

Rで計算ファイナンス (ワークショップ事前学習資料)

上村 昌司

この資料はワークショップを受講予定の方のための事前学習資料です。別添の参考資料『計算ファイナンス』から、基本的なRのプログラムを抽出しました。Rの扱いに慣れていない方は、『Rのインストール手引き』に従いRをインストールし、下記に掲載のRプログラムを実行してみてください。下記のRのプログラムは全て関数の形で記述されています。Rに関数を読み込み、それを実行できるようにしておいてください。この時点では理論的な背景は理解できていなくて構いません。

なお、本ワークショップでは参考資料『計算ファイナンス』の第1, 4, 7章を中心に扱います。この資料は講師が東京工業大学大学院イノベーションマネジメント研究科で行った講義「計算ファイナンス」の講義ノートをもとめたものです。ワークショップではこの参考資料に金利モデルの話題を付け加えて新たに編集し直したものを提供する予定です。

1 確率過程入門

つぎのプログラムはいろいろな価格変動モデルを記述する際の基礎となるブラウン運動の見本過程（サンプルパス）をシミュレーションするものである (p.7~p.12)^{*1}。

```
Bmotion <- function(mu, sigma){
  T <- 1          #最終時刻
  N <- 250       #[0,T] 区間の分割数
  Delta <- 1/N   #1区間の長さ
  X0 <- 0        #初期値

  X <- numeric(N) #Xベクトルの初期化
  X[1] <- X0

  for (i in 1:(N-1)){
    X[i+1] <- X[i] + mu * Delta + sigma * sqrt(Delta) * rnorm(1)
  }
  plot(X, type="l") #ベクトルXの折れ線グラフ
}
```

*1 この資料に記載されているページ数は、別添の参考資料『計算ファイナンス』におけるページ数です。

(0,1)-ブラウン運動の見本過程を描くには、上記の関数を R に読み込んだ後に

```
> Bmotion(0,1)
```

を実行すればよい。> は R のプロンプトである（入力する必要はない）。実行例は図 1.1。乱数を用いたシミュレーションによって見本過程を描いているため、実行する度に異なる見本過程が描かれる。

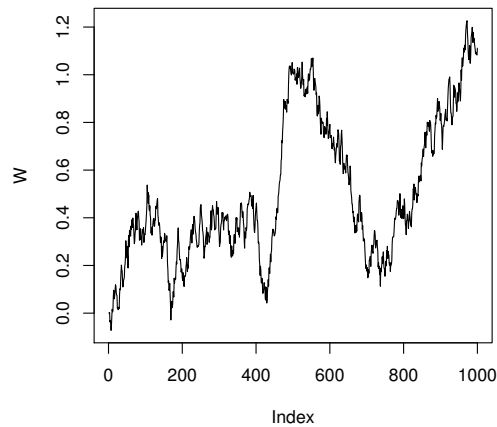


図 1.1 ブラウン運動の見本過程

つぎのプログラムは Black-Scholes モデル (p.10, 例題 1.4) の見本過程をシミュレーションするものである。

```
BS <- function(mu, sigma, X0){
  T <- 1 #最終時刻
  N <- 250 #[0,T] 区間の分割数
  Delta <- 1/N #1 区間の長さ
  #X0: X の初期値

  X <- numeric(N) #ベクトルの初期化
  X[1] <- X0

  for (i in 1:(N-1)){
    X[i + 1] <- X[i] + mu * X[i] * Delta + sigma * X[i] * sqrt(Delta) * rnorm(1)
  }
  plot(X, type="l") #ベクトル X の折れ線グラフ
}
```

例えば、例題 1.4 において $\mu = 0.1$, $\sigma = 0.2$, $X(0) = 100$ のときの見本過程をシミュレーションする場合には

```
> BS(0.1, 0.2, 100)
```

を実行すればよい。実行例は図 1.3。ただし、実行するごとに異なる見本過程が描かれることに注意せよ。

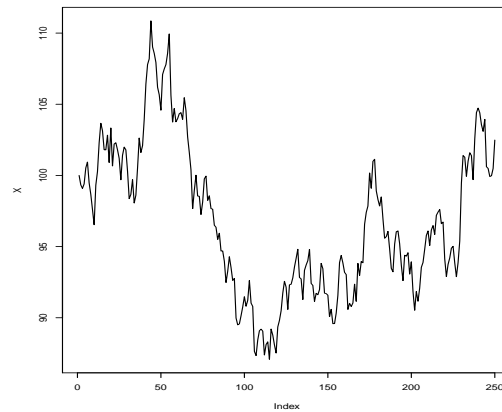


図 1.2 BS モデルの見本過程

つぎのプログラムは金利モデルでもよく使われる OU 過程 (p. 11, 例題 1.5) の見本過程をシミュレーションするものである。

```
OU <- function(lambda, barx, sigma, X0){  
  T <- 1 #最終時刻  
  N <- 250 #[0,T] 区間の分割数  
  Delta <- 1/N #1 区間の長さ  
  #X0: X の初期値  
  
  X <- numeric(N) #ベクトルの初期化  
  X[1] <- X0  
  
  for (i in 1:(N-1)){  
    X[i + 1] <- X[i] + lambda * (barx - X[i]) * Delta + sigma * sqrt(Delta) * rnorm(1)  
  }  
  plot(X, type="l") #ベクトル X の折れ線グラフ  
}
```

例えば、例題 1.5 (1.12) 式において $\lambda = 1$, $\bar{X} = 0.8$, $\sigma = 0.2$, $X(0) = 0.5$ のときを知りたい場合には

```
> OU(1, 0.8, 0.2, 0.5)
```

を実行すればよい。実行例は図 1.3。OU 過程は平均回帰性という特徴を持つ。この場合には平均回帰水準は \bar{X} となる。

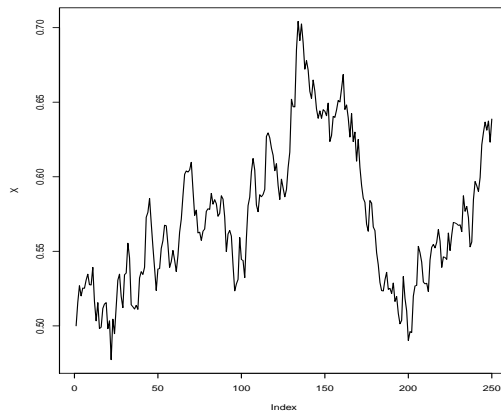


図 1.3 OU 過程の見本過程

2 オプション価格

つぎの関数はコールオプションの価格を求めるための Black-Scholes の価格公式である (p.14~p.16, (1.29) 式).

```
BScall <- function(r, sigma, T, K, S0){

  #r: 利子率
  #sigma: ボラティリティ
  #t: 現在時刻
  #T: 満期
  #K: 行使価格
  #S0: 現在の株価

  dplus <- 1 / (sigma * sqrt(T)) * (log(S0 / K) + (r + 0.5 * sigma^2) * T)
  dminus <- 1 / (sigma * sqrt(T)) * (log(S0/K)+(r - 0.5 * sigma^2) * T)

  price <- S0 * pnorm(dplus) - K * exp(-r * T) * pnorm(dminus)

  return(price)
}
```

例えば, $r = 0.1, S_0 = 62, K = 60, \sigma = 0.2, T = 5$ ヶ月 ときのコールオプションの現在価格は

```
> BScall(0.1,0.2,5/12,60,62)
[1] 5.797781
>
```

となる.

つぎの関数はコールオプションの価格をモンテカルロ法によって求めるものである. 結果が上の Black-Scholes 式によるものとほぼ一致することを確認せよ.

```
MCBScall <- function(r, sigma, T, K, X0){
  #r: 利率
  #sigma: ボラティリティ
  #T: 満期
  #K: 行使価格
  #X0: 現在の株価

  N <- 100      #[0,T] 区間の分割数
  Delta <- T/N  #1 区間の長さ
  Npath <- 10000 #生成するパスの数

  value <- numeric(Npath) #初期化

  for (j in 1:Npath){

    X <- numeric(N)
    X[1] <- X0

    for (i in 1:(N-1)) {
      X[i+1] <- X[i] + X[i] * r * Delta + X[i] * sigma * sqrt(Delta) * rnorm(1)
    }
    value[j] <- exp(-r * T) * max(X[i+1] - K, 0)
  }
  return(mean(value))
}
```

例えば, $r = 0.1, S_0 = 62, K = 60, \sigma = 0.2, T = 5$ ヶ月のときのコールオプションの現在価格は

```
> MCBScall(0.1,0.2,5/12,60,62)
[1] 5.789784
>
```

実行するたびにかなり異なった値が得られる. より正確な値を得るためには関数の中の変数 `Npath` の値を増やせばよい. ただし, `Npath` の値を大きくしすぎると, 計算時間が遅くなる.

R ではつぎのようなプログラムの方が計算が早い. ワークショップではこういった R で計算する際のコツについてもお話する予定である.

```
MCBScall2 <- function(r, sigma, T, K, X0){
```

```
Npath <- 100000
rnd <- rnorm(Npath)

X <- X0 * exp((r - 0.5 * sigma^2) * T + sigma * sqrt(T) * rnd)
return(mean(exp(-r*T) * pmax(X - K, 0)))
}
```