Week_1

2022-10-19

C1.3

```
monte <- function (n,h) {
  norm1 <- pmax(rnorm(n,mean = 0, sd=1),0)
  norm2 <- pmax(rnorm(n,mean = 0 , sd=1+h),0)

  derivative <- 1/h*(mean(norm2)-mean(norm1))
  return(derivative)
}
set.seed(12345)
run_many <- c()
 for (x in 1:10000) {
  it <- monte(1000,h=0.1)
  run_many <- c(run_many,it)
}

bias <- mean(run_many)-1/sqrt(1/2*pi)
eff <- mean((run_many-mean(run_many))^2)
mse <- mean((run_many-1/sqrt(1/2*pi))^2)
```

The random observation has bias -0.4, efficiency 0.07 and mean square error 0.23. Setting h closer to 0 would likely reduce bias.

(ii) Now, use coupling method to define norm1 and norm2.

```
monte1 <- function (n,h) {
  normz <- rnorm(n,mean = 0, sd=1)
  norm1 <- pmax(normz,0)
  norm2 <- pmax(normz*(1+h),0)

  derivative <- 1/h*(mean(norm2)-mean(norm1))
  return(derivative)
}
set.seed(12345)
run_many <- c()
 for (x in 1:10000) {
  it <- monte(1000,h=0.1)
  run_many <- c(run_many,it)
}

bias <- mean(run_many)-1/sqrt(1/2*pi)
eff <- mean((run_many-mean(run_many))^2)
mse <- mean((run_many-1/sqrt(1/2*pi))^2)
```

The random observation has bias -0.4, efficiency 0.07 and mean square error 0.23.

Repetition 2: h=0.01

```
set.seed(12345)
run_many <- c()
 for (x in 1:10000) {
  it <- monte1(1000,h=0.01)
  run_many <- c(run_many,it)
}

bias <- mean(run_many)-1/sqrt(1/2*pi)
eff <- mean((run_many-mean(run_many))^2)
mse <- mean((run_many-1/sqrt(1/2*pi))^2)
```

The random observation has bias -0.4, efficiency 0 and mean square error 0.16.

Repetition 3: h=0.05

```
set.seed(12345)
run_many <- c()
 for (x in 1:10000) {
  it <- monte1(1000,h=0.05)
  run_many <- c(run_many,it)
}

bias <- mean(run_many)-1/sqrt(1/2*pi)
eff <- mean((run_many-mean(run_many))^2)
mse <- mean((run_many-1/sqrt(1/2*pi))^2)
```

The random observation has bias -0.4, efficiency 0 and mean square error 0.16.

Repetition 4: h=0.0001

```
set.seed(12345)
run_many <- c()
 for (x in 1:10000) {
  it <- monte1(1000,h=0.0001)
  run_many <- c(run_many,it)
}

bias <- mean(run_many)-1/sqrt(1/2*pi)
eff <- mean((run_many-mean(run_many))^2)
mse <- mean((run_many-1/sqrt(1/2*pi))^2)
```

The random observation has bias -0.4, efficiency 0 and mean square error 0.16.

Run same with monte (part (i))

Repetition 2: h=0.01

```
set.seed(12345)
run_many <- c()
 for (x in 1:10000) {
  it <- monte(1000,h=0.01)
  run_many <- c(run_many,it)
}

bias <- mean(run_many)-1/sqrt(1/2*pi)
eff <- mean((run_many-mean(run_many))^2)
mse <- mean((run_many-1/sqrt(1/2*pi))^2)
```

The random observation has bias -0.4, efficiency 6.64 and mean square error 6.8.

Repetition 3: h=0.05

```
set.seed(12345)
run_many <- c()
 for (x in 1:10000) {
  it <- monte(1000,h=0.05)
  run_many <- c(run_many,it)
}

bias <- mean(run_many)-1/sqrt(1/2*pi)
eff <- mean((run_many-mean(run_many))^2)
mse <- mean((run_many-1/sqrt(1/2*pi))^2)
```

The random observation has bias -0.4, efficiency 0.28 and mean square error 0.44.

Repetition 4: h=0.0001

```
set.seed(12345)
run_many <- c()
 for (x in 1:10000) {
  it <- monte(1000,h=0.0001)
  run_many <- c(run_many,it)
}

bias <- mean(run_many)-1/sqrt(1/2*pi)
eff <- mean((run_many-mean(run_many))^2)
mse <- mean((run_many-1/sqrt(1/2*pi))^2)
```

The random observation has bias -0.69, efficiency 6.57823×10^4 and mean square error 6.578277×10^4.