションの実行
> set.seed(123) # 乱数シードの固定
> MC <- 100 # シミュレーション回数
> n <- 100 # 区間の分割数
> result <- replicate(MC, simulateXt(n)) # シミュレーション
> Time <- seq(0, pi, by = pi/n) # サンプル時点
> matplot(Time, result, type = "l", lty = 1, col = "gray",
+         xlab = "t", ylab = "X(t)") # シミュレーションしたサンプルパスの描画
```



1.2 ブラウン運動

定義 1.1 (ブラウン運動). 確率過程 $(B(t))_{t \in [0, T]}$ が**ブラウン運動 (Brownian motion)** であるとは, 以下の4つの条件が満たされることをいう:

- (i) $B(0) = 0$.
- (ii) $(B(t))_{t \in [0, T]}$ のサンプルパスは連続である.¹
- (iii) $0 \leq t_0 < t_1 < \dots < t_n \leq T$ ならば, $B(t_1) - B(t_0), B(t_2) - B(t_1), \dots, B(t_n) - B(t_{n-1})$ は独立な確率変数列となる.
- (iv) $0 \leq s < t \leq T$ ならば, 確率変数 $B(t) - B(s)$ と確率変数 $B(t - s)$ は共通の確率分布をもち, その平均は0, 分散は $t - s$ で与えられる.

上の定義において, 特に条件 (iii) は**独立増分性**, 条件 (iv) は**定常増分性**と呼ばれる.

ブラウン運動は**ウィーナー過程 (Wiener process)** とも呼ばれる. 上の定義のブラウン運動を特に**標準ブラウン運動**と呼ぶことがある.

ブラウン運動の定義では, 一見すると確率変数 $B(t)$ が特定の確率分布に従うことは要求されていないように見える. しかし, 実は上の定義から $B(t)$ は必然的に正規分布に従うことが導かれることが知られている. より正確には次の命題が成り立つ (証明は例えば [11], Theorem 11.4 を参照のこと).

命題 1.1. $(B(t))_{t \in [0, T]}$ をブラウン運動とする. このとき, $0 \leq s < t \leq T$ に対して, 確率変数 $B(t) - B(s)$ は平均0, 分散 $t - s$ の正規分布に従う.

上の事実と独立増分性を使うことで, サンプル時点は $t_i = Ti/n$ ($i = 0, 1, \dots, n$) が与えられた際にブラウン運動のサンプルパスを以下の手順でシミュレーションできる:

1. n 個の標準正規乱数 Z_1, Z_2, \dots, Z_n を発生させる.

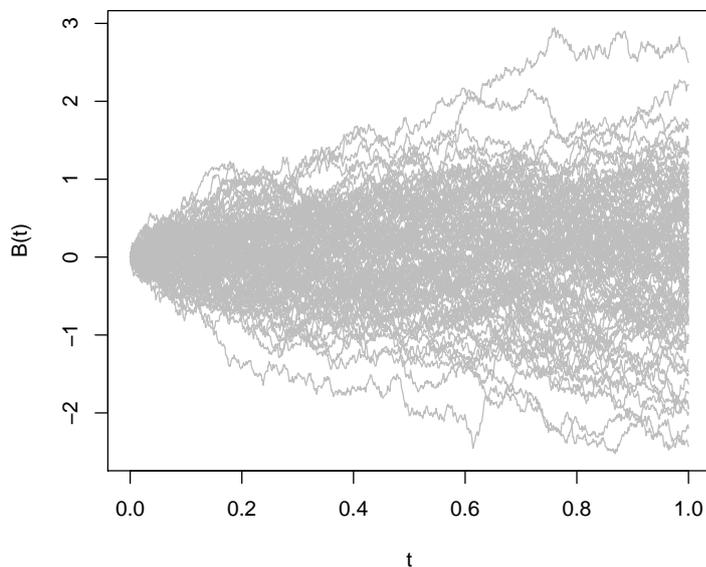
¹厳密には, 「 $(B(t))_{t \in [0, T]}$ のサンプルパスが連続である確率は1である」とすべきである (数学では, 確率1で成立する事象は必ずしも常に成立するわけではない). 以下で述べるサンプルパスに関する性質も, 厳密にはすべて確率1でのみ成立する.

2. $B(t_0) := 0$ とする. また, 各 $i = 1, 2, \dots, n$ について $B(t_i) := (Z_1 + Z_2 + \dots + Z_n) / \sqrt{n}$ とおく.

以下に R でのシミュレーション例を示す.

```
> ### ブラウン運動のシミュレーション
> ## シミュレーターの作成
> ## サンプル時点数nと区間の終端Terminalが与えられたとき,
> ## 区間[0,T]上のブラウン運動をシミュレーションする
> simulateBM <- function(n, Terminal = 1){ # Terminalは既定値を1に設定
+   Z <- rnorm(n) # 標準正規乱数の生成
+   Bt <- diffinv(Z)/sqrt(n) # Btのサンプルパス
+   return(Bt)
+ }
> ## モンテカルロ・シミュレーションの実行
> set.seed(123) # 乱数シードの固定
> MC <- 100 # シミュレーション回数
> n <- 1000 # 区間の分割数
> result <- replicate(MC, simulateBM(n)) # シミュレーション
> Time <- seq(0, 1, by = 1/n) # サンプル時点
> matplot(Time, result, type = "l", lty = 1, col = "gray",
+         xlab = "t", ylab = "B(t)") # シミュレーションしたサンプルパスの描画
```

(bm.r)



1.3 ブラウン運動の諸性質

1.3.1 二次変動の収束

ブラウン運動のサンプルパスは定義より連続ではあるが, 前節のプロットからも見てとれるように, 非常に激しく変動する. そのため, 微分積分学で通常扱うような関数をもつ「良い」性質を持たない. 例えば, ブラウン運動のサンプルパスはいたるところ微分不可能であることが知られている.² 別の「悪い」性質とし

²例えば [21] の定理 1.5.7 を参照のこと.

て、以下で説明するように、ブラウン運動のサンプルパスは有界変動性をもたない。特にこの事実は、次章で見ると、「ブラウン運動に関する積分」を考える際のボトルネックとなる。他方、ブラウン運動のサンプルパスの「良い」性質として、二次変動が収束するという事実がある。この事実が上述の「ブラウン運動に関する積分」を考える上で重要な役割を果たす。

命題 1.2 (二次変動の収束). $(B(t))_{t \in [0, T]}$ をブラウン運動とする。また、 $(\Delta_n)_{n \in \mathbb{N}}$ を $n \rightarrow \infty$ のとき 0 に収束する正数列とし、 $t_{n,i} := i\Delta_n, i = 0, 1, \dots$, とおく。このとき、すべての $t \in [0, T]$ について

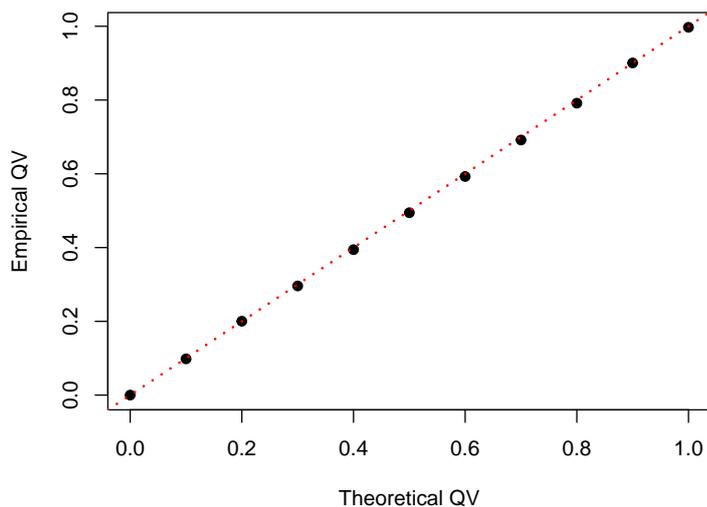
$$\lim_{n \rightarrow \infty} \sum_{i: t_{n,i} \leq t} (B(t_{n,i}) - B(t_{n,i-1}))^2 = t \quad (1.1)$$

が成り立つ。³

証明は例えば [21], 定理 1.5.9 を参照のこと。ここでは命題 1.2 の成立を数値シミュレーションによって観察する。

```
> ### 二次変動の収束(T = 1)
> Time.qv <- seq(0, 1, by = 0.1) # 二次変動の評価時点
> n <- 10000 # ブラウン運動のサンプル時点数
> # (二次変動の評価時点より細かく取っておく)
> set.seed(123) # 乱数シードの固定
> B <- simulateBM(n) # Brown運動のシミュレーション
> qv <- diffinv(diff(B)^2) # 二次変動の計算
> qv <- qv[seq(1, length(qv), by = 1000)] # 評価時点のみ残す
> plot(qv ~ Time.qv, pch = 19, xlab = "Theoretical QV",
+      ylab = "Empirical QV") # 理論値 vs 観測値の散布図
> abline(0, 1, lty = "dotted", col = "red", lwd = 2) # 45度線
```

(qv.r)



³厳密には、極限 (1.1) は確率収束極限を意味する (確率収束の定義は 1.4 節を参照)。さらに条件 $\{t_{n,i} \mid i = 0, 1, \dots\} \subset \{t_{n+1,i} \mid i = 0, 1, \dots\}$ ($n = 1, 2, \dots$) が成立するならば、この収束は確率 1 で成立することも知られている ([6], 定理 2.3 系)。以下特に断らない限り、極限はすべて確率収束極限を考えている。

サンプルパスの非有界変動性

命題 1.2 から, 特に $t > 0$ のとき

$$\lim_{n \rightarrow \infty} \sum_{i: t_{n,i} \leq t} |B(t_{n,i}) - B(t_{n,i-1})| = \infty$$

となることもわかる. 実際, もし $\sum_{i: t_{n,i} \leq t} |B(t_{n,i}) - B(t_{n,i-1})|$ が $n \rightarrow \infty$ のとき無限大に発散しなければ,

$$\sum_{i: t_{n,i} \leq t} (B(t_{n,i}) - B(t_{n,i-1}))^2 \leq \max_{i: t_{n,i} \leq t} |B(t_{n,i}) - B(t_{n,i-1})| \sum_{i: t_{n,i} \leq t} |B(t_{n,i}) - B(t_{n,i-1})|$$

が成り立つことと, ブラウン運動のサンプルパスが連続であることから

$$\lim_{n \rightarrow \infty} \max_{i: t_{n,i} \leq t} |B(t_{n,i}) - B(t_{n,i-1})| = 0$$

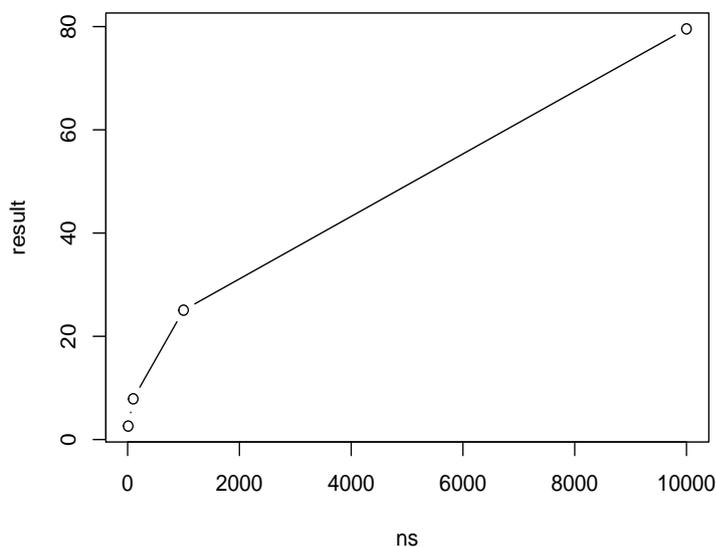
となることに注意すると, はさみうちの原理より

$$\lim_{n \rightarrow \infty} \sum_{i: t_{n,i} \leq t} (B(t_{n,i}) - B(t_{n,i-1}))^2 = 0$$

となって命題 1.2 の結論と矛盾するからである.

```
> ### ブラウン運動のサンプルパスの非有界変動性(T = 1)
> ## ブラウン運動のサンプルパスのシミュレーション
> set.seed(123) # 乱数シードの固定
> n <- 10000 # ブラウン運動のサンプル時点数
> B <- simulateBM(n) # サンプルパスのシミュレーション
> ## サブサンプルしたデータの全変動を計算してプロット
> ns <- 10^(1:4) # サブサンプリングする時点数(10,100,1000,10000)
> result <- double(length(ns)) # 結果を保持するためのベクトル
> for(i in 1:length(ns)){
+   subB <- B[seq(1, n+1, length = ns[i]+1)] # サブサンプリング
+   result[i] <- sum(abs(diff(subB))) # 全変動の計算
+ }
> plot(ns, result, type = "b") # プロット
```

(tv.r)



1.3.2 最大値の確率分布

ブラウン運動 $(B(t))_{t \in [0, T]}$ の最大値

$$M(T) = \max_{0 \leq t \leq T} B(t)$$

の従う確率分布は以下のように明示的に与えることができる:

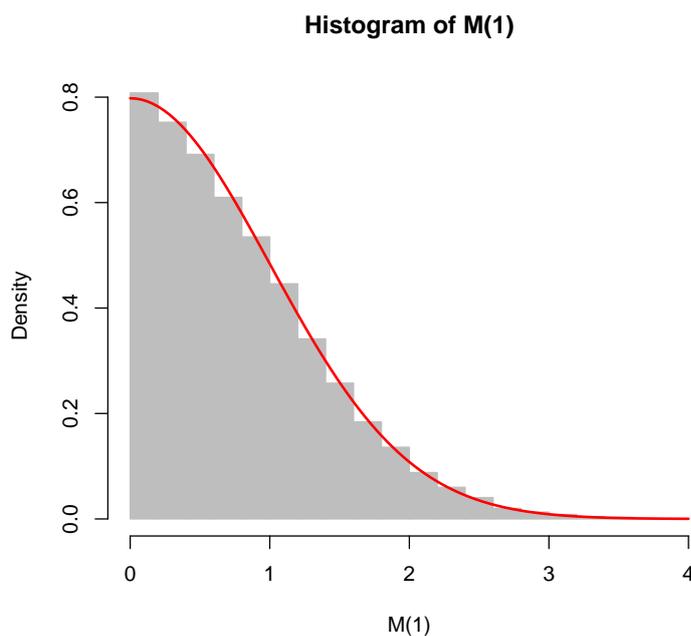
命題 1.3 (ブラウン運動の最大値の分布). $M(T)$ は $|B(T)|$ と同じ確率分布に従う. 特に, $M(T)$ の分布の確率密度関数は以下で与えられる:

$$f_{M(T)}(x) = \sqrt{\frac{2}{\pi T}} e^{-x^2/(2T)}, \quad x > 0.$$

証明は [6], 命題 2.9 を参照のこと. R でのシミュレーション例は以下のとおりである.

```
> ### ブラウン運動の最大値の確率分布(T = 1)
> ## シミュレーターの作成
> maxBM <- function(n){ # nはブラウン運動のサンプル時点数
+   B <- simulateBM(n) # ブラウン運動のシミュレーション
+   return(max(B))
+ }
> ## モンテカルロ・シミュレーションの実行
> set.seed(123) # 乱数シードの固定
> MC <- 10000 # シミュレーション回数
> n <- 10000 # サンプル時点数
> M <- replicate(MC, maxBM(n)) # シミュレーション
> ## ヒストグラムの描画
> hist(M, freq = FALSE, breaks = 25, col = "gray",
+   border = "gray", main = "Histogram of M(1)", xlab = "M(1)")
> f <- function(x) sqrt(2/pi) * exp(-x^2/2) # 理論上の確率密度関数
> curve(f, add = TRUE, col="red", lwd = 2) # ヒストグラムに上書き
```

(maxBM.r)



1.3.3 逆正弦法則

前節ではブラウン運動のサンプルパスの最大値が従う確率分布を確認したが、実はブラウン運動のサンプルパスが最大値に到達する時刻が従う確率分布の解析的表現も知られており、本節ではこの結果と関連する結果について説明する。

ブラウン運動 $(B(t))_{t \in [0, T]}$ のサンプルパスは、時間区間 $[0, T]$ においてただ一度だけその最大値 $M(T) = \max_{0 \leq t \leq T} B(t)$ に到達することが知られているため、その到達時刻を θ_T と書くことにする。このとき、次の結果が成り立つことが知られている⁴:

命題 1.4 (最大値への到達時刻に関する逆正弦法則). 確率変数 θ_T の累積分布関数は以下の逆正弦法則で与えられる:

$$P(\theta_T \leq x) = \frac{2}{\pi} \arcsin \sqrt{\frac{x}{T}}, \quad x \in [0, T].$$

特に、 θ_T の分布の確率密度関数は次式で与えられる:

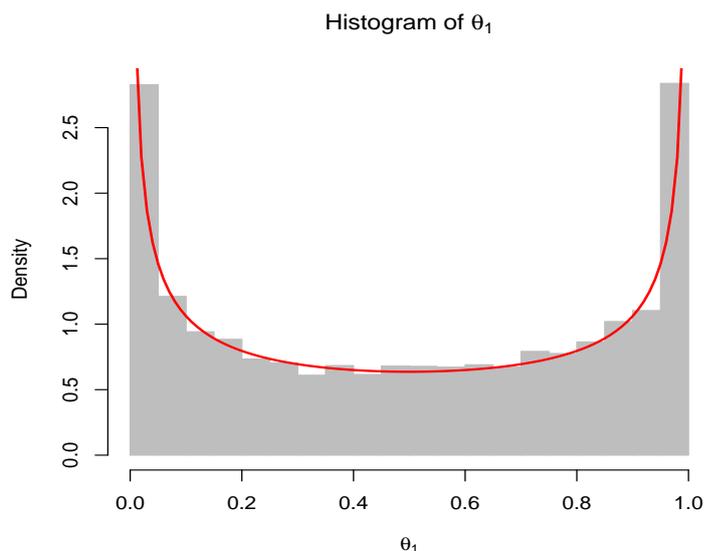
$$f_{\theta_T}(x) = \frac{1}{\pi \sqrt{x(T-x)}}, \quad x \in (0, T).$$

以下に上記の逆正弦法則のシミュレーション例を示す。

```
> ### 逆正弦法則(最大値への到達時刻)
> ## シミュレーターの作成(T = 1)
> hitmax <- function(n){ # nはブラウン運動のサンプル時点数
+   B <- simulateBM(n) # ブラウン運動のシミュレーション
+   return(Time[which.max(B)])
+ }
> ## モンテカルロ・シミュレーションの実行
> set.seed(123) # 乱数シードの固定
> MC <- 10000 # シミュレーション回数
> n <- 1000 # サンプル時点数
> theta <- replicate(MC, hitmax(n)) # シミュレーション
> ## ヒストグラムの描画
> hist(theta, freq = FALSE, breaks = 25, col = "gray",
+   border = "gray", xlab = expression(theta[1]),
+   main = expression(paste("Histogram of ", theta[1])))
> # 理論上の確率密度関数
> # (いまの場合パラメーター(0.5, 0.5)のベータ分布の確率密度関数)
> f <- function(x) dbeta(x, 0.5, 0.5)
> curve(f, add = TRUE, col="red", lwd = 2) # ヒストグラムに上書き
```

(arcsin.r)

⁴[12], 第2章問題 8.17, もしくは [11], Theorem 11.16 を参照のこと.



逆正弦法則はブラウン運動に関する別の統計量の確率分布としても現れる. 一例としてブラウン運動の正の領域での滞在時間を考える. ブラウン運動 $(B(t))_{t \in [0, T]}$ のサンプルパスが正の値を取る時間帯の総時間数を Φ_T と書くことにすると, 次の結果が成り立つ⁵:

命題 1.5 (正の領域での滞在時間に関する逆正弦法則). 確率変数 Φ_T の累積分布関数は以下の逆正弦法則で与えられる:

$$P(\Phi_T \leq x) = \frac{2}{\pi} \arcsin \sqrt{\frac{x}{T}}, \quad x \in [0, T].$$

特に, Φ_T の分布の確率密度関数は次式で与えられる:

$$f_{\Phi_T}(x) = \frac{1}{\pi \sqrt{x(T-x)}}, \quad x \in (0, T).$$

ブラウン運動がしばしば「公平な賭けを繰り返した際の収益の時間発展」を表すためのモデルとして利用されることを鑑みると, 上の結果は興味深い示唆を与える. 実際, $f_{\Phi_T}(x)$ の値は x が 0 または T に近いほど大きくなるから, これは Φ_T は 0 もしくは T 付近の値をとる確率が高いことを示唆する. すなわち, ブラウン運動のサンプルパスは「ほとんどの時間正である」もしくは「ほとんどの時間負である」のいずれかである確率が高いことになる. ブラウン運動を「公平な賭けを繰り返した際の収益の時間発展」とみなした場合, 上の事実は「ほとんどの時間で収益が正である」もしくは「ほとんどの時間で収益が負である」という状況が起きる確率が高いことを示唆しており, 「収益が正である時間と負である時間が半々である」という状況が起きる確率は実は低いことになる. シミュレーションで確認するには例えば以下のようにすればよい.

```
> ### 逆正弦法則(正の領域での滞在時間)
> ## シミュレーターの作成(T = 1)
> positive.time <- function(Time){ # nはブラウン運動のサンプル時点
+   B <- simulateBM(n) # ブラウン運動のシミュレーション
+   return(sum((B[-1] > 0) * 1/n)) # B(t) > 0であるような区間幅を合計
+ }
> ## モンテカルロ・シミュレーションの実行
> set.seed(123) # 乱数シードの固定
> MC <- 10000 # シミュレーション回数
> n <- 1000 # サンプル時点数
> Phi <- replicate(MC, positive.time(n)) # シミュレーション
```

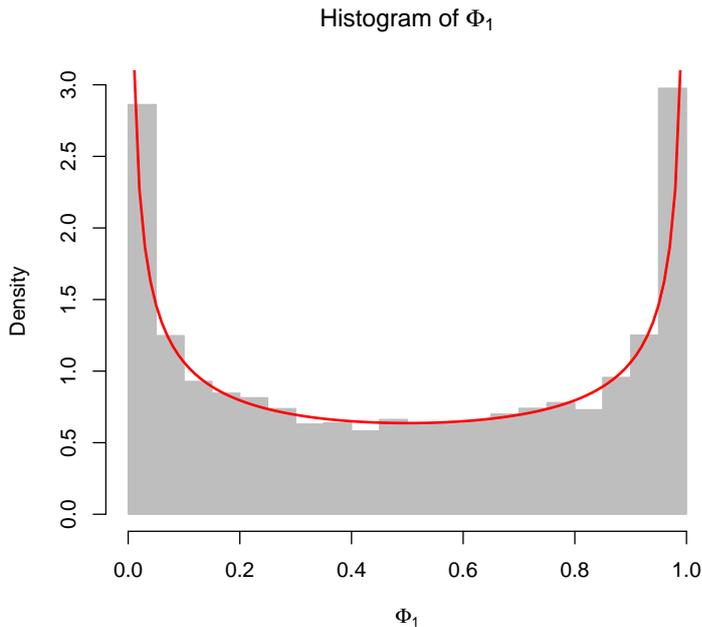
⁵[12], 第 4 章命題 4.11, もしくは [11], Theorem 11.16 を参照のこと.

```

> ## ヒストグラムの描画
> hist(Phi, freq = FALSE, breaks = 25, col = "gray",
+     border = "gray", xlab = expression(Phi[1]),
+     main = expression(paste("Histogram of ", Phi[1])))
> # 理論上の確率密度関数
> # (いまの場合パラメーター(0.5, 0.5)のベータ分布の確率密度関数)
> f <- function(x) dbeta(x, 0.5, 0.5)
> curve(f, add = TRUE, col="red", lwd = 2) # ヒストグラムに上書き

```

(positive.r)



1.4 参考: 確率変数列の収束概念

この節では、確率変数列の収束概念に関する補足を述べる。確率変数列の収束には様々な種類があり、用途によって使い分ける必要があるのだが、ここでは定義だけを述べ、詳細については専門書に譲ることにする。例えば [20] の 1.7 節を参照のこと。

$(X_n)_{n=1}^{\infty}$ を確率変数列, X を (極限として現れる) 確率変数とする。

定義 1.2 (概収束). $(X_n)_{n=1}^{\infty}$ が X に**概収束**するとは、「 $(X_n)_{n=1}^{\infty}$ が X に収束する」という事象が確率 1 で起きることをいう。すなわち、

$$P\left(\lim_{n \rightarrow \infty} X_n = X\right) = 1$$

が成り立つことをいう。

ここで述べた「概収束」は、「確率変数列の収束」といった場合におそらく最も直感にあう概念だと思われる。しかし、概収束は技術的に扱いにくい場合が多々あるため、そのような場合には条件を少し緩めた次の収束概念が用いられる:

定義 1.3 (確率収束). $(X_n)_{n=1}^{\infty}$ が X に**確率収束**するとは、任意の $\varepsilon > 0$ に対して、「 $|X_n - X| > \varepsilon$ となる」という事象が起きる確率が $n \rightarrow \infty$ のとき 0 に収束することをいう。すなわち、

$$\lim_{n \rightarrow \infty} P(|X_n - X| > \varepsilon) = 0$$

が成り立つことをいう。

統計学では、推定量がサンプル数を大きくするに従って推定したいパラメーターに近づいていくことを保証するために、概収束や確率収束といった概念が用いられる。

他によく現れる概念として以下のものがある：

定義 1.4 (平均収束). $p > 0$ とする. $(X_n)_{n=1}^{\infty}$ が X に p 次平均収束するとは、

$$\lim_{n \rightarrow \infty} E[|X_n - X|^p] = 0$$

が成り立つことをいう. p 次平均収束のことを L^p -収束ともいう。

定義 1.5 (分布収束). $(X_n)_{n=1}^{\infty}$ が X に分布収束するとは、任意の有界連続関数 $f(x)$ に対して

$$\lim_{n \rightarrow \infty} E[f(X_n)] = E[f(X)]$$

が成り立つことをいう. 分布収束のことを法則収束ともいう。

p 次平均収束すれば確率収束も言えるという意味で、 p 次平均収束は確率収束よりも強い意味での収束概念であるが、 p 次平均収束と概収束の間には、一般には強弱関係は存在しない。一方で、分布収束は確率収束よりも弱い収束概念であり、統計学では統計量の誤差の分布が別の (よく知られた) 分布で近似できることを主張する際に用いられる。

第2章 確率積分と伊藤の公式

本章では、伊藤清によって創出された、確率過程を解析するための代表的な手法である確率積分の理論(伊藤解析)を概観し、いくつかの結果をシミュレーションによって確認する。数学的結果を厳密に定式化・証明することは本稿の目的ではないため、そちらに関心がある場合は確率解析の専門書を参照してほしい。日本語で読める専門書としてはカラザス・シュレーブ [12]、谷口・松本 [22] や渡辺 [24] などがある。¹ 洋書の代表的文献としては Ikeda & Watanabe [8] や Revuz & Yor [19] などがある。

以下では $(B(t))_{t \in [0, T]}$ はブラウン運動を表すとする。

2.1 確率積分

$(X(t))_{t \in [0, T]}$ と $(Y(t))_{t \in [0, T]}$ を2つの確率過程とし、両者とも連続なサンプルパスを持つとする。この節の目的は、「積分」

$$\int_0^t X(s) dY(s), \quad t \in [0, T] \quad (2.1)$$

に適切な定義を与えることである。いまもし $(Y(t))_{t \in [0, T]}$ のサンプルパスが有界変動をもつならば、微分積分学でよく知られているように、 $(\Delta_n)_{n \in \mathbb{N}}$ を $n \rightarrow \infty$ のとき0に収束する正数列とし、 $t_{n,i} := i\Delta_n, i = 0, 1, \dots$ とおくと、リーマン和

$$\sum_{i: t_{n,i} \leq t} X(t_{n,i-1})(Y(t_{n,i}) - Y(t_{n,i-1}))$$

は $n \rightarrow \infty$ のとき極限をもち、この極限をもって積分 (2.1) を定義することができる。ここでの我々の目標は、 $(Y(t))_{t \in [0, T]}$ としてブラウン運動 $(B(t))_{t \in [0, T]}$ を考えた場合にこの定義の類推を与えることである。前章で観察したように、ブラウン運動のサンプルパスは有界変動を持たないので、リーマン和

$$\sum_{i: t_{n,i} \leq t} X(t_{n,i-1})(B(t_{n,i}) - B(t_{n,i-1})) \quad (2.2)$$

が極限を持つとは限らない。しかし、「被積分過程」 $(X(t))_{t \in [0, T]}$ に適当な制約を課すことで、リーマン和 (2.2) の極限の存在を保証することができ、この極限をもって積分

$$\int_0^t X(s) dB(s) \quad (2.3)$$

を定義することができる。このようにして定義される積分 (2.3) を B に関する X の**確率積分**もしくは**伊藤積分**と呼ぶ。

リーマン和 (2.2) の極限の存在を保証するような被積分過程 $(X(t))_{t \in [0, T]}$ への制約はいくつか知られているが、応用上最も便利なものに次の条件がある：

- 各時点 $t \in [0, T]$ において、確率変数 $X(t)$ は時点 t までのブラウン運動のサンプルパスの関数となる：

$$X(t) = f((B(s))_{s \in [0, t]}).$$

ここに、 f は区間 $[0, t]$ 上の関数を変数とする関数で、適切な条件を満たすものである。²

¹ 渡辺 [24] は文庫版がちくま学芸文庫から出版されている。

² 「適切な条件」とは関数 f の可測性に関するものであるが、詳細は非常に技術的となるためここでは割愛する。

上の条件を満たすような確率過程 $(X(t))_{t \in [0, T]}$ は**適合過程**であるという。³

適合過程の代表例は、ある連続関数 $f(x)$ によって $X(t) = f(B(t))$ ($t \in [0, T]$) と書ける場合である。従ってこの場合はリーマン和 (2.2) が極限をもち、確率積分

$$\int_0^t f(B(s))dB(s) \quad (2.4)$$

が定義される。以下に、 $f(x) = x, t = 1$ の場合に確率積分 (2.4) に対応するリーマン和が実際に収束することを確認するコードを示す。なお、この場合、後述する伊藤の公式から

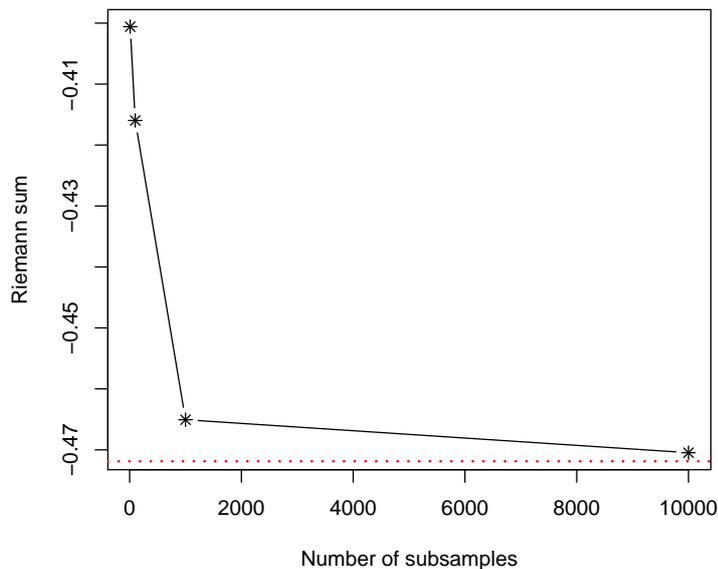
$$\int_0^1 B(s)dB(s) = \frac{B(1)^2 - 1}{2}$$

が成り立つため、リーマン和は $(B(1)^2 - 1)/2$ に収束するはずである。

```
> ### 確率積分を定義するリーマン和の収束
> ## ブラウン運動のシミュレーション
> set.seed(123) # 乱数シードの固定
> n <- 10000 # サンプル時点数
> B <- simulateBM(n) # サンプルパスのシミュレーション
> ## サブサンプルしたデータのリーマン和を計算してプロット
> ns <- 10^(1:4) # サブサンプリングする時点数(10,100,1000,10000)
> result <- double(length(ns)) # 結果を保持するためのベクトル
> for(i in 1:length(ns)){
+   subB <- B[seq(1, n+1, length = ns[i]+1)] # サブサンプリング
+   result[i] <- sum(subB[1:ns[i]] * diff(subB)) # リーマン和の計算
+ }
> plot(ns, result, type = "b", pch = 8,
+       xlab = "Number of subsamples",
+       ylab = "Riemann sum") # プロット
> abline(h = (B[length(B)]^2 - 1)/2, col = "red",
+        lty = "dotted", lwd = 2) # 理論値
```

(riemann.r)

³厳密には、「 $(B(t))_{t \in [0, T]}$ によって生成されるフィルトレーションに関して」適合過程であるという。



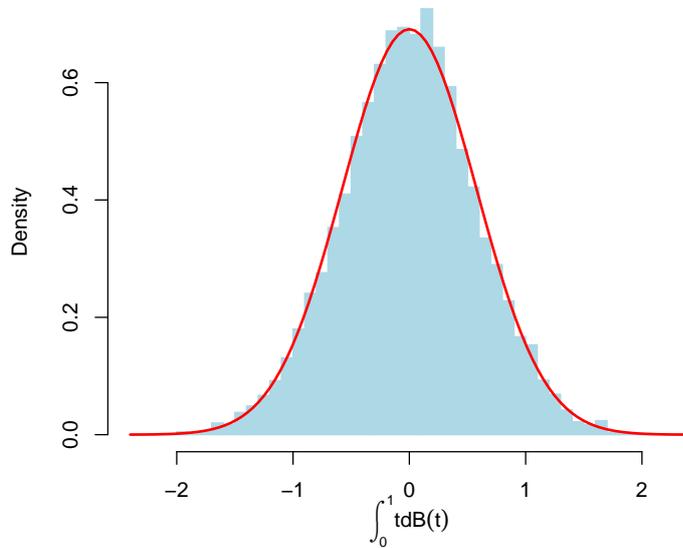
別の代表的な適合過程の例として, $(X(t))_{t \in [0, T]}$ が非ランダムな場合が挙げられる. この場合の確率積分 (2.3) を特に**ウィーナー積分 (Wiener integral)**と呼ぶ場合がある. この場合は確率積分 (2.3) は正規分布に従うことが知られている:

命題 2.1. $(X(t))_{t \in [0, T]}$ は非ランダムであるとする. このとき, 確率変数 $\int_0^t X(s)dB(s)$ は平均 0, 分散 $\int_0^t X(s)^2 ds$ の正規分布に従う.

ウィーナー積分 $\int_0^1 t dB(t)$ が実際に平均 0, 分散 $\int_0^1 t^2 dt = 1/3$ の正規分布に従うことをシミュレーションで確認してみる.

```
> ### ウィーナー積分が正規分布に従うことの確認
> ## シミュレーターの作成
> # ウィーナー積分を離散化する時点数nを引数として,
> # 対応するリーマン和をシミュレート
> wiener.integral <- function(n){
+   B <- simulateBM(n) # ブラウン運動のシミュレーション
+   return(sum(seq(0, 1 - 1/n, by = 1/n) * diff(B)))
+ }
> ## モンテカルロ・シミュレーションの実行
> set.seed(123) # 乱数シードの固定
> MC <- 10000 # シミュレーション回数
> n <- 1000 # サンプル時点数
> result <- replicate(MC, wiener.integral(n)) # シミュレーション
> ## ヒストグラムの描画
> hist(result, freq = FALSE, breaks = 50, main = "",
+       xlab = expression(integral(t*dB(t),0,1)),
+       col = "lightblue", border = "lightblue")
> curve(dnorm(x, sd = 1/sqrt(3)), add = TRUE, col="red",
+       lwd = 2) # 理論上の確率密度関数
```

(wiener-int.r)



最後に, $(X(t))_{t \in [0, T]}$ が適合過程であれば確率積分 (2.3) は定義されるが, この場合

$$Z(t) = \int_0^t X(s)dB(s), \quad t \in [0, T]$$

とおくことで新たな確率過程 $(Z(t))_{t \in [0, T]}$ が定義できる. この確率過程 $(Z(t))_{t \in [0, T]}$ も適合過程となり, かつ連続なサンプルパスをもつことに注意しておく. 従って再び確率積分

$$\int_0^t Z(s)dB(s), \quad t \in [0, T]$$

を定義することができる.

2.2 後向き伊藤積分・ストラトノヴィッチ積分

前節に引き続き, $(X(t))_{t \in [0, T]}$ と $(Y(t))_{t \in [0, T]}$ を 2 つの確率過程とし, 両者とも連続なサンプルパスを持つとする. また, $(\Delta_n)_{n \in \mathbb{N}}$ を $n \rightarrow \infty$ のとき 0 に収束する正数列とし, $t_{n,i} := i\Delta_n$, $i = 0, 1, \dots$, とおく. $(Y(t))_{t \in [0, T]}$ のサンプルパスが有界変動をもつ場合, 微分積分学においてよく知られているように, 時点 $\tau_{n,1}, \tau_{n,2}, \dots$ を

$$t_{n,i-1} \leq \tau_{n,i} \leq t_{n,i} \quad i = 1, 2, \dots$$

を満たすようにとりさえすれば, リーマン和

$$\sum_{i: t_{n,i} \leq t} X(\tau_{n,i})(Y(t_{n,i}) - Y(t_{n,i-1}))$$

は $n \rightarrow \infty$ のとき積分 $\int_0^t X(s)dY(s)$ に収束する. しかし, $(Y(t))_{t \in [0, T]}$ がブラウン運動 $(B(t))_{t \in [0, T]}$ の場合, このことは一般には成り立たない. 具体例として, $(X(t))_{t \in [0, T]}$ と $(Y(t))_{t \in [0, T]}$ がともに同じブラウン運動 $(B(t))_{t \in [0, T]}$ であり, かつ $\tau_{n,i} := t_{n,i}$ とした場合を考える. このとき, リーマン和は

$$\begin{aligned} & \sum_{i: t_{n,i} \leq t} B(t_{n,i})(B(t_{n,i}) - B(t_{n,i-1})) \\ &= \sum_{i: t_{n,i} \leq t} B(t_{n,i-1})(B(t_{n,i}) - B(t_{n,i-1})) + \sum_{i: t_{n,i} \leq t} (B(t_{n,i}) - B(t_{n,i-1}))^2 \end{aligned}$$

のように書き直せるから, 命題 1.2 より $n \rightarrow \infty$ のとき

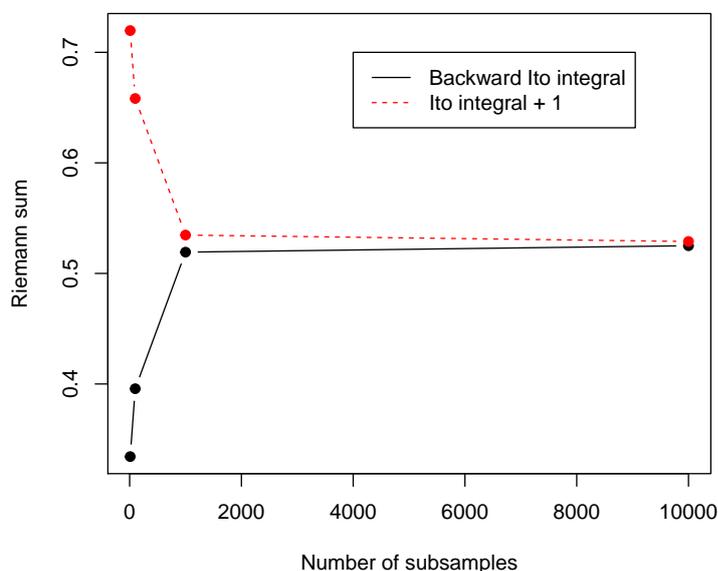
$$\int_0^t B(s)dB(s) + t$$

に収束することがわかる. このように, リーマン和 (2.2) において $X(t_{n,i-1})$ のかわりに $X(t_{n,i})$ を考えた場合に極限が存在する場合, その極限を B に関する X の**後向き伊藤積分**と呼ぶ.

以下に上の議論をシミュレーションで確認するための R コードを示す.

```
> ### 後向き伊藤積分
> ## ブラウン運動のシミュレーション
> set.seed(111) # 乱数シードの固定
> n <- 10000 # サンプル時点数
> B <- simulateBM(n) # サンプルパスのシミュレーション
> #yuimaB <- setData(zoo(B, seq(0,1,by=1/n))) # yuimaオブジェクトに変換しておく
> ## サブサンプルしたデータのリーマン和を計算してプロット
> ns <- 10^(1:4) # サブサンプリングする時点数(10,100,1000,10000)
> result <- matrix(0, length(ns), 2) # 結果を保持するための行列
> for(i in 1:length(ns)){
+   subB <- B[seq(1, n+1, length = ns[i]+1)] # サブサンプリング
+   result[i, 1] <- sum(subB[1:ns[i]] * diff(subB)) + 1 # 確率積分のリーマン和 + 1
+   result[i, 2] <- sum(subB[-1] * diff(subB)) # 後向き伊藤積分のリーマン和
+ }
> matplot(ns, result, type = "b", pch = 19,
+         col = c("black", "red"), lty = c(1, 2),
+         xlab = "Number of subsamples",
+         ylab = "Riemann sum") # プロット
> legend(4000, 0.7, col = c("black", "red"), lty = c(1, 2),
+       legend = c("Backward Ito integral",
+                 "Ito integral + 1"))
```

(backward.r)



関連する積分概念として**ストラトノヴィッチ積分 (Stratonovich integral)**がある。これは、リーマン和として

$$\sum_{i:t_{n,i} \leq t} \frac{X(t_{n,i-1}) + X(t_{n,i})}{2} (B(t_{n,i}) - B(t_{n,i-1}))$$

を考え、この量の $n \rightarrow \infty$ のときの極限として定義される積分である。 B に関する X のストラトノヴィッチ積分は記号

$$\int_0^t X(s) \circ dB(s)$$

で表されることが多い。例えば、上と同様の議論より

$$\int_0^t B(s) \circ dB(s) = \int_0^t B(s) dB(s) + \frac{t}{2}$$

が成り立つことがわかる。

2.3 伊藤の公式 (ブラウン運動の場合)

適合過程 $(X(t))_{t \in [0, T]}$ が微分可能なサンプルパスをもつとする。このとき、 $f(x)$ を C^1 級関数とすると、微分積分学の基本定理と連鎖律より

$$\begin{aligned} f(X(t)) - f(X(0)) &= \int_0^t \frac{d}{ds} f(X(s)) ds = \int_0^t f'(X(s)) \frac{d}{ds} X(s) ds \\ &= \int_0^t f'(X(s)) dX(s) \end{aligned}$$

が成り立つ。このようにして $f(X(t))$ の積分表現を得ることができる。別の見方をすると、この等式は積分

$$\int_0^t f'(X(s)) dX(s)$$

をリーマン和による近似ではなく $f(X(t)) - f(X(0))$ によって直接計算する手段を与えており、積分計算の上で非常に有用である。確率積分のリーマン和近似によるシミュレーションでは、誤差を小さくするために非常に細かい区間の分割を考える必要があるため、上のような積分表現の類推をブラウン運動に対して導出する問題は非常に重要である。しかし、ブラウン運動のサンプルパスは微分可能でないから、上の議論はもちろん適用できない。実際、ブラウン運動の場合は追加の補正項が必要なことが以下の議論からわかる。

仮定を強めて $f(x)$ は C^2 級であると仮定すると、テイラー展開によって

$$\begin{aligned} f(B(t_{n,i})) - f(B(t_{n,i-1})) &= f'(B(t_{n,i-1})) (B(t_{n,i}) - B(t_{n,i-1})) + \frac{1}{2} f''(\xi_{t_{n,i}}) (B(t_{n,i}) - B(t_{n,i-1}))^2, \\ B(t_{n,i-1}) \leq \xi_{n,i} \leq B(t_{n,i}), \quad i = 1, 2, \dots \end{aligned}$$

と書けることがわかる。従って、 $t_{n,i} \leq t$ を満たす最大の番号 i を $i_n(t)$ と書くことにすると、

$$\begin{aligned} f(B(t)) - f(B(0)) &= \sum_{i:t_{n,i} \leq t} \{f(B(t_{n,i})) - f(B(t_{n,i-1}))\} + \{f(B(t)) - f(B(t_{n,i_n(t)}))\} \\ &= \sum_{i:t_{n,i} \leq t} f'(B(t_{n,i-1})) (B(t_{n,i}) - B(t_{n,i-1})) + \frac{1}{2} \sum_{i:t_{n,i} \leq t} f''(\xi_{n,i}) (B(t_{n,i}) - B(t_{n,i-1}))^2 \\ &\quad + \{f(B(t)) - f(B(t_{n,i_n(t)}))\} \end{aligned} \tag{2.5}$$

と書ける. ここで, 確率積分の定義より, 第一項について

$$\lim_{n \rightarrow \infty} \sum_{i: t_{n,i} \leq t} f'(B_{t_{n,i-1}})(B_{t_{n,i}} - B_{t_{n,i-1}}) = \int_0^t f'(B(s))dB(s)$$

が成り立つ. 一方で, ブラウン運動は連続なサンプルパスをもつことから, 第三項について

$$\lim_{n \rightarrow \infty} |f(B(t)) - f(B(t_{n,i_n(t)}))| = 0$$

が言える. 最後に, 命題 1.2 を用いると, 第二項について

$$\lim_{n \rightarrow \infty} \frac{1}{2} \sum_{i: t_{n,i} \leq t} f''(\xi_{n,i})(B_{t_{n,i}} - B_{t_{n,i-1}})^2 = \frac{1}{2} \int_0^t f''(B(s))ds$$

が成り立つことが示せる. 以上より, $f(B(t))$ の積分表現として

$$f(B(t)) - f(B(0)) = \int_0^t f'(B(s))dB(s) + \frac{1}{2} \int_0^t f''(B(s))ds$$

を得る. これが著名な**伊藤の公式**の特別な場合である. 以下に応用例とそのシミュレーションによる確認をいくつか示す.

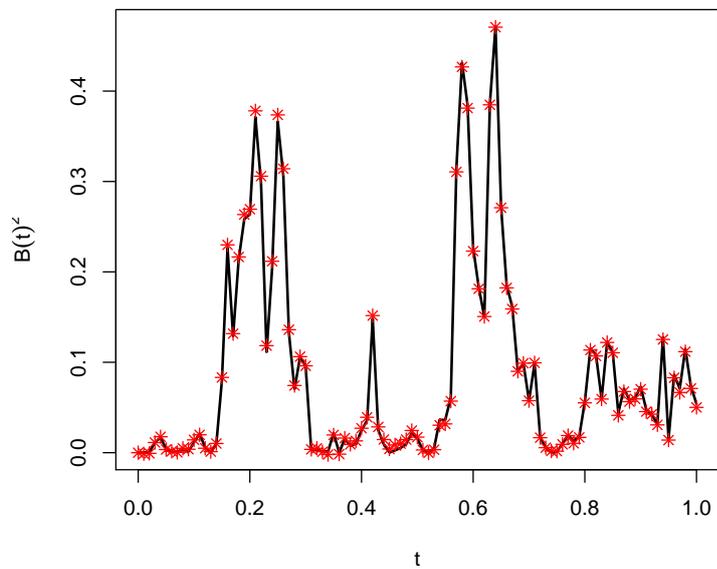
例 2.1. $f(x) = x^2$ とすれば, $f'(x) = 2x$, $f''(x) = 2$ であるから

$$B(t)^2 = 2 \int_0^t B(s)dB(s) + t$$

を得る. シミュレーション例は以下のようになる.

```
> ### 伊藤の公式のシミュレーション(1)
> Time.ito <- seq(0, 1, by = 0.01) # 伊藤の公式の評価時点
> n <- 10000 # サンプル時点数
> # (伊藤の公式の評価時点より細かく取っておく)
> set.seed(111) # 乱数シードの固定
> B <- simulateBM(n) # ブラウン運動のシミュレーション
> M <- diffinv(B[1:n] * diff(B)) # 確率積分の計算
> ito <- 2 * M + seq(0, 1, by = 1/n) # 伊藤の公式の計算
> B <- B[seq(1, n + 1, by = 100)] # 評価時点のみ残す
> ito <- ito[seq(1, n + 1, by = 100)] # 評価時点のみ残す
> ## 時系列プロットによる確認
> plot(Time.ito, B^2, type = "l", lwd = 2, xlab = "t",
+       ylab = expression(B(t)^2)) # B(t)^2のプロット
> points(Time.ito, ito, pch = 8, col = "red") # 伊藤の公式による値を上書き
```

(ito.r)

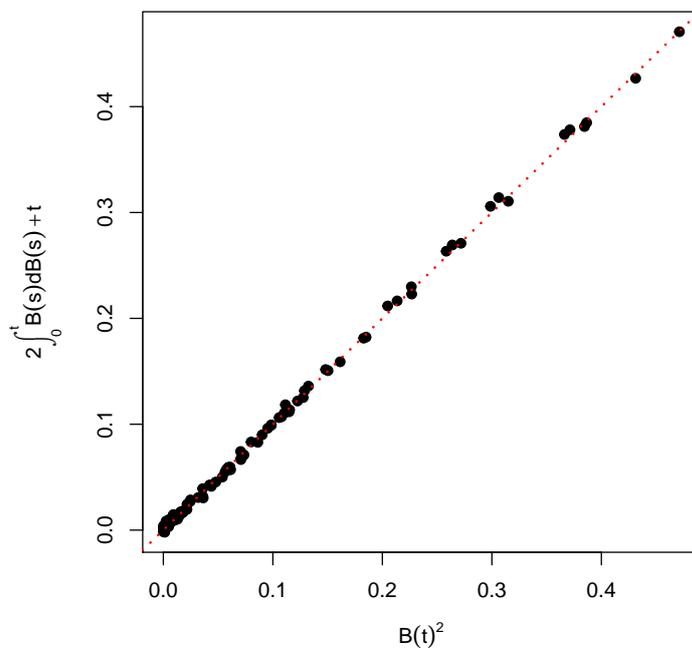


```

> ## 散布図による確認
> op <- par(mar = c(4, 6, 2, 2)) # 余白の設定
> plot(B^2, ito, pch = 19, xlab = expression(B(t)^2),
+       ylab = expression(2*integral(B(s)*dB(s),0,t)+t)) # 散布図
> abline(0, 1, lty = "dotted", col = "red", lwd = 2) # 45度線
> par(op) # 余白の設定をリセット

```

(ito-plot.r)



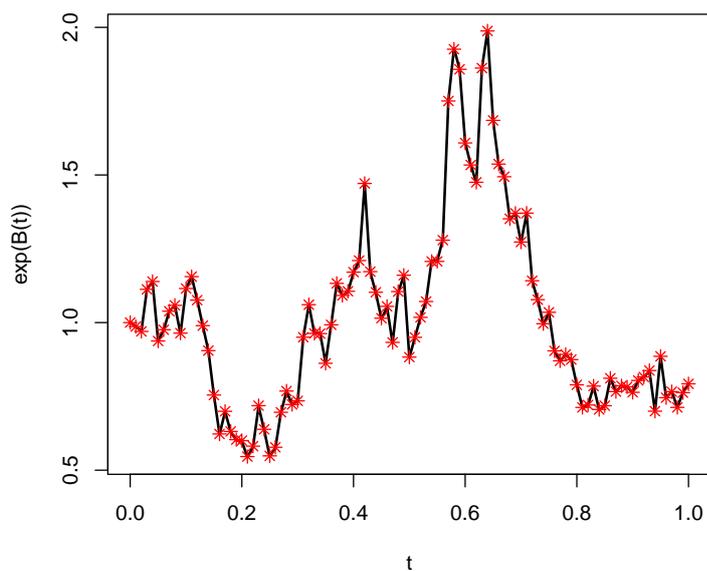
例 2.2. $f(x) = e^x$ とすると, $f'(x) = f''(x) = e^x$ であるから,

$$e^{B(t)} = 1 + \int_0^t e^{B(s)} dB(s) + \frac{1}{2} \int_0^t e^{B(s)} ds$$

を得る. シミュレーション例は以下のようになる.

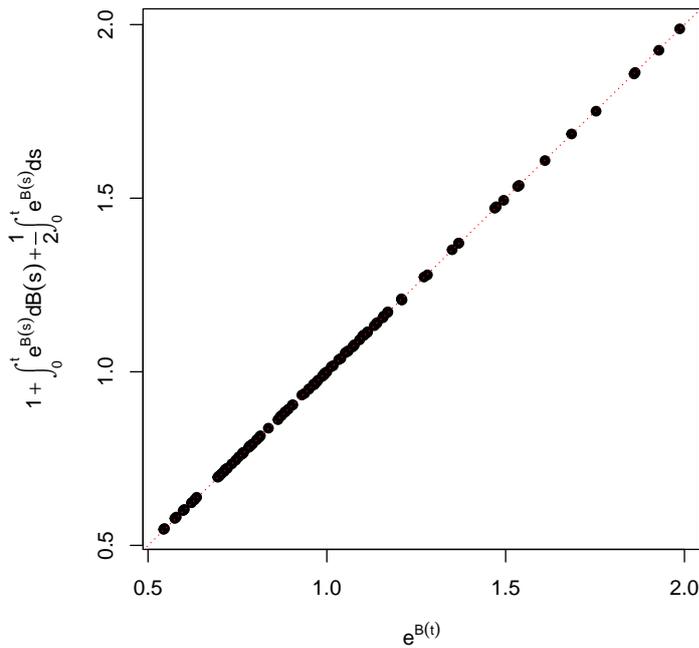
```
> ### 伊藤の公式のシミュレーション(2)
> Time.ito <- seq(0, 1, by = 0.01) # 伊藤の公式の評価時点
> n <- 10000 # サンプル時点数
> # (伊藤の公式の評価時点より細かく取っておく)
> set.seed(111) # 乱数シードの固定
> B <- simulateBM(n) # ブラウン運動のシミュレーション
> X <- exp(B)
> M <- diffinv(X[1:n] * diff(B)) # 確率積分の計算
> A <- diffinv(X[1:n])/n # 通常の積分の計算
> ito <- 1 + M + A/2 # 伊藤の公式の計算
> X <- X[seq(1, n + 1, by = 100)] # 評価時点のみ残す
> ito <- ito[seq(1, n + 1, by = 100)] # 評価時点のみ残す
> ## 時系列プロットによる確認
> plot(Time.ito, X, type = "l", lwd = 2, xlab = "t",
+       ylab = "exp(B(t))") # exp(B(t))のプロット
> points(Time.ito, ito, pch = 8, col = "red") # 伊藤の公式による値を上書き
```

(ito2.r)



```
> ## 散布図による確認
> op <- par(mar = c(4, 6, 2, 2)) # 余白の設定
> plot(X, ito, pch = 19, xlab = expression(e^{B(t)}), # 散布図
+       ylab = expression(1+integral(e^{B(s)}*dB(s),0,t)+
+                             frac(1,2)*integral(e^{B(s)}*ds,0,t)))
> abline(0, 1, lty = "dotted", col = "red") # 45度線
> par(op) # 余白の設定をリセット
```

(ito2-plot.r)



2.4 確率積分の二次変動と一般の場合の伊藤の公式

前節で説明したブラウン運動に対する伊藤の公式の導出では、ブラウン運動の二次変動が収束するという事実が重要な役割を果たした。そこで、本節では二次変動が収束するようなより広いクラスの確率過程を与えることで、より一般の場合の伊藤の公式を与える。

まずはじめに二次変動の概念を一般的に定義しておく。

定義 2.1 (二次変動). $(X(t))_{t \in [0, T]}$ を連続なサンプルパスをもつ確率過程とする。次の条件を満たす確率過程 $(Q(t))_{t \in [0, T]}$ が存在するとき、 $(Q(t))_{t \in [0, T]}$ を $(X(t))_{t \in [0, T]}$ の二次変動過程と呼び、記号 $([X](t))_{t \in [0, T]}$ で表す。

- $(\Delta_n)_{n \in \mathbb{N}}$ を $n \rightarrow \infty$ のとき 0 に収束する正数列とし、 $t_{n,i} := i\Delta_n$, $i = 0, 1, \dots$, とすると、すべての $t \in [0, T]$ に対して

$$Q(t) = \lim_{n \rightarrow \infty} \sum_{i: t_{n,i} \leq t} (X(t_{n,i}) - X(t_{n,i-1}))^2$$

が成り立つ。

命題 1.2 よりブラウン運動 $(B(t))_{t \in [0, T]}$ は二次変動過程をもち、それは

$$[B](t) = t, \quad t \in [0, T]$$

で与えられる。また、命題 1.2 の後の議論から、確率過程 $(A(t))_{t \in [0, T]}$ のサンプルパスが連続かつ有界変動をもつならば、 $(A(t))_{t \in [0, T]}$ は二次変動過程をもち、それは

$$[A](t) = 0, \quad t \in [0, T]$$

で与えられる。

より一般に次のことが知られている。

命題 2.2. 以下のように表現される確率過程 $(X(t))_{t \in [0, T]}$ を考える：

$$X(t) = A(t) + \int_0^t \sigma(s) dB(s), \quad t \in [0, T]. \quad (2.6)$$

ここに, $(A(t))_{t \in [0, T]}$ はサンプルパスが連続かつ有界変動をもつような確率過程であり, $(\sigma(t))_{t \in [0, T]}$ は連続なサンプルパスをもつ適合過程であるとする. このとき, 確率過程 $(X(t))_{t \in [0, T]}$ は二次変動過程をもち, それは

$$[X](t) = \int_0^t \sigma(s)^2 ds, \quad t \in [0, T]$$

で与えられる.

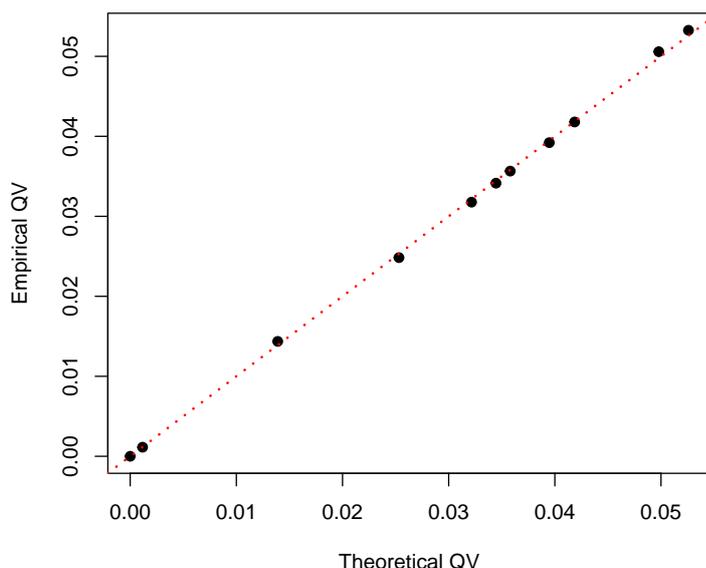
命題 2.2 より, 例えば $X(t) = \int_0^t B(s)dB(s)$ の場合は $(X(t))_{t \in [0, T]}$ は二次変動過程をもち, それは

$$[X](t) = \int_0^t B(s)^2 ds, \quad t \in [0, T]$$

で与えられることになる. 以下のコードはこのことをシミュレーションによって確認している.

```
> ### 確率積分の二次変動の収束
> ### 二次変動は時点0,0.1,...,0.9,1で評価
> n <- 10000 # サンプル時点数
> # (二次変動の評価時点より細かく取っておく)
> set.seed(123) # 乱数シードの固定
> B <- simulateBM(n) # Brown運動のシミュレーション
> X <- diffinv(B[1:n] * diff(B)) # 確率積分を近似するリーマン和の計算
> qv <- diffinv(diff(X)^2) # 二次変動の計算
> qv <- qv[seq(1, n+1, by = 1000)] # 評価時点のみ残す
> qv.true <- diffinv(B[1:n]^2/n) # 理論値の計算
> qv.true <- qv.true[seq(1, n+1, by = 1000)] # 評価時点のみ残す
> plot(qv ~ qv.true, pch = 19, xlab = "Theoretical QV",
+      ylab = "Empirical QV") # 理論値 vs 観測値の散布図
> abline(0, 1, lty = "dotted", col = "red", lwd = 2) # 45度線
```

(qv-integral.r)



命題 2.2 より, 前節の伊藤の公式の導出における (2.5) 式の第二項は, ブラウン運動を (2.6) 式で与えられる確率過程 $(X(t))_{t \in [0, T]}$ に置き換えても同様の議論で処理できる. 第三項についても, $(X(t))_{t \in [0, T]}$ が連続

なサンプルパスを持つことから、やはり同様である。残りは第一項の処理であるが、これは次の命題を使えばよい。

命題 2.3. (2.6) 式で与えられる確率過程 $(X(t))_{t \in [0, T]}$ を考える。また、 $(H(t))_{t \in [0, T]}$ を連続なサンプルパスをもつ適合過程とする。このとき、 $(\Delta_n)_{n \in \mathbb{N}}$ を $n \rightarrow \infty$ のとき 0 に収束する正数列とし、 $t_{n,i} := i\Delta_n$, $i = 0, 1, \dots$, とおくと、すべての $t \in [0, T]$ に対して

$$\lim_{n \rightarrow \infty} \sum_{i: t_{n,i} \leq t} H(t_{n,i-1})(X(t_{n,i}) - X(t_{n,i-1})) = \int_0^t H(s) dA(s) + \int_0^t H(s) \sigma(s) dB(s)$$

が成り立つ。

命題 2.3 より積分 $\int_0^t H(s) dX(s)$ は以下のように定義するのが自然である：

$$\int_0^t H(s) dX(s) := \int_0^t H(s) dA(s) + \int_0^t H(s) \sigma(s) dB(s).$$

以上の準備の下、より一般的な設定の下での伊藤の公式が以下のとおり得られる：

定理 2.1 (伊藤の公式). (2.6) 式で与えられる確率過程 $(X(t))_{t \in [0, T]}$ を考える。このとき、 C^2 級関数 $f(x)$ と $t \in [0, T]$ に対して以下の等式が成り立つ。

$$\begin{aligned} & f(X(t)) - f(X(0)) \\ &= \int_0^t f'(X(s)) dX(s) + \frac{1}{2} \int_0^t f''(X(s)) d[X](s) \\ &= \int_0^t f'(X(s)) dA(s) + \int_0^t f'(X(s)) \sigma(s) dB(s) + \frac{1}{2} \int_0^t f''(X(s)) \sigma(s)^2 ds. \end{aligned}$$

第3章 確率微分方程式とそのシミュレーション

時刻 t と状態 x を変数とする 2 つの関数 $a(t, x), b(t, x)$ を考える. また, $(B(t))_{t \in [0, T]}$ をブラウン運動とする. 確率過程 $(X(t))_{t \in [0, T]}$ に関する以下の形の方程式

$$X(t) = X(0) + \int_0^t a(s, X(s))ds + \int_0^t b(s, X(s))dB(s), \quad t \in [0, T] \quad (3.1)$$

をブラウン運動 $(B(t))_{t \in [0, T]}$ で駆動される**確率微分方程式 (stochastic differential equation, SDE)** と呼ぶ.

(3.1) 式は微分方程式というより積分方程式だが, この呼び名は歴史的経緯による.¹ 特に, (3.1) 式は慣習として以下のような微分形で書き表される:

$$dX(t) = a(t, X(t))dt + b(t, X(t))dB(t), \quad t \in [0, T]. \quad (3.2)$$

なお, 関数 $a(t, x), b(t, x)$ はそれぞれ確率微分方程式 (3.2) の**ドリフト係数**, **拡散係数**と呼ばれる.

本章では, 確率微分方程式でモデル化される確率過程の具体例および諸性質を説明し, それらのシミュレーションによる確認を行う.

3.1 オイラー・丸山法

一般に確率微分方程式が陽に解けることは稀であり, 多くの場合, 解のシミュレーションには数値的方法を用いる. 本稿では, そのような数値的方法のうち最もよく利用される代表的な方法である**オイラー・丸山法 (Euler-Maruyama scheme)** について説明する.

時間区間 $[0, T]$ を n 等分割した時点 $t_i = i\Delta_n$ ($i = 0, 1, \dots, n; \Delta_n = T/n$) をサンプル時点として確率微分方程式 (3.2) の解をシミュレーションする問題を考える. オイラー・丸山法では, 表現 (3.1) より,

$$X(t_{i+1}) - X(t_i) = \int_{t_i}^{t_{i+1}} a(s, X(s))ds + \int_{t_i}^{t_{i+1}} b(s, X(s))dB(s), \quad i = 1, \dots, n$$

が成り立つことを用いる. n が十分大きければ, 上の式の右辺は

$$a(t_i, X(t_i))\Delta_n + b(t_i, X(t_i))(B(t_{i+1}) - B(t_i))$$

でよく近似できると考えられる. この考えに基づいて, シミュレーションしたい系列 $(X(t_i))_{i=0}^n$ を, 以下のように帰納的に定義される系列 $(\hat{X}_n(t_i))_{i=0}^n$ で近似する: $\hat{X}_n(0) := X(0)$ として,

$$\hat{X}_n(t_{i+1}) = \hat{X}_n(t_i) + a(t_i, \hat{X}_n(t_i))\Delta_n + b(t_i, \hat{X}_n(t_i))(B(t_{i+1}) - B(t_i)), \quad i = 1, \dots, n$$

と定義する. 適当な条件下でこの近似法が正当化できること, すなわち

$$\lim_{n \rightarrow \infty} \max_{i=0, 1, \dots, n} |\hat{X}_n(t_i) - X(t_i)| = 0$$

が成り立つことが知られている (例えば [14], 6.1 節参照).

¹実際, (3.1) 式の形の方程式が導入された当時は, 「確率微分方程式」という用語は別のものを指しており, 方程式 (3.1) 自身は「確率積分方程式」と呼ばれていた. [10] を参照のこと.

Rのパッケージ `yuima` では、直感的な記述によって、一般的な確率微分方程式モデルのシミュレーション・パラメータ推定・モデル評価という一連の統計解析を実行するためのフレームワークを用意している。その詳細については [7] を参照していただきたい。ここでは例として、未知パラメータ θ を含む確率微分方程式

$$dX(t) = -\theta X(t)dt + \frac{1}{1+X(t)^2}dB(t), \quad t \in [0, 1]$$

を `yuima` によってシミュレーションする方法を説明する。

```
> ### yuimaパッケージによる確率微分方程式のシミュレーション
> library(yuima) # パッケージのロード
> ## 確率微分方程式モデルの定義
> mod <- setModel(drift = "-theta*x", # ドリフト係数
+               diffusion = "1/(1+x^2)", # 拡散係数
+               state.variable = "x" # 状態変数(既定値が"x"のため省略可)
+ )
> str(mod) # モデルを表すオブジェクトの構造
```

```
Formal class 'yuima.model' [package "yuima"] with 16 slots
 ..@ drift          : expression((-theta * x))
 ..@ diffusion      :List of 1
 .. ..$ : expression((1/(1 + x^2)))
 ..@ hurst          : num 0.5
 ..@ jump.coeff     : list()
 ..@ measure        : list()
 ..@ measure.type   : chr(0)
 ..@ parameter      :Formal class 'model.parameter' [package "yuima"] with 7 slots
 .. .. ..@ all      : chr "theta"
 .. .. ..@ common   : chr(0)
 .. .. ..@ diffusion: chr(0)
 .. .. ..@ drift    : chr "theta"
 .. .. ..@ jump     : chr(0)
 .. .. ..@ measure  : chr(0)
 .. .. ..@ xinit    : chr(0)
 ..@ state.variable: chr "x"
 ..@ jump.variable  : chr(0)
 ..@ time.variable  : chr "t"
 ..@ noise.number   : num 1
 ..@ equation.number: int 1
 ..@ dimension      : int [1:6] 1 0 0 1 0 0
 ..@ solve.variable : chr "x"
 ..@ xinit          : expression((0))
 ..@ J.flag         : logi FALSE
```

```
> ## サンプル時点の定義
> n <- 1000
> samp <- setSampling(Initial = 0, # 初期時点(既定値が0なので省略可)
+                   Terminal = 1, # 終端時点(既定値が1なので省略可)
+                   n = n # サンプリング時点数(初期時点は除く)
```

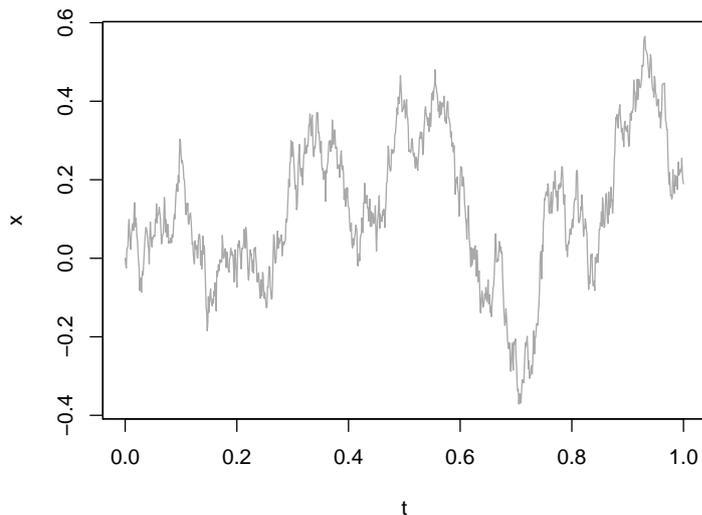
```
+ )
> str(samp) # サンプル時点を表すモデルの構造
```

```
Formal class 'yuima.sampling' [package "yuima"] with 11 slots
..@ Initial      : num 0
..@ Terminal     : num 1
..@ n            : int 1000
..@ delta        : num 0.001
..@ grid         :List of 1
.. ..$ : num [1:1001] 0 0.001 0.002 0.003 0.004 0.005 0.006 0.007 0.008 0.009 ...
..@ random       : logi FALSE
..@ regular      : logi TRUE
..@ sdelta       : num(0)
..@ sgrid        : num(0)
..@ oindex       : num(0)
..@ interpolation: chr "pt"
```

```
> ## 統計モデルの定義
> smod <- setYuima(model = mod, # 確率微分方程式モデル
+                 sampling = samp # サンプリング時点
+ )
> str(samp) # 統計モデルを表すモデルの構造
```

```
Formal class 'yuima.sampling' [package "yuima"] with 11 slots
..@ Initial      : num 0
..@ Terminal     : num 1
..@ n            : int 1000
..@ delta        : num 0.001
..@ grid         :List of 1
.. ..$ : num [1:1001] 0 0.001 0.002 0.003 0.004 0.005 0.006 0.007 0.008 0.009 ...
..@ random       : logi FALSE
..@ regular      : logi TRUE
..@ sdelta       : num(0)
..@ sgrid        : num(0)
..@ oindex       : num(0)
..@ interpolation: chr "pt"
```

```
> ## シミュレーションとサンプルパスの描画
> set.seed(123) # 乱数シードの固定
> x0 <- 0 # 初期値
> param <- list(theta = 3) # パラメーターの値
> # シミュレーションの実行
> result <- simulate(smod, # 統計モデル
+                  xinit = x0, # 初期値
+                  true.parameter = param # パラメーターの値
+ )
> plot(result, col = "darkgray") # プロット
```



3.2 代表的なモデル

この節では、いくつかの著名な確率微分方程式モデルとその性質を説明する。

3.2.1 幾何ブラウン運動

μ を定数, σ を正の定数とする. ドリフト係数が $a(t, x) = \mu x$, 拡散係数が $b(t, x) = \sigma x$ の場合の確率微分方程式 (3.2)

$$dX(t) = \mu X(t)dt + \sigma X(t)dB(t), \quad t \in [0, T] \quad (3.3)$$

の解をドリフト μ , ボラティリティ σ の**幾何ブラウン運動**と呼ぶ. ファイナンス分野では, この確率過程によってリスク資産の価格過程をモデル化し, そのリスク資産に紐づけられた他の商品の理論価格を導出するという方法が成功をおさめたため, 提案者の名前にちなんで**ブラック・ショールズモデル (Black-Scholes model)** と呼ばれることが多い.

確率微分方程式 (3.3) は陽に解くことができ, その解は

$$X(t) = X(0) \exp\left(\left(\mu - \frac{\sigma^2}{2}\right)t + \sigma B(t)\right), \quad t \in [0, T] \quad (3.4)$$

で与えられる. 実際, (3.4) 式で与えられる確率過程 $(X(t))_{t \in [0, T]}$ が確率微分方程式 (3.3) の解となることは, 伊藤の公式を用いると確認できる.

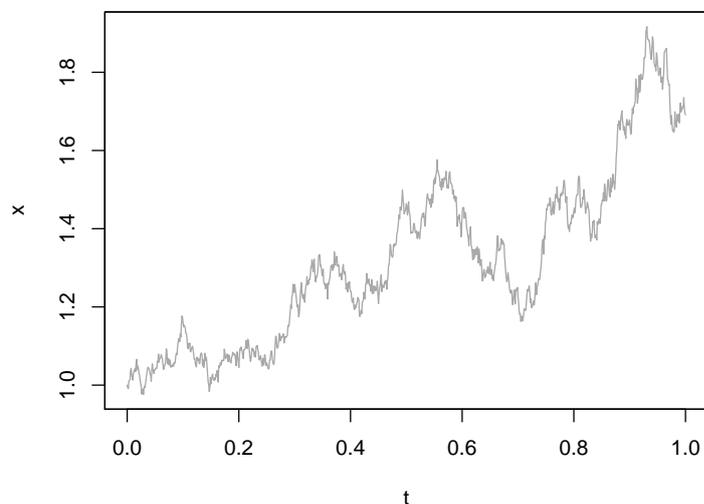
```
> ### yuimaパッケージによる幾何ブラウン運動のシミュレーション
> library(yuima) # パッケージのロード
> ## 確率微分方程式モデルの定義
> mod <- setModel(drift = "mu*x", # ドリフト係数
+                 diffusion = "sigma*x", # 拡散係数
+                 state.variable = "x" # 状態変数
+ )
> ## サンプル時点の定義(区間[0,1]上でシミュレート)
> n <- 1000 # サンプル時点数
```

```

> samp <- setSampling(n = n)
> ## 統計モデルの定義
> gbm <- setYuima(model = mod, # 確率微分方程式モデル
+               sampling = samp # サンプリング時点
+ )
> ## シミュレーション
> set.seed(123)
> x0 <- 1 # 初期値
> param <- list(mu = 0.4, sigma = 0.4) # パラメーターの値
> # ここでは後で解析解と結果を比較したいため、あらかじめ駆動する
> # ブラウン運動の増分をシミュレーションしておく
> dB <- rnorm(n)/sqrt(n)
> # シミュレーションの実行
> result <- simulate(gbm, # 統計モデル
+                  xinit = x0, # 初期値
+                  true.parameter = param, # パラメーターの値
+                  increment.W = matrix(dB,1,n)
+                  # 駆動するブラウン運動の増分(技術的な理由から行列に変換)
+ )
> plot(result, col = "darkgray")

```

(yuimaGBM1.r)



```

> ## 解析解との比較
> mu <- param$mu # ドリフト
> sigma <- param$sigma # ボラティリティ
> Time <- samp@grid[[1]] # サンプル時点
> B <- diffinv(dB) # 駆動するブラウン運動
> X <- x0 * exp((mu - sigma^2/2) * Time + sigma * B) # 解析解のシミュレーション
> ## 時系列プロットによる比較
> ## まず、simulate関数で生成したサンプルパスに対応するzooオブジェクトを取得しプロット
> Xhat <- get.zoo.data(result)[[1]]

```

```
> str(Xhat)
```

```
‘zooreg’ series from 0 to 1
```

```
Data: num [1:1001] 1 0.993 0.991 1.011 1.012 ...
```

```
Index: num [1:1001] 0 0.001 0.002 0.003 0.004 0.005 0.006 0.007 0.008 0.009 ...
```

```
Frequency: 1000
```

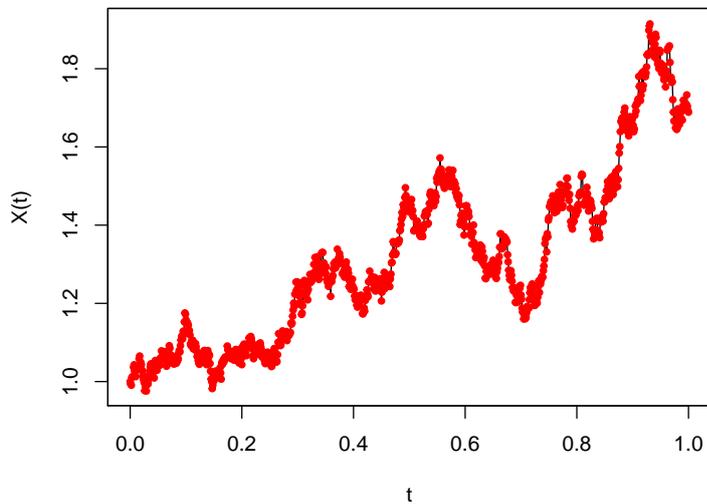
```
> plot(Xhat, xlab = "t", ylab = "X(t)") # プロット
```

```
> ## 次に、解析解によるサンプルパスを上書き
```

```
> lines(zoo(X, Time), # zooに変換しておく
```

```
+ type = "p", pch = 20, col = "red")
```

(yuimaGBM2.r)

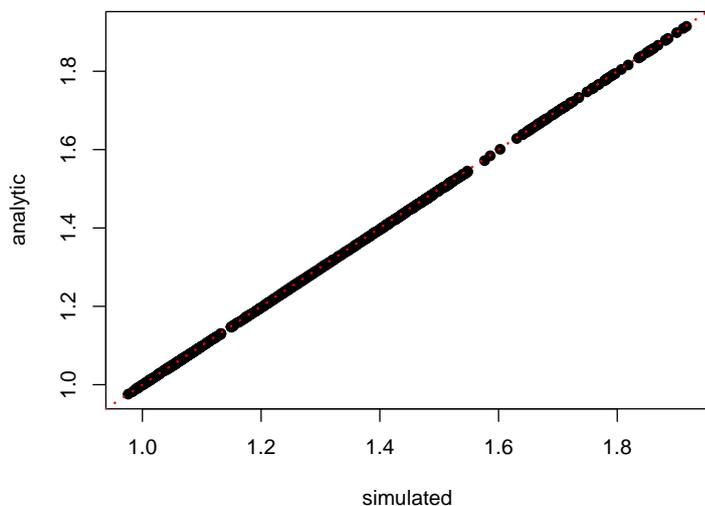


```
> ## 散布図による比較
```

```
> plot(Xhat, X, pch = 19, xlab = "simulated", ylab = "analytic")
```

```
> abline(0, 1, lty = "dotted", col = "red", lwd = 2) # 45度線
```

(yuimaGBM3.r)



3.2.2 オルンシュタイン・ウーレンベック過程

θ, μ, σ を定数とする. 確率微分方程式

$$dX(t) = -\theta(X(t) - \mu)dt + \sigma dB(t) \quad (3.5)$$

の解として与えられる確率過程 $(X(t))_{t \in [0, T]}$ を **オルンシュタイン・ウーレンベック過程 (Ornstein-Uhlenbeck process)** と呼ぶ. 確率微分方程式 (3.5) は陽に解くことができ、その解は

$$X(t) = X(0)e^{-\theta t} + \mu(1 - e^{-\theta t}) + \sigma \int_0^t e^{-\theta(t-s)} dB(s)$$

で与えられる (伊藤の公式を用いると確認できる). 特に, 初期値 $X(0)$ が定数であれば, 命題 2.1 より $X(t)$ は平均 $X(0)e^{-\theta t} + \mu(1 - e^{-\theta t})$, 分散 $\sigma^2 \int_0^t e^{-2\theta(t-s)} ds = \frac{\sigma^2}{2\theta}(1 - e^{-2\theta t})$ の正規分布に従う.

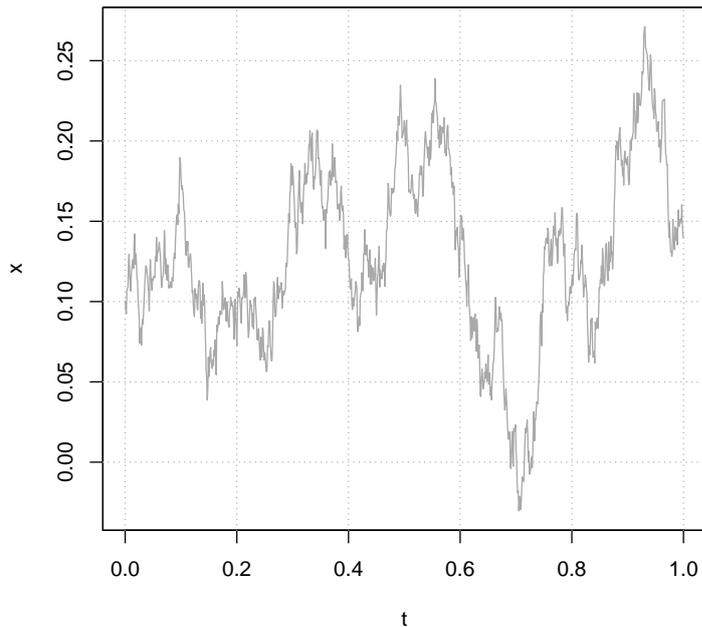
```
> ### オルンシュタイン・ウーレンベック過程のシミュレーション
> ## 確率微分方程式モデルの定義
> mod.ou <- setModel(drift = "-theta*(x-mu)", # ドリフト係数
+                   diffusion = "sigma" # 拡散係数
+ )
> ## サンプル時点の定義
> n <- 1000 # サンプル時点数
> samp <- setSampling(n = n)
> ## 統計モデルの定義
> ou <- setYuima(model = mod.ou, # 確率微分方程式モデル
+               sampling = samp # サンプル時点
+ )
> ## シミュレーションとサンプルパスの描画
> set.seed(123)
> x0 <- 0.1 # 初期値
> param <- list(theta = 3, mu = 0.09, sigma = 0.3) # パラメータの値
> # シミュレーションの実行
```

```

> result <- simulate(ou, # 統計モデル
+                   xinit = x0, # 初期値
+                   true.parameter = param # パラメーターの値
+ )
> plot(result, col = "darkgray")
> grid(col = "darkgray") # グリッドの追加

```

(ou.r)



```

> ### オルンシュタイン・ウーレンバック過程の周辺分布
> ## シミュレーターの作成
> simulateXT <- function(smod, x0, param){
+   out <- simulate(smod, xinit = x0, true.parameter = param)
+   XT <- tail(get.zoo.data(out)[[1]], n = 1) # 最後の観測値を取り出す
+   return(XT)
+ }
> ## パラメーターの設定
> x0 <- 0.1 # 初期値
> theta <- 3 # 平均回帰スピード
> mu <- 0.09 # 平均回帰レベル
> sigma <- 0.3 # ボラティリティ
> param <- list(theta = theta, mu = mu, sigma = sigma)
> ## モンテカルロ・シミュレーションの実行
> set.seed(123) # 乱数シードの固定
> MC <- 2000 # シミュレーション回数
> result <- replicate(MC, simulateXT(ou, x0, param)) # シミュレーション
> ## ヒストグラムの描画
> hist(result, freq = FALSE, breaks = 20,
+       xlab = "X(1)", main = "Histogram of X(1)",

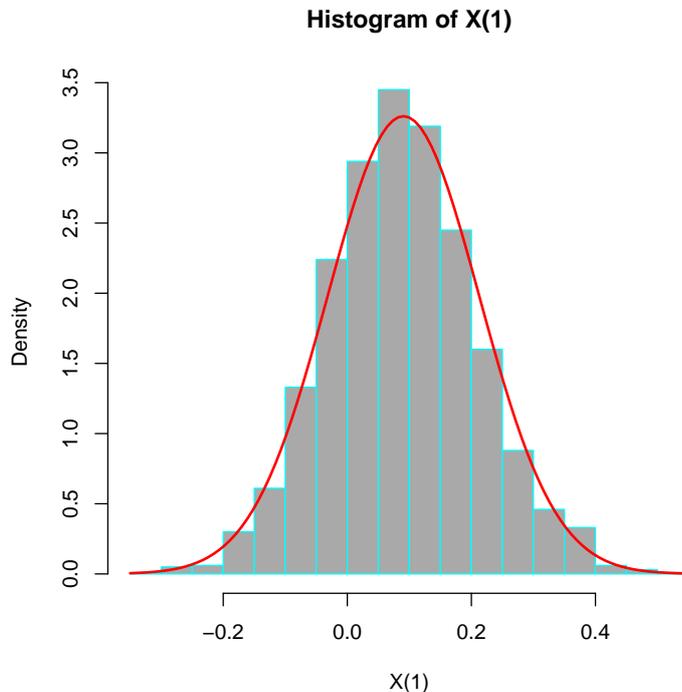
```

```

+     col = "darkgray", border = "cyan")
> ## 理論分布の上書き
> t.mu <- x0 * exp(-theta) + mu * (1 - exp(-theta)) # 理論上の平均
> t.var <- (sigma^2/(2*theta)) * (1 - exp(-2*theta)) # 理論上の分散
> curve(dnorm(x, t.mu, sqrt(t.var)), add = TRUE,
+     col="red", lwd = 2)

```

(ou-dist.r)



$\theta > 0$ の場合, オルンシュタイン・ウーレンバック過程は**平均回帰性**をもつ. すなわち, 確率過程のサンプルパスがある一定の水準 (平均回帰レベルと呼ばれる) を上下動し, その水準から大きく乖離しても長期的にはその水準へと戻っていく性質をもつ. 実際, 確率微分方程式 (3.5) のドリフト係数は

$$-\theta(X(t) - \mu) \begin{cases} < 0 & \text{if } X(t) > \mu, \\ > 0 & \text{if } X(t) < \mu \end{cases}$$

という性質を持つから, 常に平均回帰レベル μ へと過程を押し戻す力を働かせている. 特に, パラメーター θ は平均回帰レベルへと向かうスピードを表すと解釈することができるため, **平均回帰スピード**とも呼ばれる.

ファイナンスの分野では, 金利の時系列過程は上述の「平均回帰性」をもつべきだと認識されている. そのため, 平均回帰性をもつ確率過程の代表格であるオルンシュタイン・ウーレンバック過程は金利モデルとしても利用されてきた. そのため, モデルの提案者にちなんで, ファイナンス分野ではオルンシュタイン・ウーレンバック過程のことを**バシチェクモデル (Vasicek model)**とも呼ぶ.

```

> ### オルンシュタイン・ウーレンバック過程の平均回帰スピードの比較
> ## サンプル時点の定義
> n <- 1000 # サンプル時点数
> samp <- setSampling(Terminal = 10, # 終端時点
+     n = n # サンプル時点数
+ )
> Time <- samp@grid[[1]] # サンプル時点
> ## 統計モデルの定義

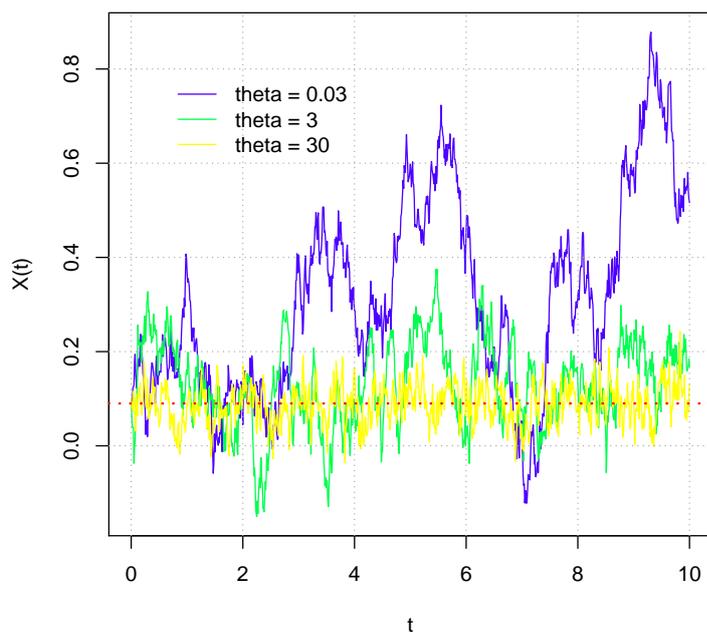
```

```

> ou <- setYuima(model = mod.ou, # 確率微分方程式モデル
+               sampling = samp # サンプリング時点
+ )
> ## パラメーターの設定
> x0 <- 0.1 # 初期値
> theta <- c(0.03, 3, 30) # 平均回帰スピードの候補ベクトル
> mu <- 0.09 # 平均回帰レベル
> sigma <- 0.3 # ボラティリティ
> ## シミュレーション
> set.seed(123) # 乱数シードの固定
> result <- matrix(0, n + 1, length(theta)) # 結果を保持する行列
> for(i in 1:length(theta)){
+   param <- list(theta = theta[i], mu = mu, sigma = sigma)
+   out <- simulate(ou, xinit = x0, true.parameter = param)
+   result[,i] <- get.zoo.data(out)[[1]]
+ }
> ## プロット
> matplot(Time, result, type = "l", lty = 1,
+         col = topo.colors(length(theta)),
+         xlab = "t", ylab = "X(t)")
> abline(h = mu, col = "red", lwd = 2, lty = "dotted") # 平均回帰レベル
> grid(col = "darkgray") # グリッドの追加
> ## 凡例の追加
> legend(0.5, 0.8, legend = paste("theta =", theta),
+       lty = 1, col = topo.colors(length(theta)), bty = "n")

```

(ou-mean.r)



3.2.3 コックス・インガーソル・ロスモデル

オルンシュタイン・ウーレンベック過程は負の値をとりうるため、マイナス金利導入以前の世界ではこの点がバシチェックモデルの欠点とされていた。この問題点を解消するために、[4]は確率微分方程式

$$dX(t) = -\theta(X(t) - \mu)dt + \sigma\sqrt{X(t)}dB(t), \quad X(0) \geq 0 \quad (3.6)$$

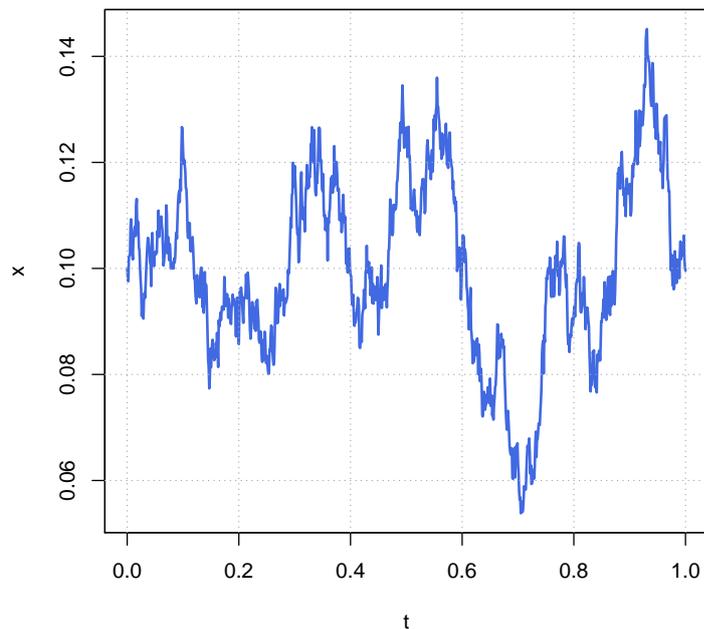
の解として与えられる確率過程 $(X(t))_{t \in [0, T]}$ を新たな金利モデルとして提案し、現在は**コックス・インガーソル・ロスモデル (Cox-Ingersoll-Ross model, CIR model)** と呼ばれている。ただし、 μ, σ は0以上の定数、 θ は定数である。ドリフト係数の形から明らかのように、 $\theta > 0$ であれば CIR モデルもバシチェックモデル同様平均回帰性をもつ。

方程式 (3.6) の拡散係数 $\sigma\sqrt{X(t)}$ は $X(t)$ が0以上でないと定義できないため、(3.6) の解 $(X(t))_{t \in [0, T]}$ はすべての $t \in [0, T]$ について $X(t) \geq 0$ という条件を満たす必要があるが、そのような解は常にただ一つ存在することが知られている ([16], 6章定理 2.3)。さらに、すべての $t \in [0, T]$ について $X(t)$ が正となるような解が存在するためには、 $X(0) > 0$ かつ、パラメーターについて $2\theta\mu \geq \sigma^2$ という条件 (**フェラー条件 (Feller condition)** と呼ばれる) があればよいことも知られている ([16], 6章命題 2.4)。

バシチェックモデルと異なり、CIR モデルは扱いやすい解析解を持たないため、シミュレーションには通常オイラー・丸山法を用いる。このとき注意が必要なのは、オイラー・丸山法は近似であるため、シミュレーションされた値が負になりうるということである。そのためシミュレーションの際には負となった値は0以上の値に読みかえる工夫が必要である。以下のコードにそのような工夫の一つを例示する。

```
> ### CIRモデルのシミュレーション
> ## 確率微分方程式モデルの定義
> ## 拡散係数に現れる平方根の中では、負に評価された値を0と読みかえるようにする
> mod <- setModel(drift = "-theta*(x-mu)", # ドリフト係数
+               diffusion = "sigma * sqrt(max(x, 0))" # 拡散係数
+ )
> ## サンプル時点の定義
> n <- 1000 # サンプル時点数
> samp <- setSampling(n = n)
> ## 統計モデルの定義
> cir <- setYuima(model = mod, # 確率微分方程式モデル
+               sampling = samp # サンプル時点
+ )
> ## シミュレーションとサンプルパスの描画
> set.seed(123)
> x0 <- 0.1 # 初期値
> param <- list(theta = 3, mu = 0.09, sigma = 0.3) # パラメーターの値
> # シミュレーションの実行
> result <- simulate(cir, # 統計モデル
+                  xinit = x0, # 初期値
+                  true.parameter = param # パラメーターの値
+ )
> plot(result, col = "royalblue", lwd = 2)
> grid(col = "darkgray") # グリッドの追加
```

(cir.r)

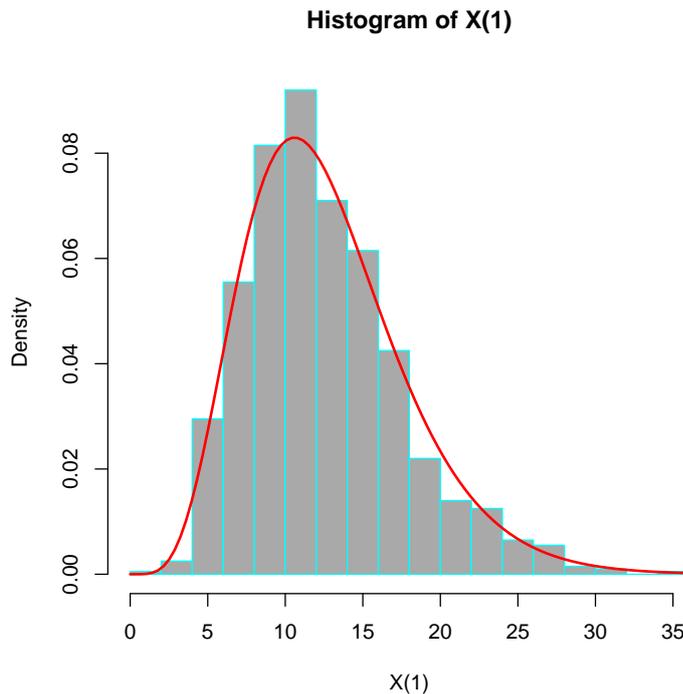


初期値 $X(0)$ が定数の場合, 各時点 $t \in [0, T]$ における $X(t)$ の分布は以下の通り具体的に知られている. 実際,

$$c_t := \frac{2\theta}{(1 - e^{-\theta t})\sigma^2}$$

とおくと, $2c_t X(t)$ は自由度 $4\theta\mu/\sigma^2$, 非心パラメータ $-2c_t X(0)e^{-\theta t}$ の非心 χ^2 分布に従うことが知られている ([16], 158 頁).

```
> ### CIRモデルの周辺分布
> ## パラメータの設定
> x0 <- 0.1 # 初期値
> theta <- 3 # 平均回帰スピード
> mu <- 0.09 # 平均回帰レベル
> sigma <- 0.3 # ボラティリティ
> param <- list(theta = theta, mu = mu, sigma = sigma)
> ## モンテカルロ・シミュレーションの実行
> set.seed(123) # 乱数シードの固定
> MC <- 1000 # シミュレーション回数
> result <- replicate(MC, simulateXT(cir, x0, param)) # シミュレーション
> ## ヒストグラムの描画
> ct <- 2 * theta / ((1 - exp(-theta)) * sigma^2)
> hist(2 * ct * result, freq = FALSE, breaks = 20,
+      xlab = "X(1)", main = "Histogram of X(1)",
+      col = "darkgray", border = "cyan")
> ## 理論分布の上書き
> df <- 4 * theta * mu / sigma^2 # 自由度
> ncp <- 2 * ct * x0 * exp(-theta) # 非心パラメータ
> curve(dchisq(x, df = df, ncp = ncp), add = TRUE, col="red", lwd = 2)
```



3.2.4 ヤコビ過程

θ, σ を正の定数, μ を定数とする. 確率微分方程式

$$dX(t) = -\theta(X(t) - \mu)dt + \sigma\sqrt{X(t)(1-X(t))}dB(t), \quad 0 < X(0) < 1 \quad (3.7)$$

の解として与えられる確率過程 $(X(t))_{t \in [0, T]}$ を**ヤコビ過程 (Jacobi process)**と呼ぶ. 拡散係数 $\sigma\sqrt{X(t)(1-X(t))}$ が定義できるためにはすべての $t \in [0, T]$ について $0 \leq X(t) \leq 1$ となっていなければならないが, このためには

$$\frac{\sigma^2}{2\theta} \leq \mu \leq 1 - \frac{\sigma^2}{2\theta}$$

というパラメータ制約が満たされていればよいことが知られている ([5] 参照. この場合, 実際にはすべての $t \in [0, T]$ について $0 < X(t) < 1$ となる).

ヤコビ過程は 0 と 1 の間を時間発展する確率過程の典型例であり, 適当な 1 次変換を施すことで, 上限と下限があるような時系列のモデル化に利用することができる. 例えば, [23] は相関パラメータが時間発展するようなモデルにおいて, その時間発展をヤコビ過程によってモデル化することを提案している (正確には, 上限が 1, 下限が -1 となるように確率過程 $(2X(t) - 1)_{t \in [0, T]}$ を考える). 他にも, [5] はヤコビ過程を金利モデルとして用いることを提案している.

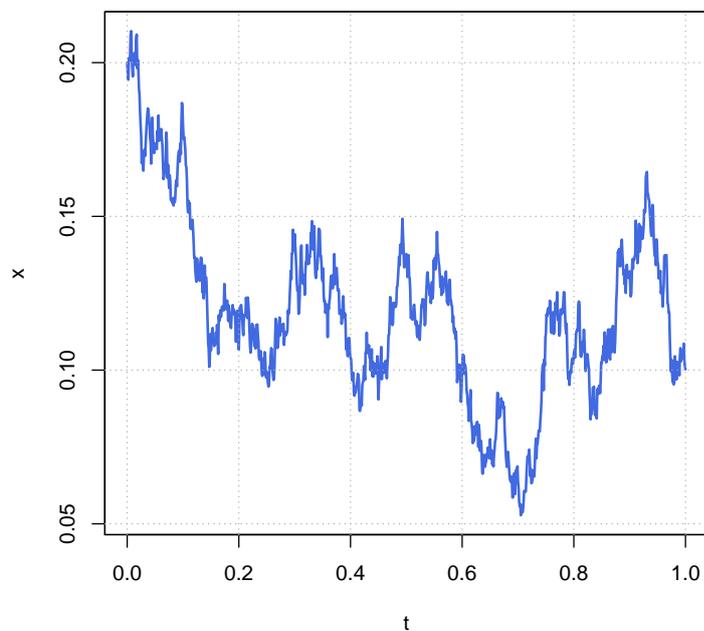
```
> ### ヤコビ過程のシミュレーション
> ## 確率微分方程式モデルの定義
> ## 拡散係数に現れる平方根の中では, 負に評価された値を0と読みかえるようにする
> mod <- setModel(drift = "-theta*(x-mu)", # ドリフト係数
+                 diffusion = "sigma * sqrt(max(x*(1-x), 0))"
+                 # 拡散係数
+ )
> ## サンプル時点の定義
> n <- 1000
```

```

> samp <- setSampling(n = n)
> ## 統計モデルの定義
> jacobi <- setYuima(model = mod, sampling = samp)
> ## シミュレーションとサンプルパスの描画
> set.seed(123)
> x0 <- 0.2 # 初期値
> param <- list(theta = 8, mu = 0.1, sigma = 0.4) # パラメーターの値
> # シミュレーションの実行
> result <- simulate(jacobi, # 統計モデル
+                   xinit = x0, # 初期値
+                   true.parameter = param # パラメーターの値
+ )
> plot(result, col = "royalblue", lwd = 2)
> grid(col = "darkgray") # グリッドの追加

```

(jacobi.r)



3.2.5 ベッセル過程

δ を 2 以上の定数とする. 確率微分方程式

$$dX(t) = \frac{\delta - 1}{2X(t)} dt + dB(t), \quad X(0) > 0 \quad (3.8)$$

の解として与えられる確率過程 $(X(t))_{t \in [0, T]}$ を δ 次元**ベッセル過程 (Bessel process)** と呼ぶ. 方程式 (3.8) の解はすべての $t \in [0, T]$ について $X(t) > 0$ を満たすことが知られている ([22], 例 4.8.4 参照).

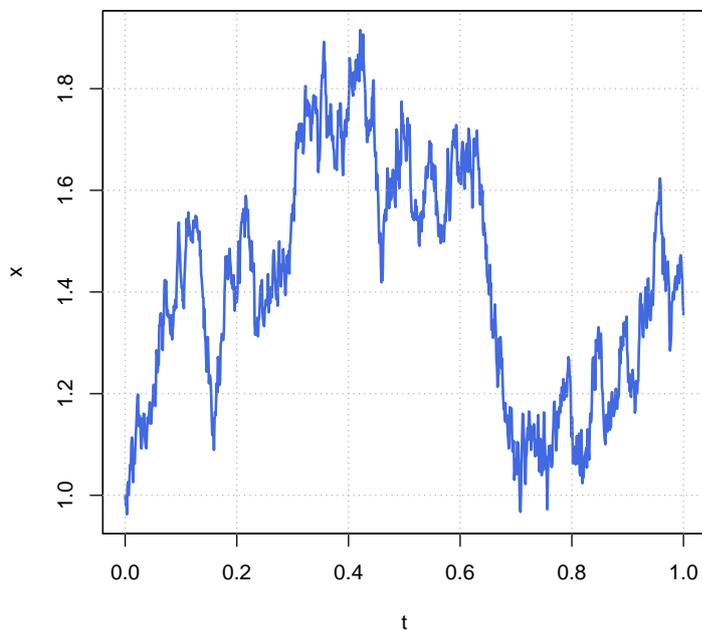
δ が整数の場合, $(B_1(t))_{t \in [0, T]}, \dots, (B_\delta(t))_{t \in [0, T]}$ を δ 個の独立なブラウン運動とすると, $(X(t))_{t \in [0, T]}$ は $(\sqrt{B_1(t)^2 + \dots + B_\delta(t)^2})_{t \in [0, T]}$ と同じ確率分布をもつことが知られている ([19], 446 頁参照). $\sqrt{B_1(t)^2 + \dots + B_\delta(t)^2}$ は δ 次元空間の点 $(B_1(t), \dots, B_\delta(t))$ の原点からのユークリッド距離であり, このことが「 δ 次元」という用語の語源である.

```

> ### ベッセル過程のシミュレーション
> ## 確率微分方程式モデルの定義
> mod <- setModel(drift = "(delta-1)/(2*x)", # ドリフト係数
+               diffusion = "1" # 拡散係数
+ )
> ## サンプル時点の定義
> n <- 1000
> samp <- setSampling(n = n)
> ## 統計モデルの定義
> besel <- setYuima(model = mod, sampling = samp)
> ## シミュレーションとサンプルパスの描画
> set.seed(1)
> x0 <- 1 # 初期値
> param <- list(delta = 3) # パラメーターの値
> # シミュレーションの実行
> result <- simulate(besel, # 統計モデル
+                   xinit = x0, # 初期値
+                   true.parameter = param # パラメーターの値
+ )
> plot(result, col = "royalblue", lwd = 2)
> grid(col = "darkgray") # グリッドの追加

```

(besel.r)



$\delta = 3$ の場合、初期値 $X(0)$ が正の定数 x_0 であれば確率変数 $X(t)$ の分布の確率密度関数 $f(y)$ は次式で与えられることが知られている:

$$f(y) = \frac{y}{t} \sqrt{\frac{y}{x_0}} e^{-(x_0^2 + y^2)/(2*t)} \sqrt{\frac{2}{\pi x_0 y}} \sinh\left(\frac{x_0 y}{t}\right), \quad y > 0.$$

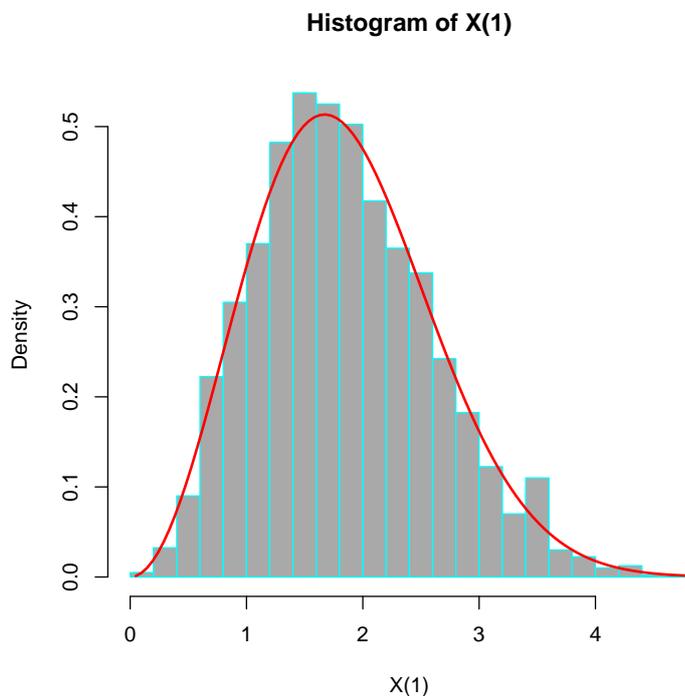
ここに, \sinh は双曲線正弦関数を表す:

$$\sinh(z) = \frac{e^z - e^{-z}}{2}.$$

一般の δ の場合の $X(t)$ の分布の確率密度関数は第1種変形ベッセル関数という特殊関数を用いると陽に書くことができる. 詳細は [18], 10.4 節, もしくは [19], 446 頁を参照のこと.

```
> ### 3次元ベッセル過程の周辺分布
> ## パラメーターの設定
> x0 <- 1 # 初期値
> delta <- 3 # 次元
> param <- list(delta = delta)
> ## モンテカルロ・シミュレーションの実行
> set.seed(1) # 乱数シードの固定
> MC <- 2000 # シミュレーション回数
> result <- replicate(MC, simulateXT(bessel, x0, param)) # シミュレーション
> ## ヒストグラムの描画
> hist(result, freq = FALSE, breaks = 20,
+       xlab = "X(1)", main = "Histogram of X(1)",
+       col = "darkgray", border = "cyan")
> ## 理論分布の上書き
> f <- function(y)
+   y*sqrt(y/x0)*exp(-(x0^2+y^2)/2)*sqrt(2/(pi*x0*y))*sinh(x0*y)
> curve(f, from = 0, add = TRUE, col="red", lwd = 2)
```

(bessel3-dist.r)



3.3 確率ボラティリティモデル

これまで1次元の確率過程・確率微分方程式のみを取り扱ってきたが, これらの概念は多次元の場合も同様に定義することができる. オイラー・丸山法は多次元の場合に自然に拡張できるため, それを使って多

次元の確率微分方程式もシミュレーションできるが、実際にパッケージ `yuima` の関数 `simulate()` はその方法で多次元の確率微分方程式をシミュレートする機能を備えている。

この節では、そのような多次元の確率微分方程式を `yuima` でシミュレートする方法を例示するために、ファイナンス分野で広く利用されている**確率ボラティリティモデル**をいくつか紹介する。確率ボラティリティモデルとは、平たくいうと時間発展する分散をもつような確率過程モデルで、分散の時間発展もまた確率過程で記述されるようなものの総称である。ファイナンス分野では標準偏差のことをボラティリティと呼称することが多いことが名前の由来となっている。

3.3.1 ヘストンモデル

μ, ρ を定数, θ, v, γ を正の定数とし, $|\rho| \leq 1$ であるとする。また, $(B_1(t))_{t \in [0, T]}$ および $(B_2(t))_{t \in [0, T]}$ を 2 つの独立なブラウン運動とする。確率微分方程式

$$\begin{cases} dS(t) = \mu S(t)dt + \sqrt{V(t)}S(t)dB_1(t), & S(0) > 0, \\ dV(t) = -\theta(V(t) - v)dt + \gamma\sqrt{V(t)}(\rho dB_1(t) + \sqrt{1 - \rho^2}dB_2(t)), & V(0) > 0 \end{cases} \quad (3.9)$$

で定義される 2 次元確率過程 $(S(t), V(t))_{t \in [0, T]}$ を**ヘストンモデル (Heston model)** と呼ぶ。ヘストンモデルでは、確率過程 $(S(t))_{t \in [0, T]}$ は株価の時系列過程を表し、その収益率過程が時間発展する分散 $(V(t))_{t \in [0, T]}$ をもつようにモデル化されている。 $B(t) := \rho B_1(t) + \sqrt{1 - \rho^2}B_2(t)$ とおくと、確率過程 $(B(t))_{t \in [0, T]}$ はブラウン運動となることが確認できる。このとき、 $(V(t))_{t \in [0, T]}$ を定義する確率微分方程式は

$$dV(t) = -\theta(V(t) - v)dt + \gamma\sqrt{V(t)}dB(t)$$

と書き直せるが、これは CIR モデルである。すなわち、ヘストンモデルとは収益率過程の分散の時間発展を CIR モデルで記述した確率ボラティリティモデルとなっている。

パラメーター ρ は収益率過程と分散過程の間の相関係数に対応する。株価の時系列データでは、収益率とボラティリティの間に負の相関関係が観察されるという、**レバレッジ効果**と呼ばれる著名な実証結果があるため、この現象をモデルに反映させるためにパラメーター ρ が導入されている。従って、株価のモデル化にヘストンモデルを利用する場合、通常 ρ は負の値をとる。

多次元の確率微分方程式では、一般にドリフト係数はベクトル、拡散係数は行列で表現される。これは、例えば方程式 (3.9) は、形式的には

$$\begin{pmatrix} dS(t) \\ dV(t) \end{pmatrix} = \begin{pmatrix} \mu S(t) \\ -\theta(V(t) - v) \end{pmatrix} dt + \begin{pmatrix} \sqrt{V(t)}S(t) & 0 \\ \gamma\sqrt{V(t)} \cdot \rho & \gamma\sqrt{V(t)} \cdot \sqrt{1 - \rho^2} \end{pmatrix} \begin{pmatrix} dB_1(t) \\ dB_2(t) \end{pmatrix}$$

と書き表されるからである。従って、`yuima` でも多次元の確率微分方程式を記述するオブジェクトのドリフト係数、拡散係数はそれぞれベクトル、行列で表現される。具体的な方法については以下の実行例を参照していただきたい。

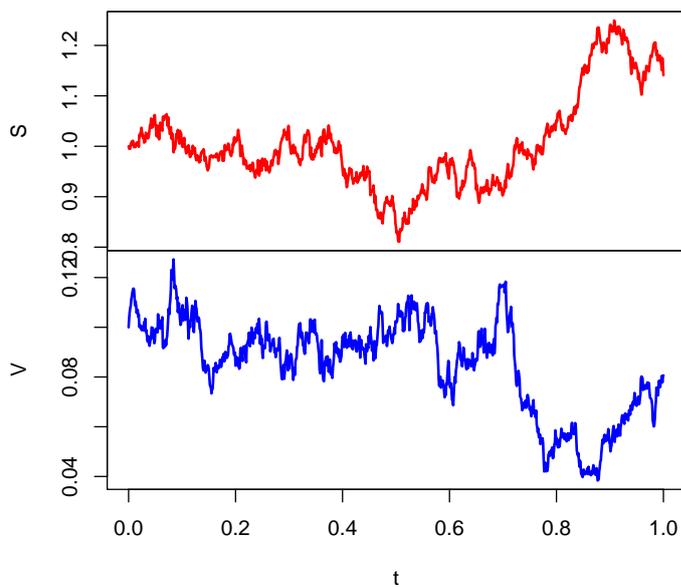
```
> ### ヘストンモデルのシミュレーション
> ## 確率微分方程式モデルの定義
> drift <- c("mu*S", "-theta*(V-v)") # ドリフト係数
> diffusion <- matrix(c("sqrt(max(V,0))*S",
+                       "gamma*sqrt(max(V,0))*rho",
+                       0,
+                       "gamma*sqrt(max(V,0))*sqrt(1-rho^2)"),
+                    2,2) # 拡散係数
> mod <- setModel(drift = drift, diffusion = diffusion,
+                 state.variable = c("S", "V") # 状態変数
+                 )
```

```

> ## サンプル時点の定義
> samp <- setSampling(n = 1000)
> ## 統計モデルの定義
> heston <- setYuima(model = mod, sampling = samp)
> ## シミュレーションとサンプルパスの描画
> set.seed(100)
> x0 <- c(1, 0.1) # 初期値
> param <- list(mu = 0.03, theta = 3, v = 0.09,
+             gamma = 0.3, rho = -0.6) # パラメーターの値
> # シミュレーションの実行
> result <- simulate(heston, # 統計モデル
+                   xinit = x0, # 初期値
+                   true.parameter = param # パラメーターの値
+ )
> plot(result, col = c("red", "blue"), lwd = 2)

```

(heston.r)



3.3.2 SABR モデル

ヘストンモデルと並んで著名な確率ボラティリティモデルとして **SABR モデル**がある。SABR モデルは次の確率微分方程式で定義される2次元確率過程 $(X(t), \alpha(t))_{t \in [0, T]}$ である:

$$\begin{aligned}
 dX(t) &= \alpha(t)X(t)^\beta dB_1(t), & X(0) > 0, \\
 d\alpha(t) &= \nu\alpha(t)(\rho dB_1(t) + \sqrt{1-\rho^2}dB_2(t)) & \alpha(0) > 0.
 \end{aligned}$$

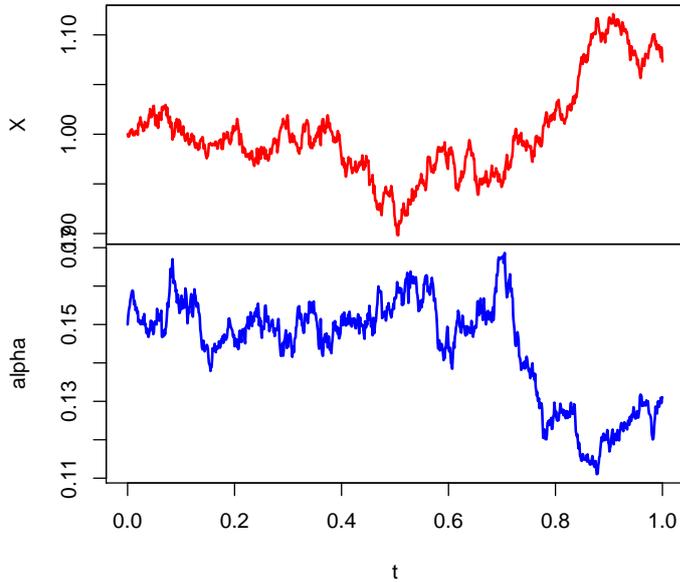
ただし、 β, ν, ρ は定数であり、 $\frac{1}{2} \leq \beta \leq 1, \nu \geq 0, |\rho| \leq 1$ を満たすとする。²

² $\beta < \frac{1}{2}$ の場合を考えることもあるが、この場合確率微分方程式の解はただ一つには定まらないため注意が必要である。[2] および [9] を参照のこと。

SABR という名前は “Stochastic Alpha Beta Rho” の略であり, モデルを記述するのに用いられた記号に由来している. パラメーター ρ はヘストンモデルと同様レバレッジ効果に対応しており, 通常は負にとる.

```
> ### SABRモデルのシミュレーション
> ## 確率微分方程式モデルの定義
> drift <- c(0, 0) # ドリフト係数
> diffusion <- matrix(c("alpha*max(X,0)^beta",
+                       "nu*alpha*rho",
+                       0,
+                       "nu*alpha*sqrt(1-rho^2)"),
+                     2,2) # 拡散係数
> mod <- setModel(drift = drift, diffusion = diffusion,
+                state.variable = c("X", "alpha") # 状態変数
+ )
> ## サンプル時点の定義
> samp <- setSampling(n = 1000)
> ## 統計モデルの定義
> sabr <- setYuima(model = mod, sampling = samp)
> ## シミュレーションとサンプルパスの描画
> set.seed(100)
> x0 <- c(1, 0.15) # 初期値
> param <- list(beta = 0.6, nu = 0.35, rho = -0.6) # パラメーターの値
> # シミュレーションの実行
> result <- simulate(sabr, # 統計モデル
+                   xinit = x0, # 初期値
+                   true.parameter = param # パラメーターの値
+ )
> plot(result, col = c("red", "blue"), lwd = 2)
```

(sabr.r)



3.4 不変分布とエルゴード性

確率微分方程式 (3.2) において, ドリフト係数 $a(t, x)$ と拡散係数 $b(t, x)$ が状態変数 x にのみ依存する場合, **時間的に一様 (homogeneous)** であるという. この場合, 係数の時間変数 t は省略してよいから, 時間的に一様な確率微分方程式は

$$dX(t) = a(X(t))dt + b(X(t))dB(t), \quad t \in [0, T] \quad (3.10)$$

と書ける.

確率分布 Π で次の条件を満たすものが存在するとき, Π を確率微分方程式 (3.10) の**不変分布**または**定常分布**と呼ぶ:

- 初期値 $X(0)$ が確率分布 Π に従って生成されるならば, すべての $t \in [0, T]$ について確率変数 $X(t)$ の分布は Π となる.

適当な仮定の下で, 確率微分方程式 (3.10) は不変分布をもち, かつ以下に述べるように具体的に書き下すことができる.

はじめに, いくつか記号を導入する. 以下, $-\infty \leq l < r \leq \infty$ とし, 拡散係数 $b(x)$ は区間 (l, r) で常に正の値をとる状況を考え, $l < x_0 < r$ なる実数 x_0 を1つとる. そして, **スケール密度**

$$s(x) = \exp\left(-2 \int_{x_0}^x \frac{a(y)}{b(y)^2} dy\right), \quad x \in (l, r),$$

および**スピード密度**

$$m(x) = \frac{2}{b(x)^2 s(x)}, \quad x \in (l, r)$$

を定義する. このとき, 適当な仮定の下で, 確率微分方程式 (3.10) は不変分布をもち, その確率密度関数 $f(x)$ は

$$f(x) = \begin{cases} m(x) / \int_l^r m(y) dy & \text{if } l < x < r, \\ 0 & \text{otherwise.} \end{cases}$$

で与えられる.³

上の事実を利用すると, 区間 (l, r) で常に正の値をとりそれ以外のところで 0 となるような確率密度関数 $f(x)$ が適当な条件を満たすときに, 平均回帰的なドリフト係数

$$a(x) = -\theta(x - \mu)$$

をもつ確率微分方程式 (3.10) で, その解が $f(x)$ に対応する確率分布を不変分布にもつようなものを具体的に構成することができる (詳細は [3] を参照). ここに, $\theta > 0$ であり, $\mu = \int_l^r x f(x) dx$ は $f(x)$ に対応する確率分布の平均を表す. 実際,

$$b(x) = \left(\frac{\theta \int_l^x (\mu - y) f(y) dy}{f(x)} \right)^2$$

とおけば求めるべき確率微分方程式が構成できる.

例 3.1 (正規分布). $f(x)$ が標準正規密度関数の場合, 対応する確率微分方程式は

$$dX(t) = -\theta X(t) dt + \sqrt{2\theta} dB(t)$$

となる. これはバシチェックモデルである.

例 3.2 (ガンマ分布). $\alpha \geq 1, \lambda > 0$ とする. $f(x)$ が形状パラメータ α , レート λ のガンマ分布の確率密度関数

$$f(x) = \frac{\lambda^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\lambda}, \quad x > 0$$

の場合, 対応する確率微分方程式は

$$dX(t) = -\theta(X(t) - \alpha/\lambda) dt + \sqrt{\frac{2\theta}{\lambda} X(t)} dB(t)$$

となる. これは CIR モデルである.

例 3.3 (ベータ分布). $\alpha, \beta > 0$ とする. $f(x)$ がパラメータ α, β のベータ分布の確率密度関数の場合, 対応する確率微分方程式は

$$dX(t) = -\theta \left(X(t) - \frac{\alpha}{\alpha + \beta} \right) dt + \sqrt{\frac{2\theta}{\alpha + \beta} X(t)(1 - X(t))} dB(t)$$

となる. これはヤコビ過程である.

例 3.4 (t 分布). $\nu > 1$ とする. $f(x)$ が自由度 ν の t 分布の確率密度関数の場合, 対応する確率微分方程式は

$$dX(t) = -\theta X(t) dt + \sqrt{\frac{2\theta}{\nu - 1} (\nu + X(t)^2)} dB(t)$$

となる. この場合の解は **スチューデント拡散過程 (Student diffusion)** と呼ばれることがある ([1], [17]).

例 3.5 (F 分布). $\alpha \geq 2, \beta > 2$ とする. $f(x)$ が自由度 α, β の F 分布の確率密度関数の場合, 対応する確率微分方程式は

$$dX(t) = -\theta \left(X(t) - \frac{\beta}{\beta - 2} \right) dt + \sqrt{\frac{4\theta}{\alpha(\beta - 2)} X(t)(\beta + \alpha X(t))} dB(t)$$

となる. この場合の解は **フィッシャー・スネデカー拡散過程 (Fisher - Snedecor diffusion)** と呼ばれることがある ([13]).

以下では t 分布と F 分布の場合に, 上で例示した確率微分方程式の不変分布が実際にそれぞれ t 分布と F 分布になっていることをシミュレーションで確認してみる.

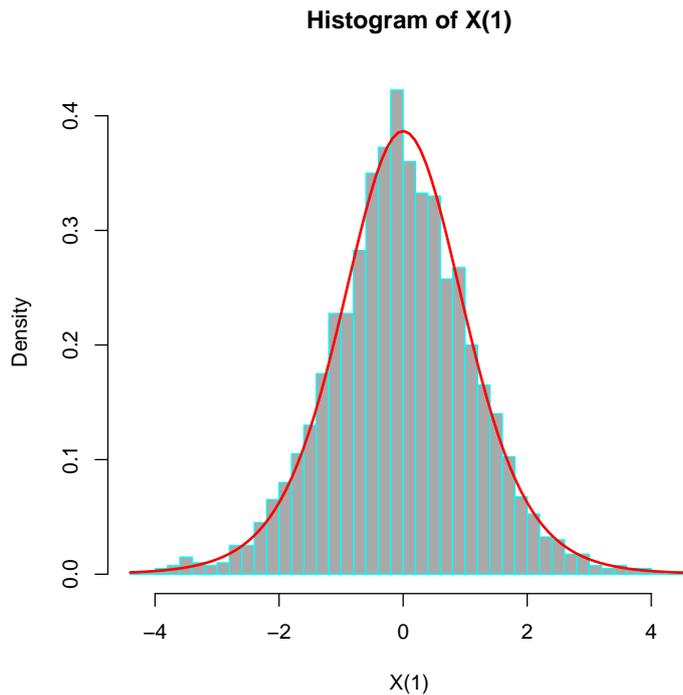
³例えば, $\int_l^{x_0} s(x) dx = \int_{x_0}^r s(x) dx = \infty$ かつ $\int_l^r m(y) dy < \infty$ であればよい. [15], Proposition 1.15 を参照のこと.

```

> ### t分布を不変分布にもつ拡散過程
> ## 確率微分方程式モデルの定義
> mod <- setModel(drift = "-theta*x", # ドリフト係数
+               diffusion = "sqrt(2*theta*(nu+x^2)/(nu-1))" # 拡散係数
+ )
> ## サンプル時点の定義
> n <- 1000
> samp <- setSampling(n = n)
> ## 統計モデルの定義
> smod <- setYuima(model = mod, sampling = samp)
> ## シミュレーターの作成
> mysim <- function(theta, nu){
+   param <- list(theta = theta, nu = nu)
+   out <- simulate(smod,
+                 xinit = rt(1, df = nu), # 不変分布から初期値を生成
+                 true.parameter = param)
+   XT <- tail(get.zoo.data(out)[[1]], n = 1) # 最後の観測値を取り出す
+   return(XT)
+ }
> ## シミュレーション
> set.seed(123)
> MC <- 2000
> result <- replicate(MC, mysim(theta = 10, nu = 8))
> hist(result, freq = FALSE, breaks = 50,
+       xlab = "X(1)", main = "Histogram of X(1)",
+       col = "darkgray", border = "cyan")
> curve(dt(x,df=8), add = TRUE, col="red", lwd = 2)

```

(t-stationary.r)



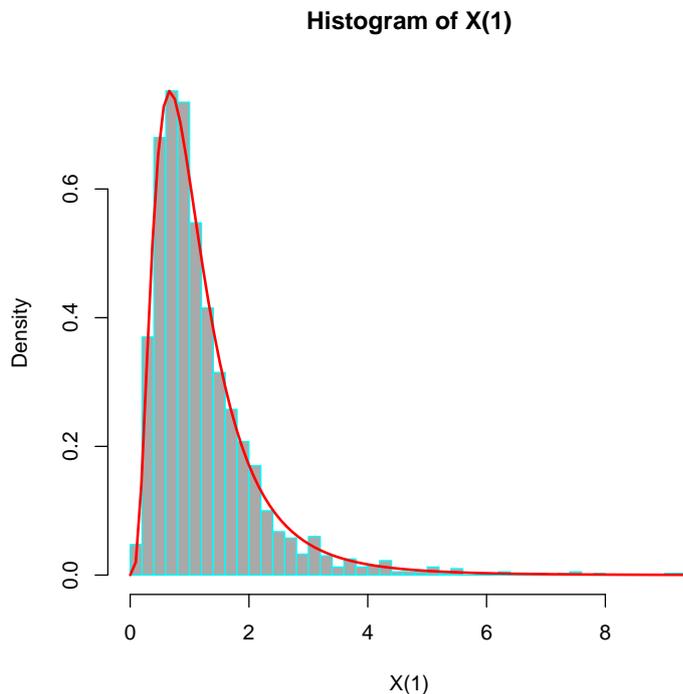
```

> ### F分布を不変分布にもつ拡散過程
> ## 確率微分方程式モデルの定義
> mod <- setModel(drift = "-theta*(x-beta/(beta-2))", # ドリフト係数
+               diffusion = "sqrt(4*theta*x*(beta+alpha*x)/(alpha*(beta-2)))" # 拡散
+               係数
+ )
> ## サンプル時点の定義
> n <- 1000
> samp <- setSampling(n = n)
> ## 統計モデルの定義
> smod <- setYuima(model = mod, sampling = samp)
> ## シミュレーターの作成
> mysim <- function(theta, alpha, beta){
+   param <- list(theta = theta, alpha = alpha, beta = beta)
+   out <- simulate(smod,
+                 xinit = rf(1, df1 = alpha, df2 = beta), # 不変分布から初期値を生成
+                 true.parameter = param)
+   XT <- tail(get.zoo.data(out)[[1]], n = 1) # 最後の観測値を取り出す
+   return(XT)
+ }
> ## シミュレーション
> set.seed(123)
> MC <- 2000
> result <- replicate(MC, mysim(theta = 1, alpha = 10, beta = 10))
> hist(result, freq = FALSE, breaks = 50,
+       xlab = "X(1)", main = "Histogram of X(1)",
+       col = "darkgray", border = "cyan")

```

```
> curve(df(x,df1=10,df2=10), add = TRUE, col="red",lwd=2)
```

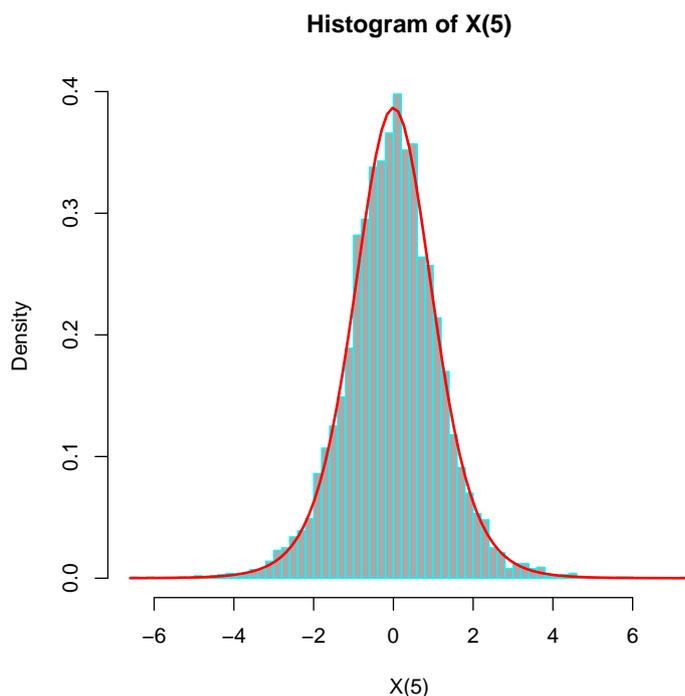
(f-stationary.r)



上で与えた例では、実はどんな初期値から出発しても、時間の経過とともに $X(t)$ の分布は不変分布へと近づいて行くことが知られている (このような性質を**エルゴード性**と呼ぶ). 以下では t 分布の場合についてこのことを確かめてみる.

```
> ### t分布を不変分布にもつ拡散過程のエルゴード性
> ## 確率微分方程式モデルの定義
> mod <- setModel(drift = "-theta*x", # ドリフト係数
+               diffusion = "sqrt(2*theta*(nu+x^2)/(nu-1))" # 拡散係数
+ )
> ## サンプル時点の定義
> n <- 1000
> samp <- setSampling(Terminal = 5, n = n)
> ## 統計モデルの定義
> smod <- setYuima(model = mod, sampling = samp)
> ## シミュレーション
> set.seed(123)
> MC <- 5000
> result <- replicate(MC, simulateXT(smod, x0 = 0,
+                                   param = list(theta = 10, nu = 8)))
> hist(result, freq = FALSE, breaks = 50,
+       xlab = "X(5)", main = "Histogram of X(5)",
+       col = "darkgray", border = "cyan")
> curve(dt(x,df=8), add = TRUE, col="red", lwd = 2)
```

(t-diffusion.r)



3.5 参考: 確率微分方程式の解の存在と一意性

この節では, 確率微分方程式 (3.2) の解の存在および一意性に関する補足を述べる. 実は, 確率微分方程式の解とその一意性の定義には強い意味と弱い意味の2種類があるのだが, ここでは強い意味でのものを考える (強い意味での解の一意性は**道ごとの一意性**と呼ばれることが多い). この点についての詳細は確率解析の専門書を参照のこと.

確率微分方程式 (3.2) の解の存在および一意性に関して最も基本的なのは次の結果である.

定理 3.1. ある定数 $K > 0$ が存在して, すべての $t \in [0, T]$ と $x, y \in \mathbb{R}$ について

$$|a(t, x) - a(t, y)| + |b(t, x) - b(t, y)| \leq K|x - y| \quad (3.11)$$

および

$$|a(t, x)| + |b(t, x)| \leq K(1 + |x|)$$

が成り立つと仮定する. このとき, 確率微分方程式 (3.2) はただ一つ解をもつ.

証明は例えば [12], 定理 2.9 もしくは [24], 第 4 章定理 2.2 系を参照のこと. 条件 (3.11) は**リプシッツ条件 (Lipschitz condition)** と呼ばれる.

多次元の確率微分方程式の場合にも上と同様の結果が成り立つ. 一方で, 1次元の場合には, 拡散係数に関するリプシッツ条件は以下のように緩めることができることが知られている:

定理 3.2. ある定数 $K > 0$ および $\beta \geq \frac{1}{2}$ が存在して, すべての $t \in [0, T]$ と $x, y \in \mathbb{R}$ について

$$|a(t, x) - a(t, y)| \leq K|x - y|, \quad |b(t, x) - b(t, y)| \leq K|x - y|^\beta$$

および

$$|a(t, x)| + |b(t, x)| \leq K(1 + |x|)$$

が成り立つと仮定する. このとき, 確率微分方程式 (3.2) はただ一つ解をもつ.

この結果は [24] の第 4 章定理 3.1 系 1 と定理 7.1 を組み合わせると従うが,⁴ 応用上非常に有用である. 実際, 本稿で説明したほとんどの確率微分方程式モデルの解の存在と一意性がこの結果から従う.

⁴[19], Chapter IX, Theorem 3.5 も参照のこと.

参考文献

- [1] N. Abourashchi, A. Veretennikov, “On exponential mixing bounds and convergence rate for Student processes”, *Theory of Probability and Mathematical Statistics* **81** 1–13 (2010).
- [2] L. Andersen, J. Andreasen, “Volatility skews and extensions of the Libor market model”, *Applied Mathematical Finance* **7** 1–32 (2000).
- [3] B. M. Bibby, I. M. Skovgaard, M. Sørensen, “Diffusion-type models with given marginal distribution and autocorrelation function”, *Bernoulli* **11** 191–220 (2005).
- [4] J. C. Cox, J. E. Ingersoll, S. A. Ross, “A theory of the term structure of interest rates”, *Econometrica* **53** 385–407 (1985).
- [5] F. Delbaen, H. Shirakawa, “An interest rate model with upper and lower bounds”, *Asia-Pacific Financial Markets* **9** 191–209 (2002).
- [6] 飛田武幸 著, 「ブラウン運動」, 岩波書店 (1975).
- [7] S. M. Iacus, N. Yoshida, “Simulation and Inference for Stochastic Processes with YUIMA”, Springer (2018).
- [8] S. Ikeda, S. Watanabe, “Stochastic differential equations and diffusion processes”, 2nd edition, North-Holland/Kodansha (1989).
- [9] O. Isah, “Solving SABR in exact form and unifying it with LIBOR market model”, working paper, available at SSRN: <https://ssrn.com/abstract=1489428> (2009).
- [10] K. Itô, “On stochastic differential equations”, *Memoirs of the American Mathematical Society* **4** 1–51 (1951).
- [11] O. Kallenberg, “Foundations of Modern Probability”, Springer (1997).
- [12] I. カラザス, S. E. シュレーブ 著, 渡邊壽夫 訳, 「ブラウン運動と確率積分」, 丸善出版 (2012).
- [13] A. M. Kulik, N. N. Leonenko, “Ergodicity and mixing bounds for the Fisher-Snedecor diffusion”, *Bernoulli* **19** 2294–2329 (2013).
- [14] 楠岡成雄 著, 「確率解析 (数理経済学叢書)」, 知泉書館 (2018).
- [15] Y. A. Kutoyants, “Statistical inference for ergodic diffusion processes”, Springer (2004).
- [16] D. ラムベルトン, B. ラペール 著, 森平爽一郎 監修, 青木信隆, 岩村伸一, 大田和亨, 中川秀敏 訳, 「ファイナンスへの確率解析」, 朝倉書店 (2000).
- [17] N. N. Leonenko, N. Šuvak, “Statistical inference for Student diffusion process”, *Stochastic Analysis and Applications* **28** 972–1002 (2010).
- [18] 松本裕行 著, 「応用のための確率論・確率過程 (SGC ライブラリ 36)」, サイエンス社 (2004).
- [19] D. Revuz, M. Yor, *Continuous martingales and Brownian motion*, 3rd edition, Springer (1999).
- [20] 清水泰隆 著, 「保険数理と統計的方法 (理論統計学教程)」, 共立出版 (2018).
- [21] 佐藤健一 著, 「加法過程 (紀伊國屋数学叢書 33)」, 紀伊國屋書店 (1990).
- [22] 谷口説男, 松本裕行 著, 「確率解析 (確率論教程シリーズ 5)」, 培風館 (2013).
- [23] A. E. D. Veraart, L. A. M. Veraart, “Stochastic volatility and stochastic leverage”, *Annals of Finance* **8** 205–233 (2012).
- [24] 渡辺信三 著, 「確率微分方程式 (数理解析とその周辺 9)」, 産業図書 (1975).