

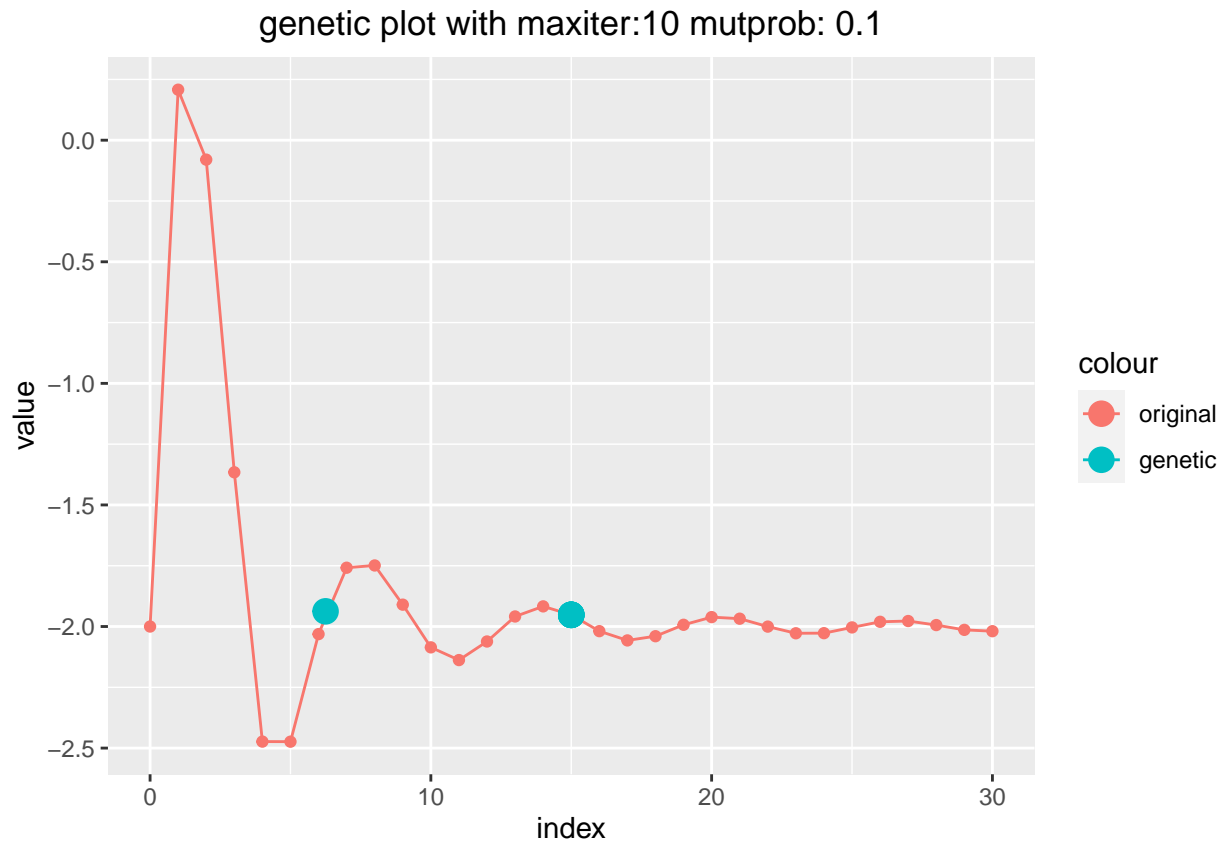
lab6

Wuhao Wang

12/2/2021

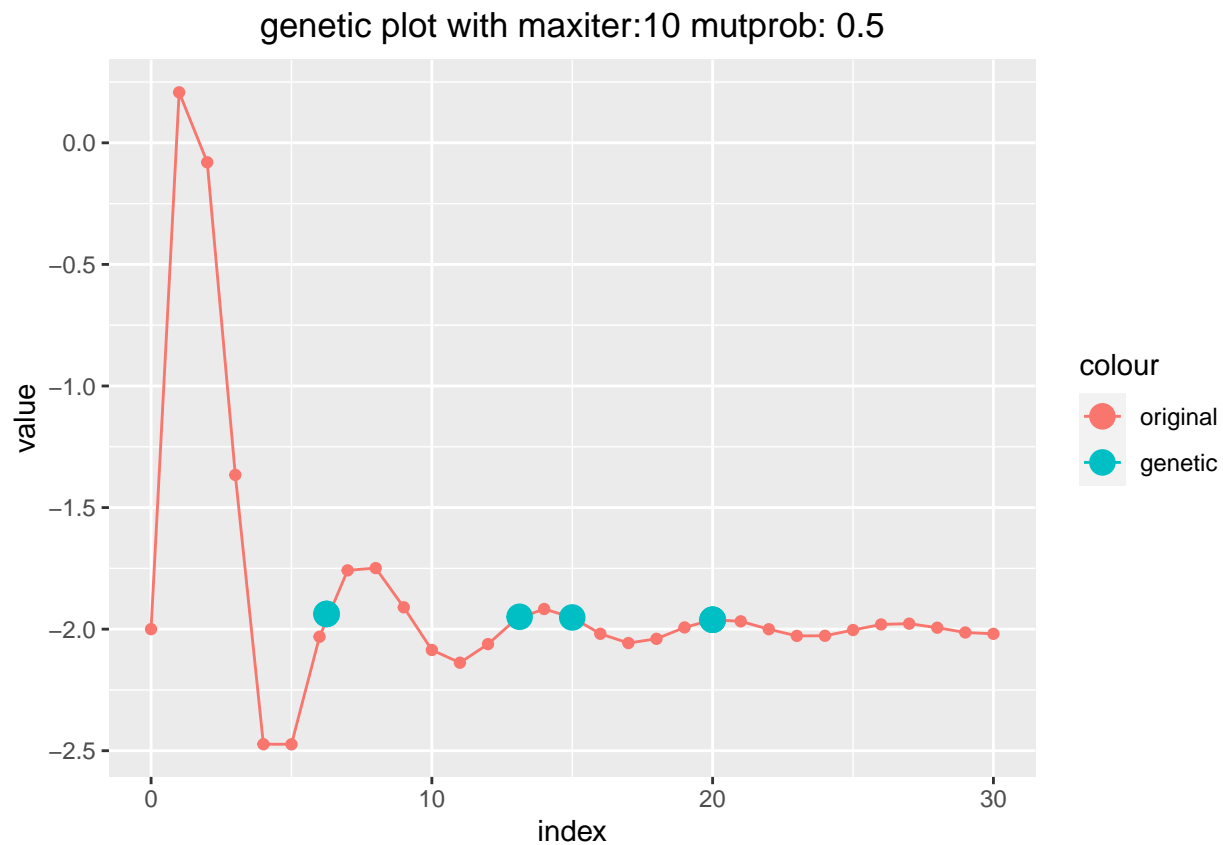
Question 1 Genetic algorithm

Implement the code mentioned in the lab6.pdf, we get results below. The number after the plot is the final observations.



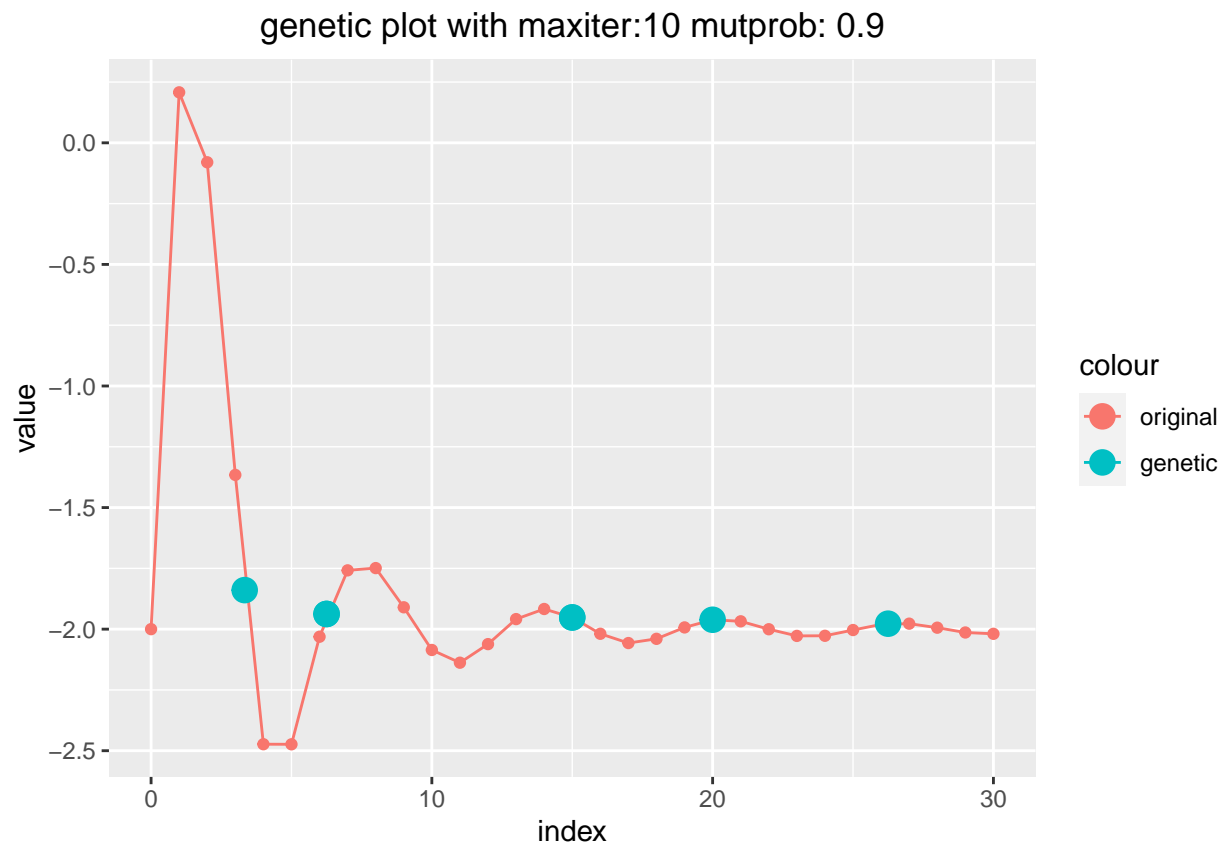
```
## [1] "final observation is :"
```

```
## [1] 15.00 15.00 15.00 15.00 15.00 6.25 15.00
```



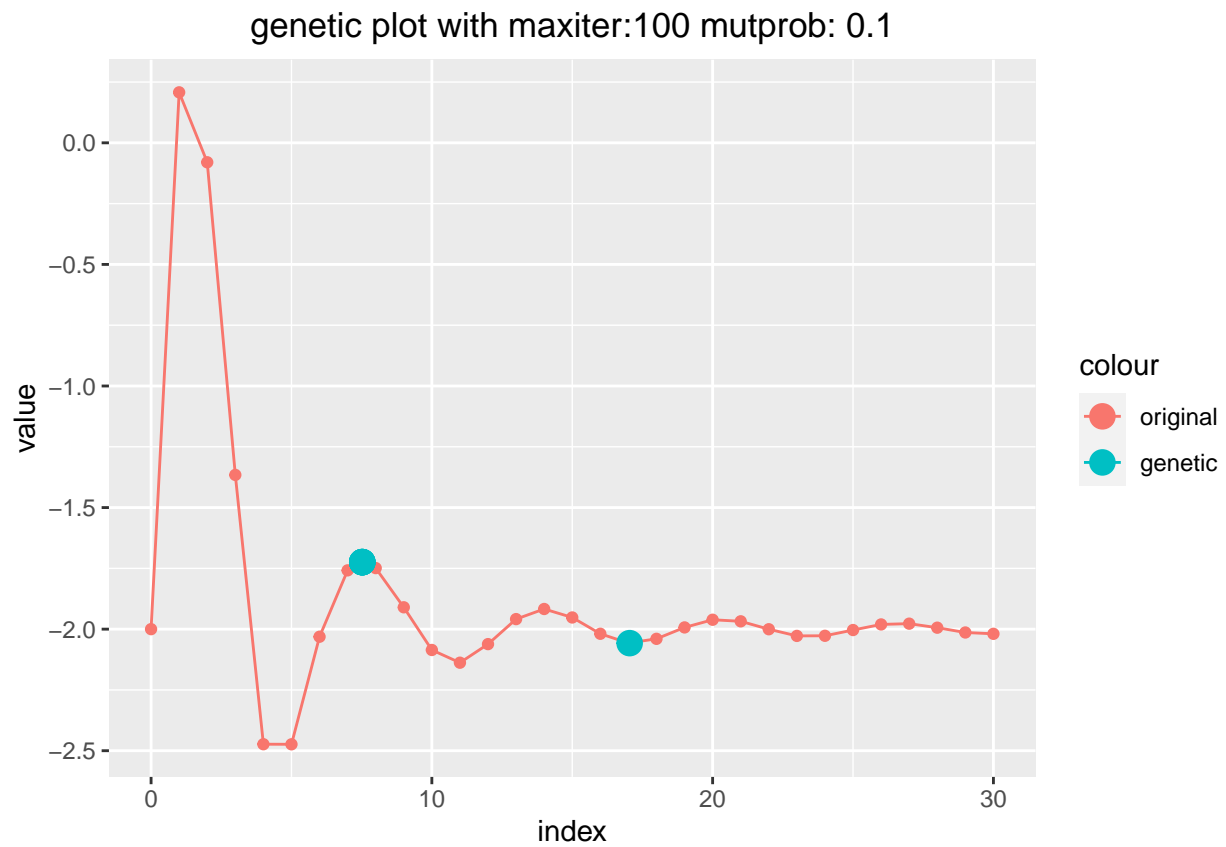
```
## [1] "final observation is :"
```

```
## [1] 13.125  6.250 20.000 15.000 20.000  6.250 20.000
```



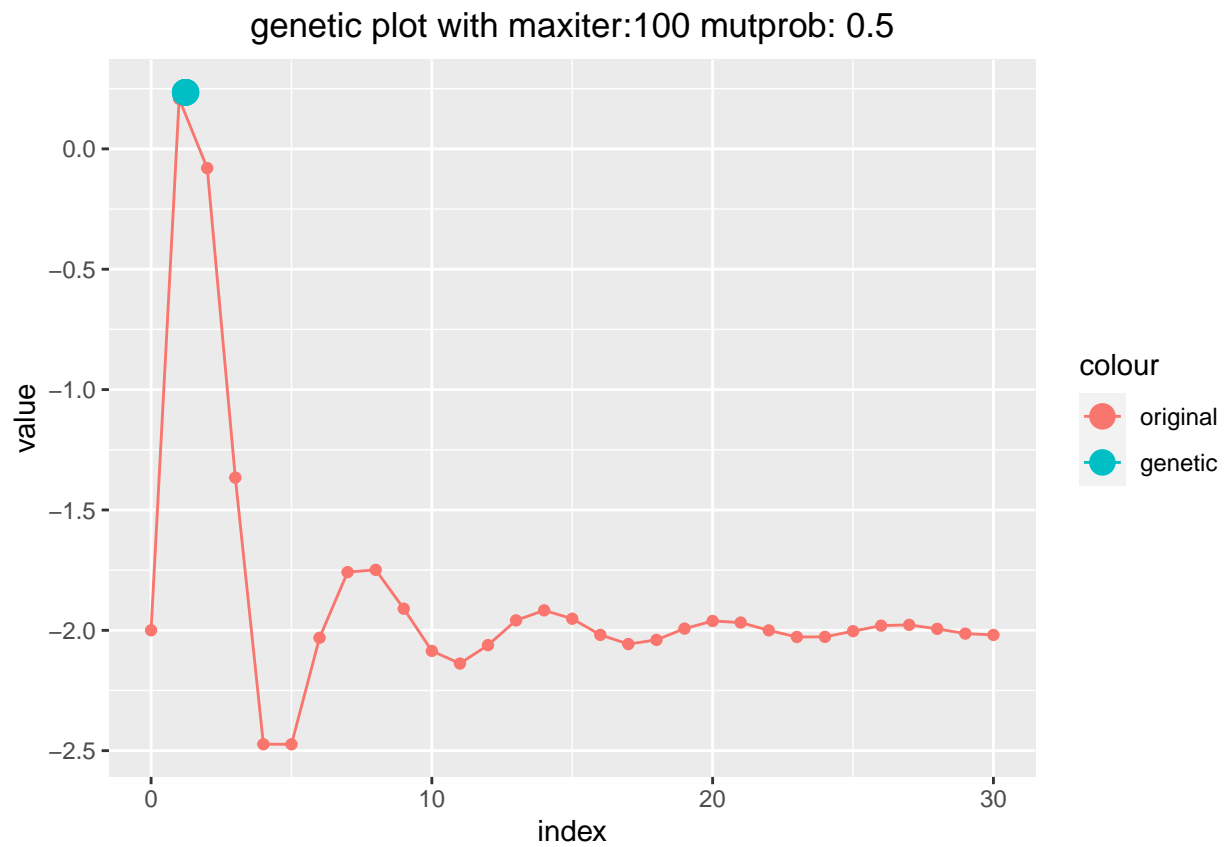
```
## [1] "final observation is :"
```

```
## [1] 6.250000 15.000000 6.250000 15.000000 20.000000 3.334045 26.250000
```



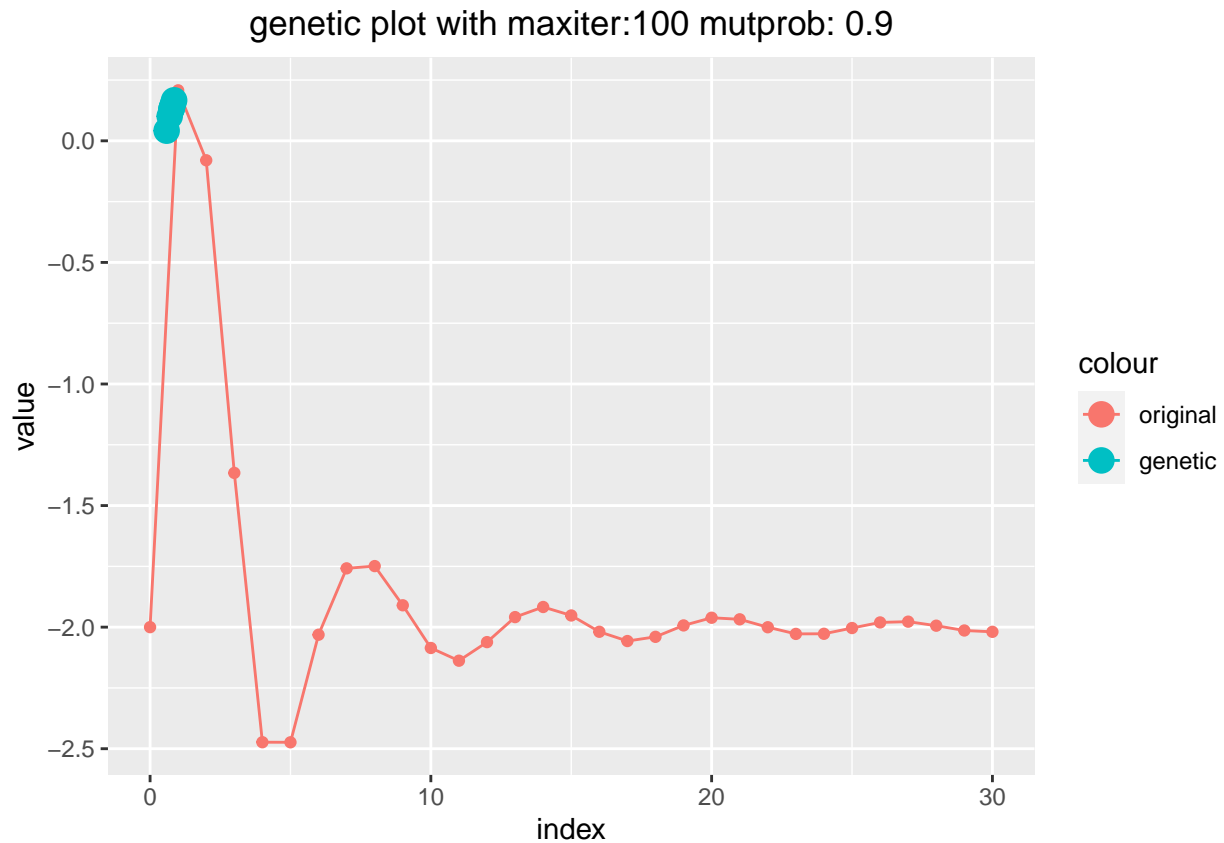
```
## [1] "final observation is :"
```

```
## [1] 7.521314 7.521448 7.521268 7.521180 17.044380 7.521109 7.521109
```



```
## [1] "final observation is :"
```

```
## [1] 1.227145 1.214345 1.239223 1.215068 1.215791 1.215791 1.247493
```



```
## [1] "final observation is :"
```

```
## [1] 0.7581143 0.8563168 0.6991026 0.8123496 0.7683824 0.7777097 0.5896889
```

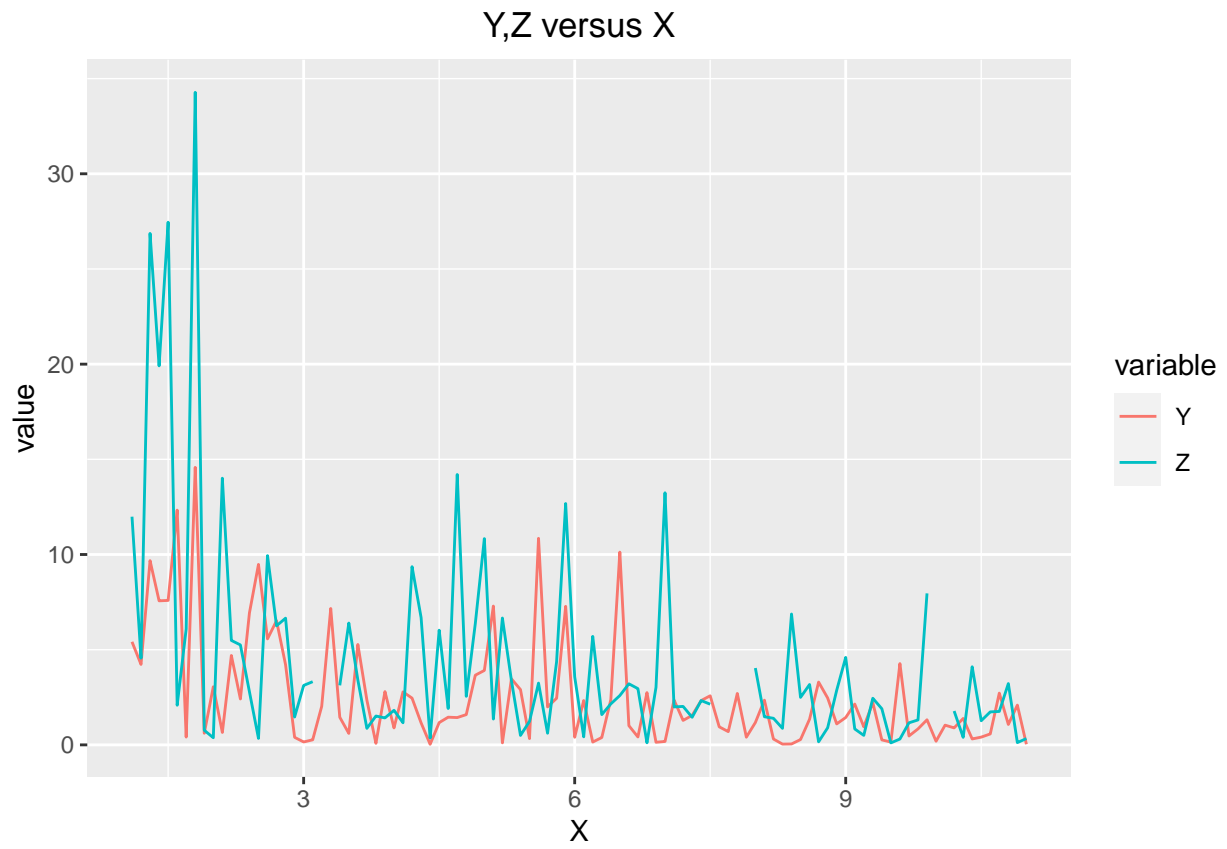
Conclusion:

The more iterations the higher probability, the easier for genetic algorithm to jump out of local optimal value. However, if the mutation probability is too high(e.g. mutation probability = 1), this algorithm will become random search, which is not a good method.

Question 2 EM algorithm

question 1 plot

Y and Z share the similar trend : both of them have 2 ‘peaks’ around $X = 1.5$ and $X = 6$. So, we would like to say that these twp processes are related to each other.





question 2 implement EM

Since we already have the relationship between λ^k and λ^{k+1} , we do not need to implement the log-likelihood function, what we need to do is implementing a iteration method. The results below is the trace of each iteration. The first column is the input λ before each iteration, the second column is the output λ after each iteration. We start from $\lambda_0 = 100$.

```
## [1] 100.00000 14.26782
## [1] 14.26782 10.83853
## [1] 10.83853 10.70136
## [1] 10.70136 10.69587
## [1] 10.69587 10.69566
```

From the result, we can see that the optimal λ is 10.69566, and we only need 5 iterations to find it.

APPENDIX

```
library(reshape2)
library(ggplot2)
library(gridExtra)
f1 <- function(x)
{
```



```

    return(x**2/exp(x)-2*exp(-9*sin(x)/(x**2+x+1)))
}

crossover <- function(x,y)
{
  return((x+y)/2)
}

mutate <- function(x)
{
  x**2%%30
}

f4 <- function(maxiter,mutprob)
{
  X <- 0:6*5
  values <- f1(X)
  extend <- c()
  for (i in 1:maxiter)
  {
    parents <- sample(X,2)
    victim_index <- order(values,decreasing = FALSE)[1]
    kid <- crossover(parents[1],parents[2])
    if(runif(1) >= 1-mutprob)
      kid <- mutate(kid)
    values[victim_index] <- f1(kid)
    X[victim_index] <- kid
    extend <- c(extend,max(values))
  }
  df <- data.frame(index = 0:30,value = c(f1(0:30)))
  df2 <- data.frame(index = X,value = values)
  p1<- ggplot(df,mapping = aes(x=index,y=value,color = 'Blue'))+
    geom_point()+
    geom_line()+
    geom_point(df2,mapping = aes(x=index,y=value,color = 'Red'),size=4)+
    xlab("index")+
    ylab("value")+
    ggtitle(paste0('genetic plot with maxiter:',maxiter,' mutprob: ',mutprob))+
    scale_color_discrete(labels=c("original","genetic"))+
    theme(plot.title = ggplot2::element_text(hjust=0.5))
  print(p1)
  print('final observation is :')
  return(X)
}

# output
f4(10,0.1)
f4(10,0.5)
f4(10,0.9)
f4(100,0.1)
f4(100,0.5)
f4(100,0.9)
#####
# QUESTION 2

```

```
#####
rm(list=ls())
data <- read.csv('physical1.csv')
df_melt <- melt(as.data.frame(data), id="X")
p1<- ggplot(df_melt,
  aes(x=X,y=value,color=variable))+
  geom_line()+
  xlab("X")+
  ylab("value")+
  ggtitle('Y,Z versus X')+
  theme(plot.title = ggplot2::element_text(hjust=0.5))

print(p1)
p2 <- ggplot(data = data.frame(X=data$X,variation=data$Y-data$Z), aes(x = X)) +
  geom_line(aes(y = variation)) +
  ggtitle("variation respect to X") +
  theme(plot.title = ggplot2::element_text(hjust=0.5))
print(p2)
EM<-function(eps,maxit){
  Zobs <- data$Z[!is.na(data$Z)]
  Zmiss <- data$Z[is.na(data$Z)]
  miss_num <- length(Zmiss)
  k <- 0
  llvalprev<-0
  llvalcurr<-100
  while ((abs(llvalprev-llvalcurr)>eps) && (k<(maxit+1))){
    llvalprev<-llvalcurr
    coe <- 1/(2*length(data$X))
    fir <- sum(data$X*data$Y)
    sec <- 0.5*sum(data$X[!is.na(data$Z)]*Zobs)+miss_num*llvalprev
    llvalcurr <- coe*(fir+sec)
    k<-k+1
    print(c(llvalprev,llvalcurr))
  }
}
EM(0.001,100)
```