Problem 15.2

The code in HMC_UTurn.R uses simulates Hamiltonian dynamics for a single particle on the distri- bution described in the previous question:

In this question we will see how the efficiency of HMC depends on choice of the number of in- termediate steps. In particular we investigate the propensity of a particle undergoing Newtonian dynamics to perform U-Turns.

Get the code provided by author of HMC sampler

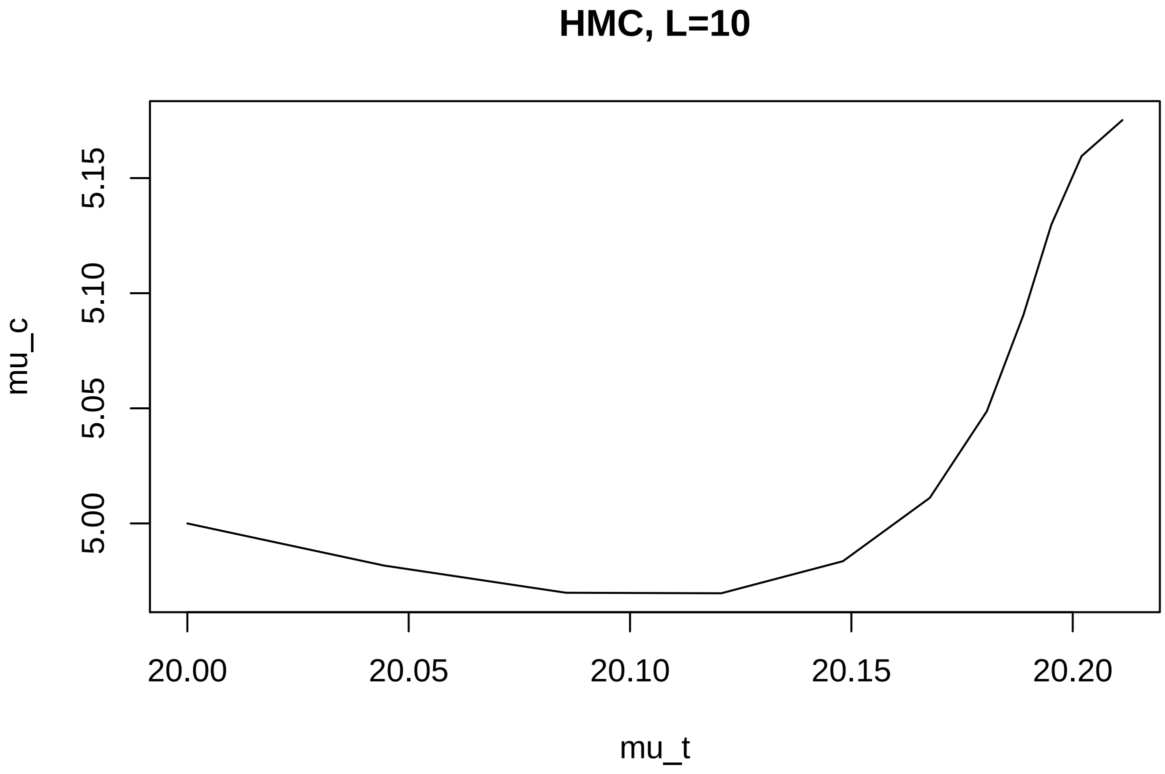
```
## Gradient of the potential with respect to x
fGradSimpleX <- function(x,y){
  aGrad <- -0.5 *exp(1/2 * (-(-20 + x) * (1.38889 * (-20 + x) -
                                            2.22222 * (-5 + y)) - (-2.22222 * (-20 + x) +
                                                                     5.55556 * (-5 + y)) * (-5 + y)) +
                       1/2 * ((-20 + x) * (1.38889 * (-20 + x) -
                                             2.22222 * (-5 + y)) + (-2.22222 * (-20 + x) +
                                                                      5.55556 * (-5 + y)) * (-5 + y))) *
  return(aGrad)
}

## Gradient of the potential with respect to y
fGradSimpleY <- function(x,y){
  aGrad <- -0.5 * exp(1/2 * (-(-20 + x) * (1.38889 * (-20 + x) -
                                             2.22222 * (-5 + y)) - (-2.22222 * (-20 + x) +
                                                                      5.55556 * (-5 + y)) * (-5 + y)) +
                        1/2 * ((-20 + x) * (1.38889 * (-20 + x) -
                                              2.22222 * (-5 + y)) + (-2.22222 * (-20 + x) +
                                                                       5.55556 * (-5 + y)) * (-5 + y)))



  return(aGrad)
}

## Gradient of U with respect to both coordinates
grad_U <- function(aQ){
  aGradX <- fGradSimpleX(aQ[[1]],aQ[[2]])
  aGrady <- fGradSimpleY(aQ[[1]],aQ[[2]])
  return(c(aGradX,aGrady))
}

## Potential function given by the -ve log of the posterior
U <- function(aQ){
  x <- aQ[[1]]
  y <- aQ[[2]]
  aU <- -log(0.265258 * exp(
    1/2 * (-(-20 + x) * (1.38889 * (-20 + x) -
                           2.22222 * (-5 + y)) - (-2.22222 * (-20 + x) +
                                                    5.55556 * (-5 + y)) * (-5 + y))))
  return(aU)
}

## Keeps all the positions of the particle over time
HMC_keep <- function (current_q, U, grad_U, epsilon, L, aSigma){
  q = current_q
  p = rnorm(length(q),0,aSigma)  # independent standard normal variates
  current_p = p

  # Make a half step for momentum at the beginning

  p = p - epsilon * grad_U(q) / 2

  # Alternate full steps for position and momentum

  lPosition <- matrix(nrow = (L+1),ncol = 2)
  lPosition[1,] <- q

  for (i in 1:L)
  {

    # Make a full step for the position

    q = q + epsilon * p
    lPosition[(i+1),] <- q

    # Make a full step for the momentum, except at end of trajectory

    if (i!=L) p = p - epsilon * grad_U(q)
  }

  # Make a half step for momentum at the end.

  p = p - epsilon * grad_U(q) / 2

  # Negate momentum at end of trajectory to make the proposal symmetric

  p = -p

  # Evaluate potential and kinetic energies at start and end of trajectory

  current_U = U(current_q)
  current_K = sum(current_p^2) / 2
  proposed_U = U(q)
  proposed_K = sum(p^2) / 2

  #   print(current_U-proposed_U)
  #   print(current_K-proposed_K)

  # Accept or reject the state at end of trajectory, returning either
  # the position at the end of the trajectory or the initial position
  r <- exp(current_U-proposed_U+current_K-proposed_K)
  # print(r)
  if (runif(1) < r)
  {
    return (list(q=q,pos=lPosition))  # accept
  }
  else
  {
    return (list(q=current_q,pos=lPosition))  # reject
  }
}
```

15.2.1

Simulate a single particle starting at (20, 5) for L = 10 steps with the following parameters = 0.18 (step size), = 0.18 (momentum proposal distribution width). Plot the path in parameter space.

```
set.seed(42)
lSamples_hmc <- HMC_keep(current_q = c(20,5),U = U, grad_U = grad_U, epsilon = 0.18,
                          L = 10, aSigma = 0.18)
```

```
plot(lSamples_hmc$pos, type = 'l', xlab = "mu_t", ylab = "mu_c", main = "HMC, L=10")
```



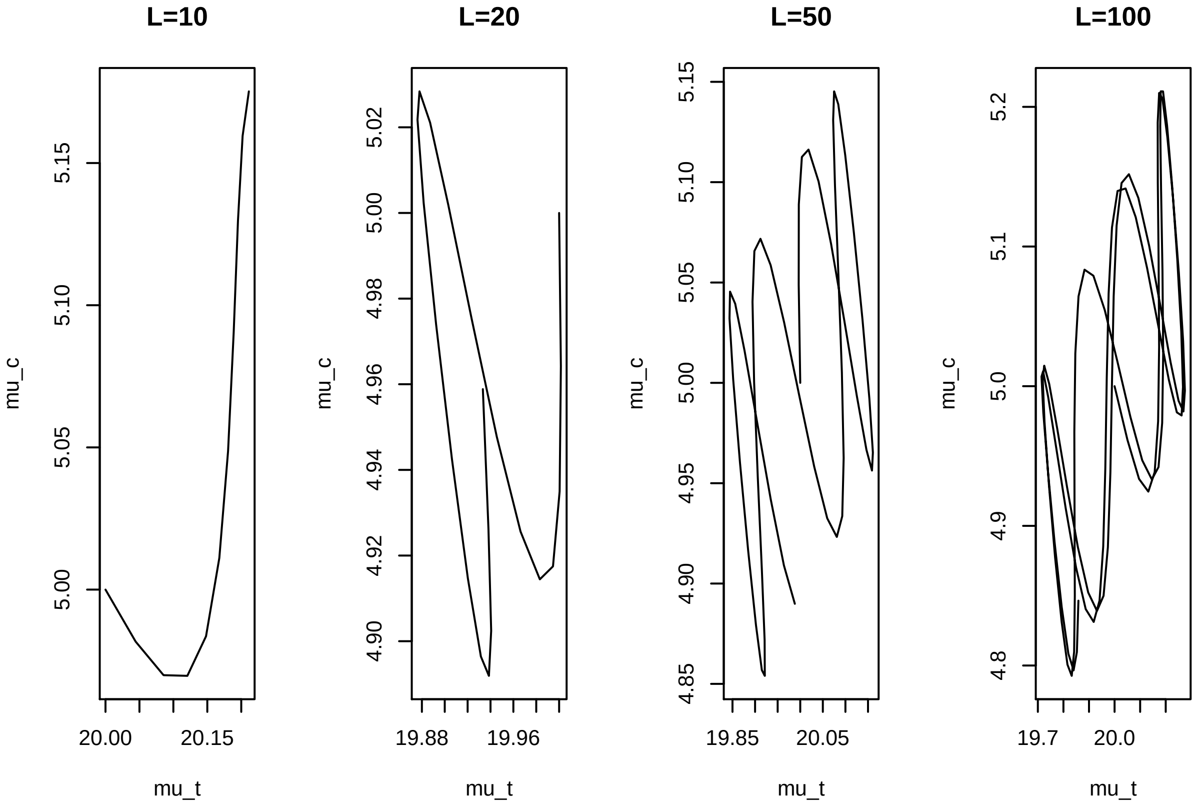
15.2.2

Now try L = 20,50,100 steps, again plotting the results what do you notice about the paths?

```
set.seed(42)
```

```r
lSamples_hmc_L10 <- HMC_keep(current_q = c(20,5),U = U, grad_U = grad_U, epsilon = 0.18,
                             L = 10, aSigma = 0.18)

lSamples_hmc_L20 <- HMC_keep(current_q = c(20,5),U = U, grad_U = grad_U, epsilon = 0.18,
                             L = 20, aSigma = 0.18)

lSamples_hmc_L50 <- HMC_keep(current_q = c(20,5),U = U, grad_U = grad_U, epsilon = 0.18,
                             L = 50, aSigma = 0.18)

lSamples_hmc_L100 <- HMC_keep(current_q = c(20,5),U = U, grad_U = grad_U, epsilon = 0.18,
                              L = 100, aSigma = 0.18)
```

```r
par(mfrow=c(1,4))
plot(lSamples_hmc_L10$pos, type = 'l', xlab = "mu_t", ylab = "mu_c", main = "L=10")

plot(lSamples_hmc_L20$pos, type = 'l', xlab = "mu_t", ylab = "mu_c", main = "L=20")

plot(lSamples_hmc_L50$pos, type = 'l', xlab = "mu_t", ylab = "mu_c", main = "L=50")

plot(lSamples_hmc_L100$pos, type = 'l', xlab = "mu_t", ylab = "mu_c", main = "L=100")
```



15.2.3.

Simulate 100 iterations of the particle starting at (20,5), with each particle running for L = 100 steps. Examine the motion of the particle in one of the parameter dimensions, and hence determine an optimal number of steps for this distribution.

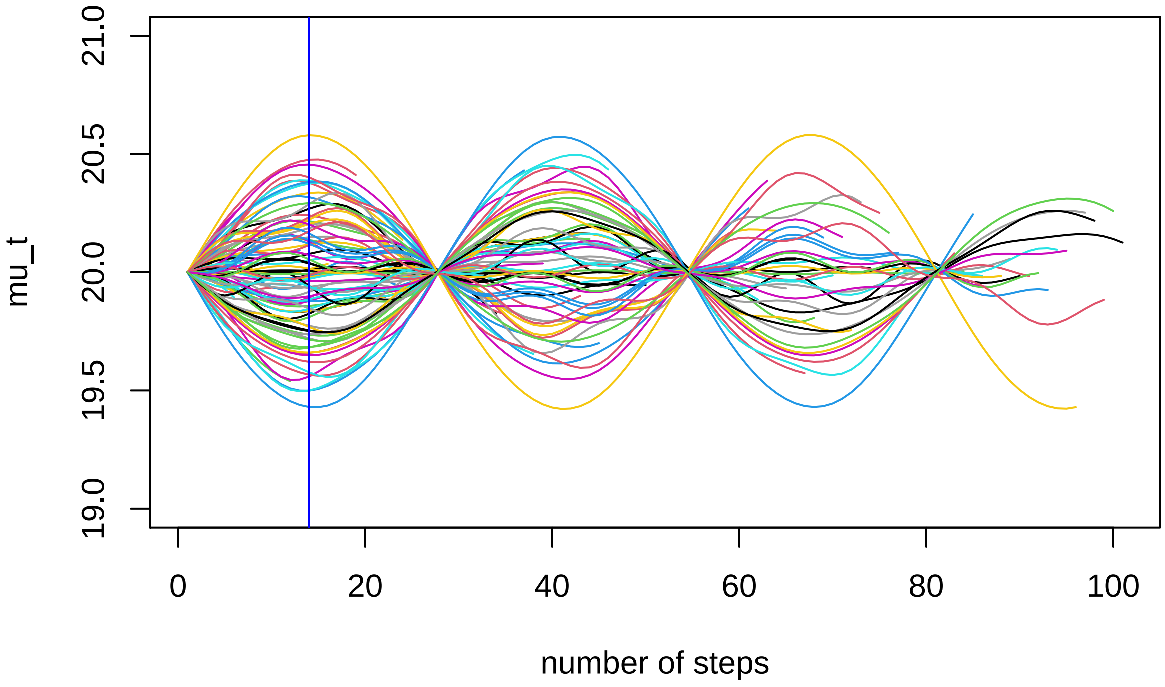
```
lSamples_hmc_v <- list()

for(it in seq(1:100)){
  lSamples_hmc_v[[it]] <- HMC_keep(current_q = c(20,5),U = U, grad_U = grad_U,
                                   epsilon = 0.18, L = it, aSigma = 0.18)$pos
}
```

```
plot(lSamples_hmc_v[[100]][,1], type="l", col=1, ylim = c(19, 21), xlab = "number of steps", ylab = "mu_
```

```
## Warning in title(...): conversion failure on 'Figure 15.12: The path of 100
## particle replicates over in the  t dimension' in 'mbcsToSbcs': dot substituted
## for <ce>

## Warning in title(...): conversion failure on 'Figure 15.12: The path of 100
## particle replicates over in the  t dimension' in 'mbcsToSbcs': dot substituted
## for <bc>
```

```
for (it in seq(2:100)){
  lines(lSamples_hmc_v[[it]][,1], col=it)
}

abline(v=14, col="blue") # After 14 U-turn starts
```

Figure 15.12: The path of 100 particle replicates over in the ..t dimens


mu_t
number of steps