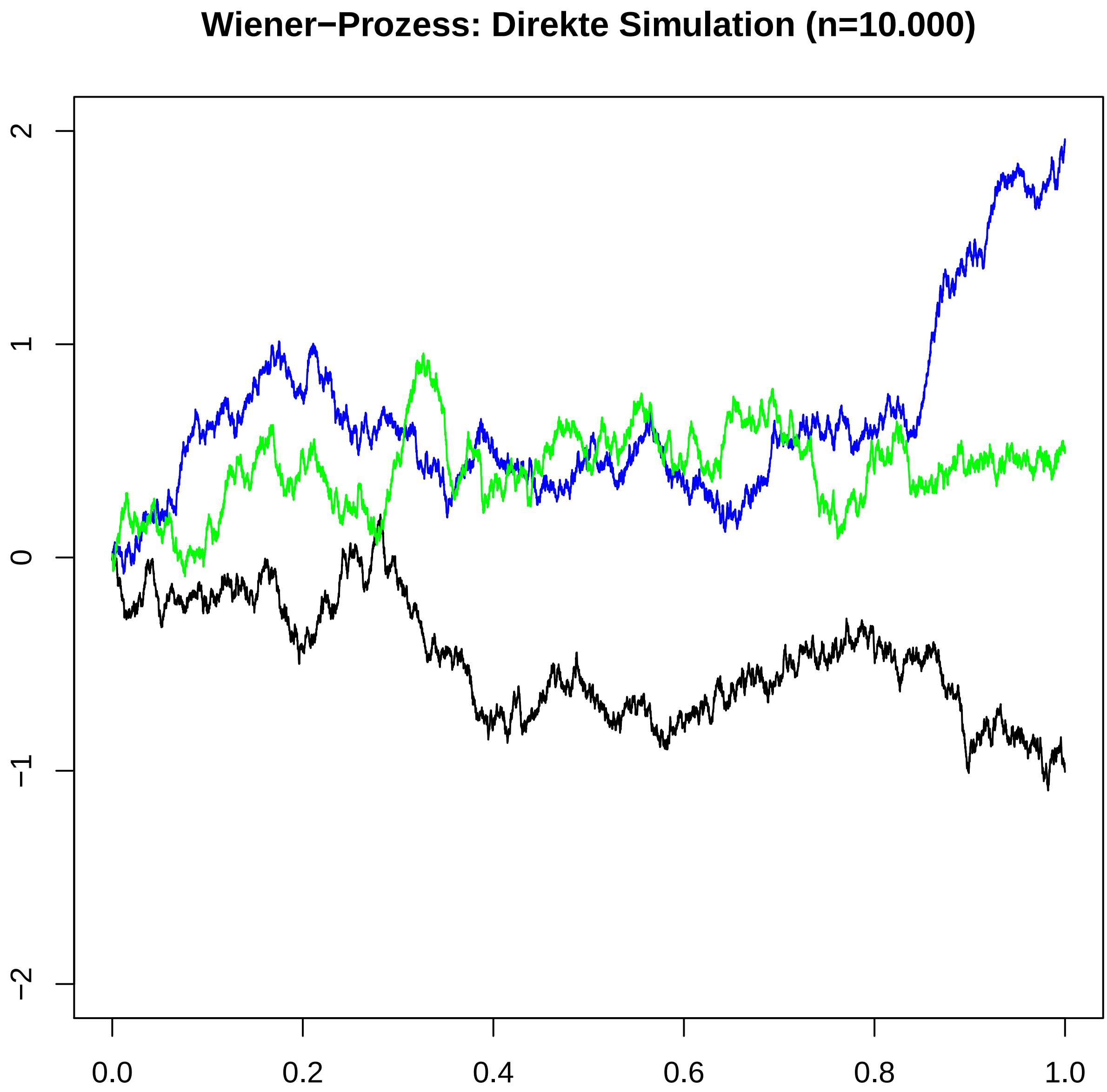
```r
# Stochastik II - Blatt 7 - Aufgabe 1 - Lösungsvorschlag

#Einstellen der Parameter
n <- 10000
delta <- 1/n

# Generieren der unabhängigen und N(0,1)-Zufallsvariablen
rand1 <- rnorm(n,0,1)
rand2 <- rnorm(n,0,1)
rand3 <- rnorm(n,0,1)

# Erstellen der Stützpunkte
x <- seq(0,1,by=1/n)

# Initialisieren der Prozesse
W1 <- W2 <- W3 <- rep(0,n+1)

# Zuweisung der Werte
for (k in 1:n+1) {
   W1[k] <- sqrt(delta)*sum(rand1[1:k])
   W2[k] <- sqrt(delta)*sum(rand2[1:k])
   W3[k] <- sqrt(delta)*sum(rand3[1:k])
}

# Zeichnen und abspeichern
pdf(file="brown_direkt.pdf", paper="a4")
plot(x, W1, type="l", lwd=1, xlab="", ylab="", ylim=c(-2,2), main="Wiener-Prozess: Direkte Simulation
(n=10.000)")
lines(x, W2, lwd=1, col="blue")
lines(x, W3, lwd=1, col="green")
dev.off()
```


Wiener−Prozess: Direkte Simulation (n=10.000)
2
1
0
−1
−2
0.0
0.2
0.4
0.6
0.8
1.0

```r
# Stochastik II - Blatt 7 - Aufgabe 3b,c - Lösungsvorschlag

# Definition der Haar-Funktion
haar <- function(t,m,j) {
   if (((j-1)/2^m <= t) & (t < (2*j-1)/2^(m+1))) { out <- 2^(m/2) }
   else if (((2*j-1)/2^(m+1) <= t) & (t < j/2^m)) { out <- -2^(m/2) }
   else {out <- 0}
   return(out)
}
haar0 <- function(t) {
   out <- 1
   return(out)
}

# Definition der Schauder-Funktion
schauder <- function(t,m,j) {
   if (((j-1)/2^m <= t) & (t < (2*j-1)/2^(m+1))) { out <- 2^(m/2)*t - (j-1)/2^(m/2) }
   else if (((2*j-1)/2^(m+1) <= t) & (t < j/2^m)) { out <- -2^(m/2)*t + j/2^(m/2) }
   else {out <- 0}
   return(out)
}
schauder0 <- function(t) {
   out <- t
   return(out)
}

# Plotten der Funktionen: Haar-Funktion
n <- 1000
x <- seq(0,0.999,by = 0.001)
y <- rep(0,999)
pdf(file="haar.pdf", paper="a4")
par(mfrow = c(4,4))
for(i in 1:n) { y[i] <- haar0(x[i])}
plot(x,y, type="l", xlab=c(), ylab=c(),ylim=c(0,1))
plot.new()
plot.new()
plot.new()
for(i in 1:n) { y[i] <- haar(x[i],0,1)}
plot(x,y, type="l", xlab=c(), ylab=c(),ylim=c(-1,1))
plot.new()
plot.new()
plot.new()
for(i in 1:n) { y[i] <- haar(x[i],1,1)}
plot(x,y, type="l", xlab=c(), ylab=c(),ylim=c(-sqrt(2),sqrt(2)))
for(i in 1:n) { y[i] <- haar(x[i],1,2)}
plot(x,y, type="l", xlab=c(), ylab=c(),ylim=c(-sqrt(2),sqrt(2)))
plot.new()
plot.new()
for(i in 1:n) { y[i] <- haar(x[i],2,1)}
plot(x,y, type="l", xlab=c(), ylab=c(),ylim=c(-2,2))
for(i in 1:n) { y[i] <- haar(x[i],2,2)}
plot(x,y, type="l", xlab=c(), ylab=c(),ylim=c(-2,2))
for(i in 1:n) { y[i] <- haar(x[i],2,3)}
plot(x,y, type="l", xlab=c(), ylab=c(),ylim=c(-2,2))
for(i in 1:n) { y[i] <- haar(x[i],2,4)}
plot(x,y, type="l", xlab=c(), ylab=c(),ylim=c(-2,2))
dev.off()

# Plotten der Funktionen: Schauder-Funktion
n <- 1000
x <- seq(0,0.999,by = 0.001)
y <- rep(0,999)
pdf(file="schauder.pdf", paper="a4")
```

```
par(mfrow = c(4,4))
for(i in 1:n) { y[i] <- schauder0(x[i])}
plot(x,y, type="l", xlab=c(), ylab=c(),ylim=c(0,1))
plot.new()
plot.new()
plot.new()
for(i in 1:n) { y[i] <- schauder(x[i],0,1)}
plot(x,y, type="l", xlab=c(), ylab=c(),ylim=c(0,1))
plot.new()
plot.new()
plot.new()
for(i in 1:n) { y[i] <- schauder(x[i],1,1)}
plot(x,y, type="l", xlab=c(), ylab=c(),ylim=c(0,1))
for(i in 1:n) { y[i] <- schauder(x[i],1,2)}
plot(x,y, type="l", xlab=c(), ylab=c(),ylim=c(0,1))
plot.new()
plot.new()
for(i in 1:n) { y[i] <- schauder(x[i],2,1)}
plot(x,y, type="l", xlab=c(), ylab=c(),ylim=c(0,1))
for(i in 1:n) { y[i] <- schauder(x[i],2,2)}
plot(x,y, type="l", xlab=c(), ylab=c(),ylim=c(0,1))
for(i in 1:n) { y[i] <- schauder(x[i],2,3)}
plot(x,y, type="l", xlab=c(), ylab=c(),ylim=c(0,1))
for(i in 1:n) { y[i] <- schauder(x[i],2,4)}
plot(x,y, type="l", xlab=c(), ylab=c(),ylim=c(0,1))
dev.off()

# Simulation
par(mfrow=c(1,1))
step <- 8
x <- seq(0,(2^step-1)/2^step, by = 1/2^step)
w1 <- w2 <- w3 <- rep(0,2^step)
z1 <- rnorm(2^step)
z2 <- rnorm(2^step)
z3 <- rnorm(2^step)
for (i in 1:(2^step-1)) { w1[i] <- w1[i] + schauder0(x[i]) * z1[1] }
for (i in 1:(2^step-1)) { w2[i] <- w2[i] + schauder0(x[i]) * z2[1] }
for (i in 1:(2^step-1)) { w3[i] <- w3[i] + schauder0(x[i]) * z3[1] }
for (m in 0:(step-1)) {
   for (j in 1:2^m) {
      for (i in 1:2^step) { # Stellen in Wiener
         w1[i] <- w1[i] + schauder(x[i],m,j) * z1[2^m+j]
         w2[i] <- w2[i] + schauder(x[i],m,j) * z2[2^m+j]
         w3[i] <- w3[i] + schauder(x[i],m,j) * z3[2^m+j]
      }
   }
}
pdf(file="brown.pdf",paper="a4")
plot(x[1:255],w1[1:255],ylab="",xlab="",ylim=c(-2,2),type="l",main="3 Simulationen des Wiener-
Prozesses")
lines(x[1:255],w2[1:255],col="blue")
lines(x[1:255],w3[1:255],col="green")
dev.off()
```

3 Simulationen des Wiener−Prozesses