

# Assignment 3

Bjørn Christian Weinbach

2nd November, 2020

Clear R environment

```
rm(list = ls())
```

## Problem 1 Random number generation and Monte Carlo integration.

a)

In this problem the task is to sample  $n$  independent samples from the probability distribution

$$p(x) = \frac{2^{\frac{1}{4}} \Gamma\left(\frac{3}{4}\right)}{\pi} \exp\left(-\frac{x^4}{2}\right), \quad -\infty < x < \infty$$

For this, we can use random walk metropolis hastings. We import the function from the lectures.

```
# general 1d Gaussian proposal random walk MH
oneD.RWMH <- function(lprob, #notice log-density kernel!
                      sigma=1.0,
                      theta1=0.0,
                      n.iter=10000){
  # space for output
  output <- numeric(n.iter)
  # first iterate given
  output[1] <- theta1

  # main iteration loop
  for(t in 2:n.iter){
    # proposal
    thetaStar <- output[t-1] + rnorm(1,sd=sigma)
    # accept probability, for numerical stability we compute
    # the log-accept prob, and then take exp
    alpha <- exp(min(0.0,lprob(thetaStar)-lprob(output[t-1])))
    # accept/reject step
    if(runif(1)<alpha && is.finite(alpha)){
      output[t] <- thetaStar
    } else {
      output[t] <- output[t-1]
    }
  }
  return(output)
}
```

```

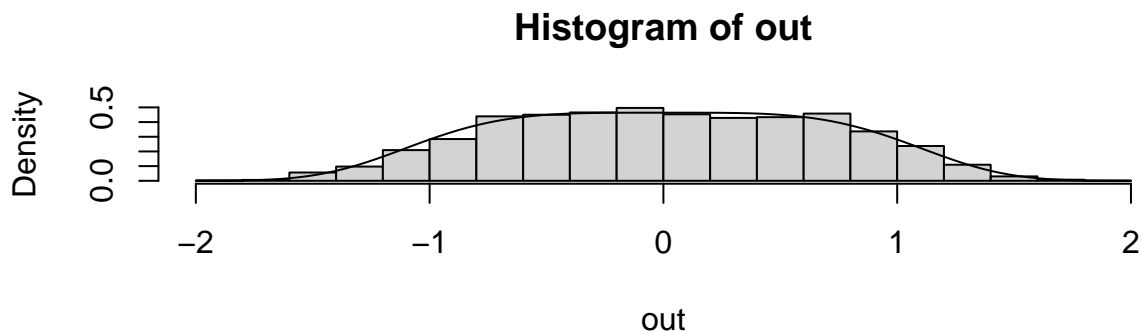
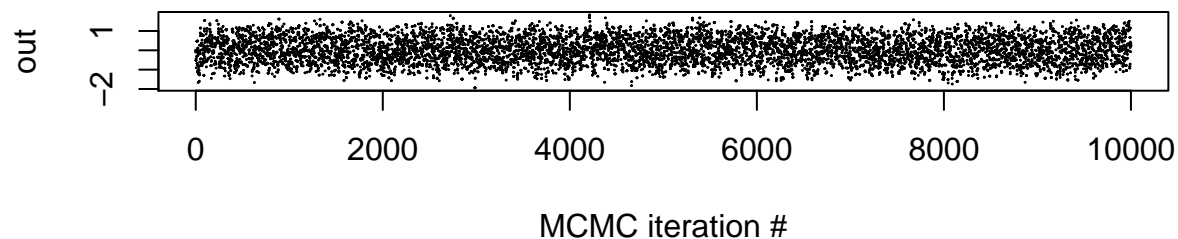
# Logarithm of pdf (needed for metropolis hastings from slides)
logpdf <- function(x){
  c <- log((2^0.25 * gamma(3/4)) / pi)
  return(c - x^4/2)
}

# pdf from mandatory pdf
pdf <- function(x) {
  return ((2^0.25*gamma(3/4)/pi) * exp(-x^4 / 2))
}

# sample n samples
n <- 10000
out <- oneD.RWMH(logpdf, theta1 = 0.0, sigma=1, n.iter = n)

# plot
par(mfrow=c(2,1))
plot(1:length(out),out,pch=20,cex=0.1,xlab="MCMC iteration #")
hist(out, probability = TRUE)
curve(pdf, add = TRUE, -2, 2)

```



We choose this algorithm since it is difficult to integrate the pdf to find the cdf for inverse sampling, but the logarithm of the pdf is quite simple and metropolis hastings random walk is quite effective in low dimensions and for distributions that are difficult to sample directly.

From the histogram where we plot the pdf we see that the sampling is quite effective. We also see from the plot that the random walk explores the distribution space quite nicely.

b)

Now let's sample from the distribution

$$p(x) = 2x \exp(-x^2), \quad 0 < x < \infty$$

This function seems to be quite simple to find the CDF for and direct sampling via inverse transform sampling is possible. We get

$$F(x) = \int_0^x p(x) = 1 - e^{-x^2}$$

And by the inverse sampling method, we can calculate

$$X = F^{-1}(U) = \pm \sqrt{-\ln(1-u)}$$

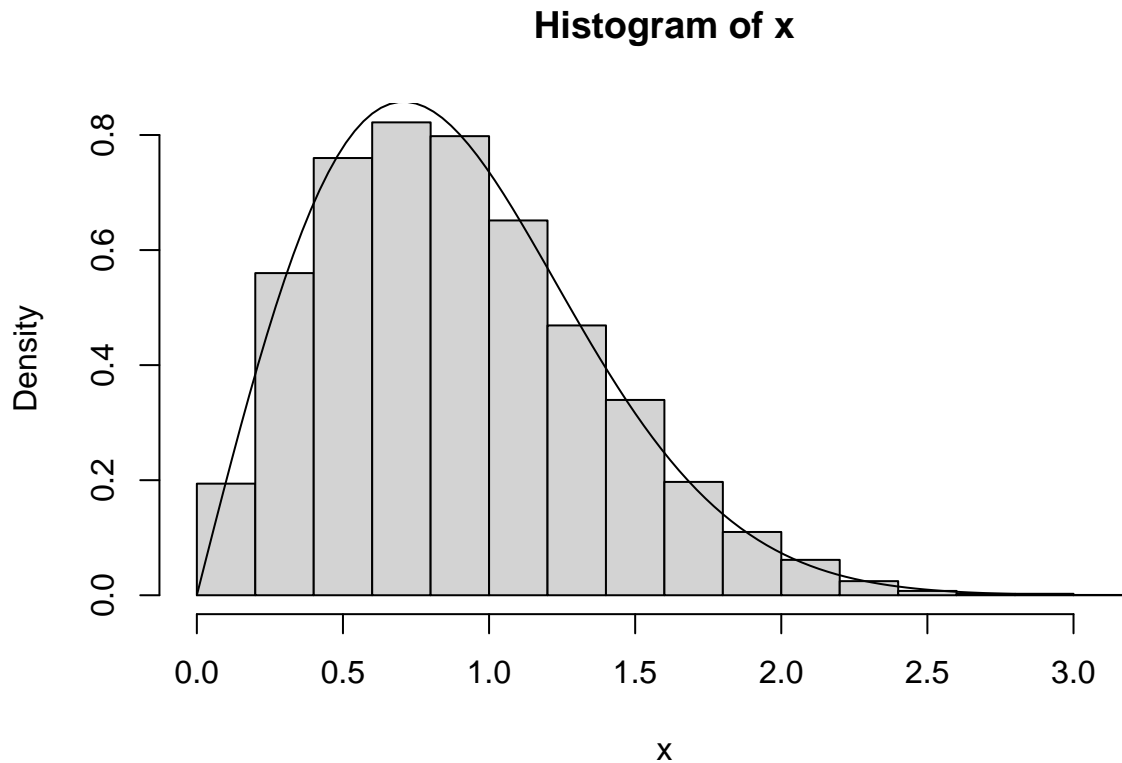
Given the nature of calculating square roots, we get positive and negative X values but since the support of the pdf clearly states that  $0 < x < \infty$  we reject the negative values.

in R:

```
# Use inverse sampling method to sample from
# p(x) = 2x exp(-x^2)
inverse_sampling <- function(Nsamples) {
  # Sample n samples from uniform distribution
  u <- runif(Nsamples)
  # Return X = F^-1(U)
  return(sqrt(-log(1-u)))
}

# PDF
pdf2 <- function(x) {
  return(2*x * exp(-x^2))
}

x <- inverse_sampling(10000)
hist(x, probability = TRUE)
curve(pdf2, 0, 3, add = TRUE)
```



c)

Now we will consider the integral

$$\int_0^{\infty} \exp(\sqrt{x}) \exp(-20(x-4)^2) dx$$

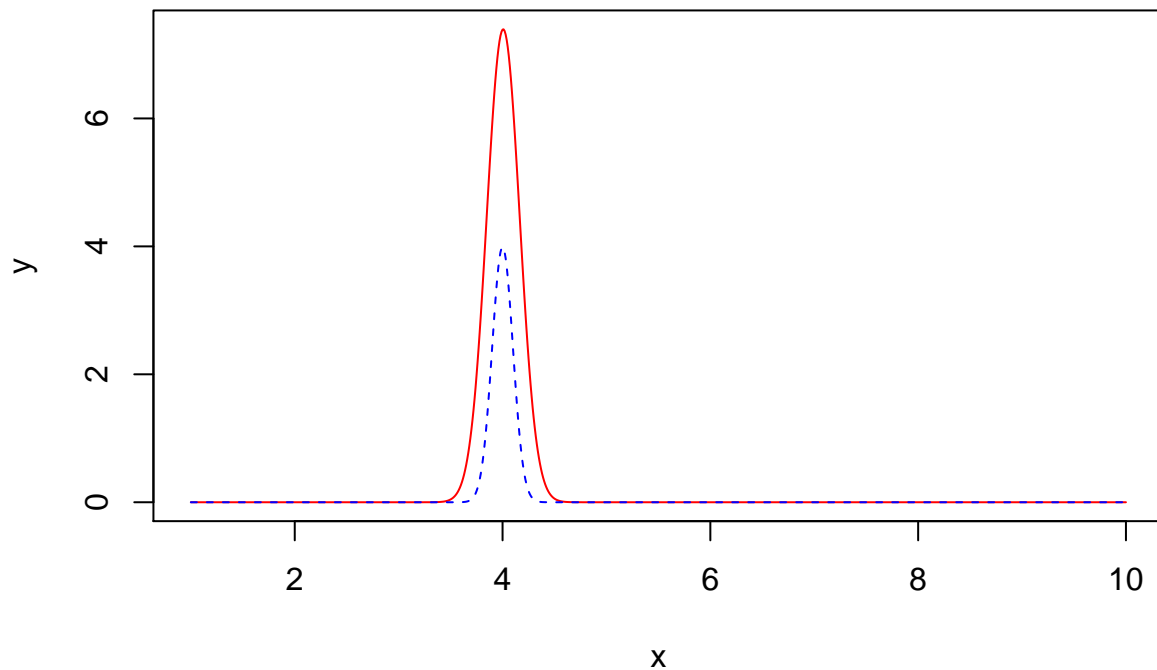
Let's evaluate the integral using importance sampling. First let's plot the function  $g(x)$  that is being integrated:

In R:

```
g <- function(x) {
  exp(sqrt(x)) * exp(-20*(x-4)^2)
}

f <- function(x) {
  dnorm(x, mean=4, sd=0.1)
}

x<-seq(1,10,0.01); y1=g(x); y2=f(x)
plot(x, y1, type="l", pch=19, col="red", xlab="x", ylab="y")
lines(x, y2, pch=18, col="blue", type="l", lty=2)
legend(1, 95, legend=c("Line 1", "Line 2"),
      col=c("red", "blue"), lty=1:2, cex=0.8)
```



With the function we want to integrate in  $g(x)$  in red and the  $f(x)$  in dashed blue. We want to calculate  $E\left(\frac{g(x)I(x \in A)}{f(x)}\right)$  by sampling  $X$  values from  $f$  and calculating the empirical mean of  $\frac{g(x)I(x \in A)}{f(x)}$  where  $I$  is the indicator function for the support.

```
importance <- function(Nsamp) {
  x <- rnorm(Nsamp, 4, 0.1)
  return(mean((g(x)*(x>0))/(f(x))))
}

# Estimate of integral using importance sampling
importance(100000)

## [1] 2.93383

# Numerical integration
integrate(g, 0, Inf)$value

## [1] 2.929669
```

## Bibliography