U1

f<-function(x){

(x^2)/exp(x) - 2\*exp(-9\*sin(x)/(x^2+ x +1))

}

crossover<-function(x1,x2){

(x1+x2)/2

}

mutate<- function(x) {

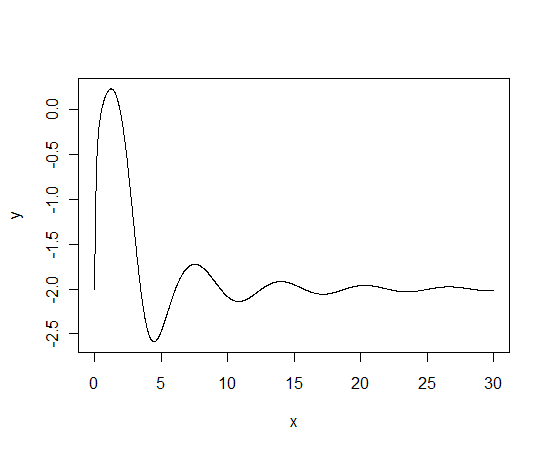
x^2%%30;

}

x=seq(0,30, by=0.01);

y=f(x);

plot(x,y, type='l');



genetic<-function(maxiter, mutprob){

x=seq(0,30, by=0.01);

y=f(x);

plot(x,y, type='l');

mypoints <-seq(0,30,by=5);

Values<-f(mypoints);

points(mypoints, Values, type="p")

maxiter=100;

for (i in 1:maxiter) {

sam<-sample(mypoints,2);

ord<-order(Values);

kid<-crossover(sam[1],sam[2]);

u<-runif(1,0,1);

if (u<0.4) {

kid<-mutate(kid)

};

mypoints[ord[1]]<-kid;

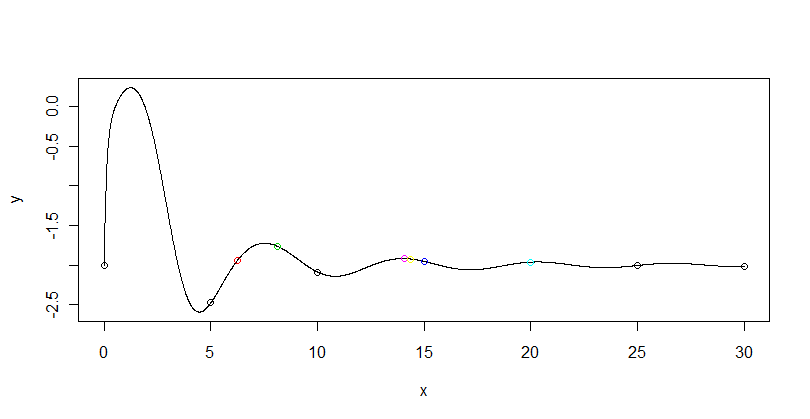
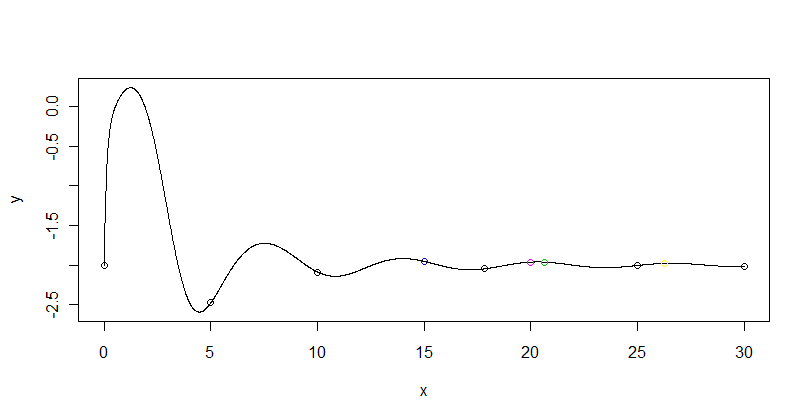
Values[ord[1]]<-f(kid);

}

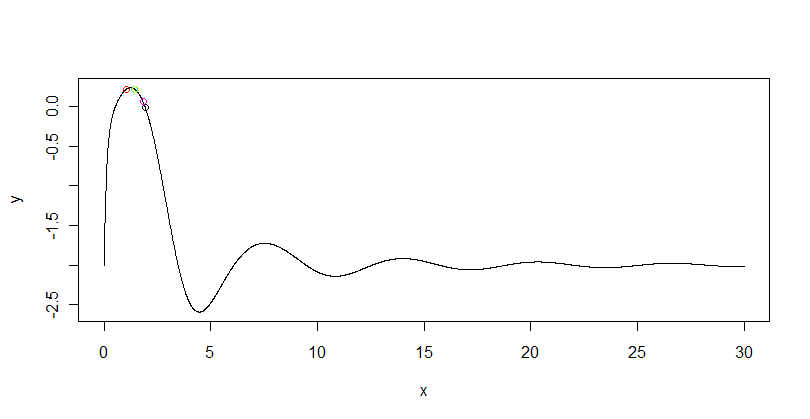
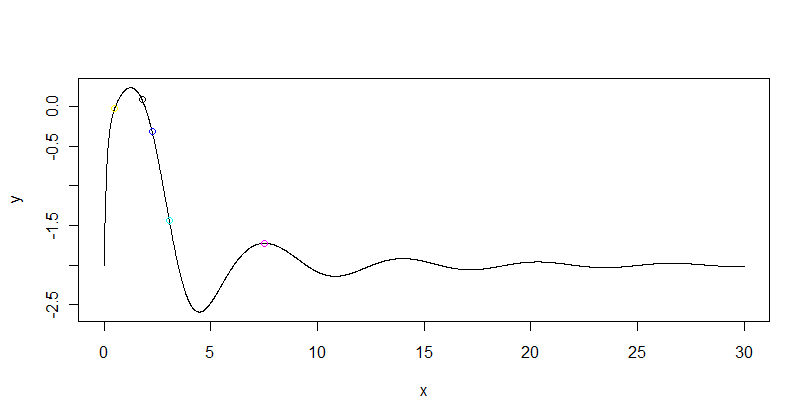
points(mypoints, Values, col=1:7);

}

genetic(10,0.1)

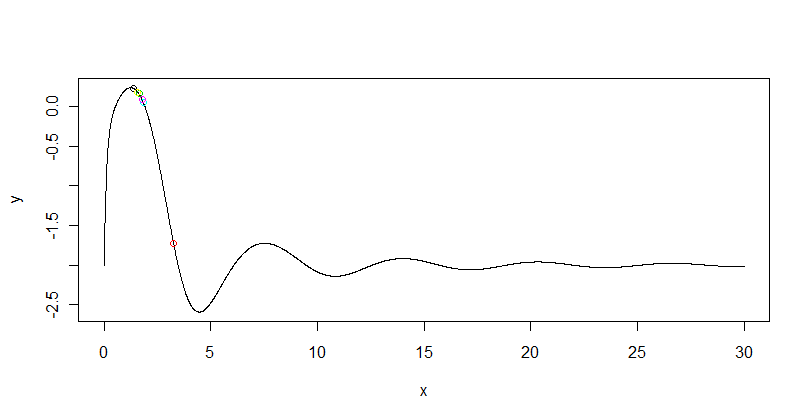
 

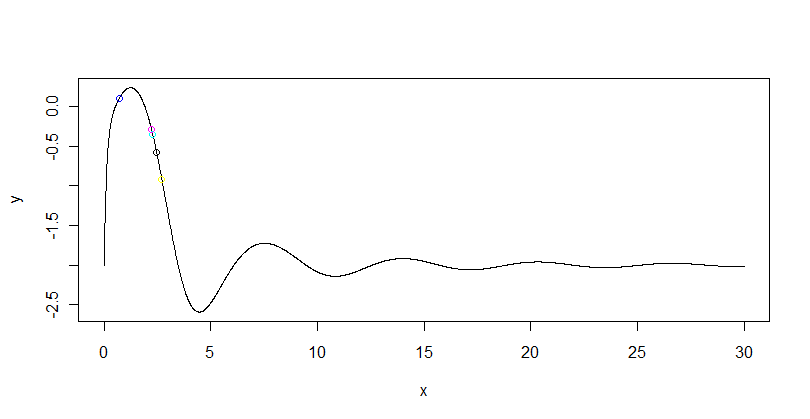
0.1 is unstable; 10 iteration rarely give optimum

0.1 is unstable; 100 iterations sometimes giv optimum, sometimes not. Sometimes found in local optimum

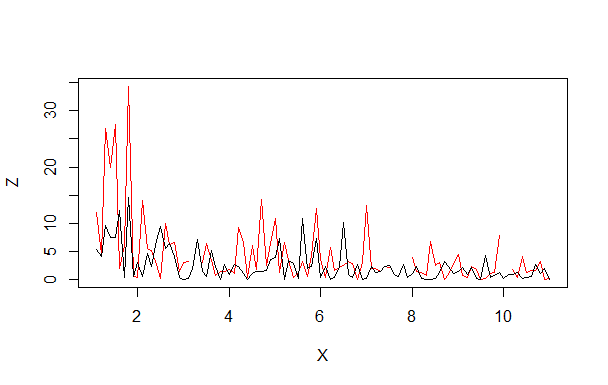
0.5: 10 iterations, sometimes in optimum sometimes in local optimum, 100iteratiions: sam3e story





0.9: 10 iter : optimum is not achieved, 100 it: almost always points in global optimum, but some points outside.

#uppgift 2



The processes seem to be related – when variation in the first increases, it also increases in the second. The variation seems to decrease with X.



[[1] 14.26782

[1] 10.83853

[1] 10.70136

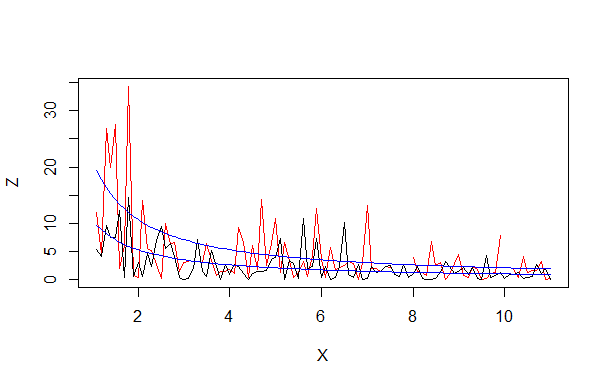
[1] 10.69587

[1] 10.69566

> print(lambda)

[1] 10.69566

5 iterations required



Blue lines seem to fit the original data nicely