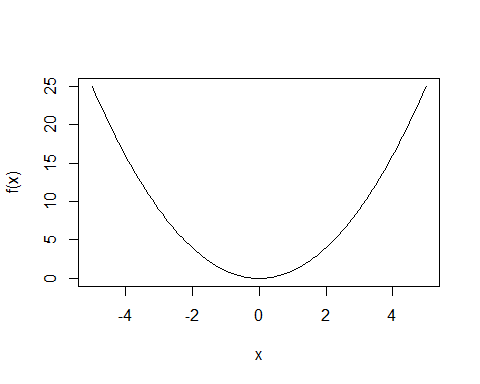
Particle Swarm Optimizer

Jack Thomas

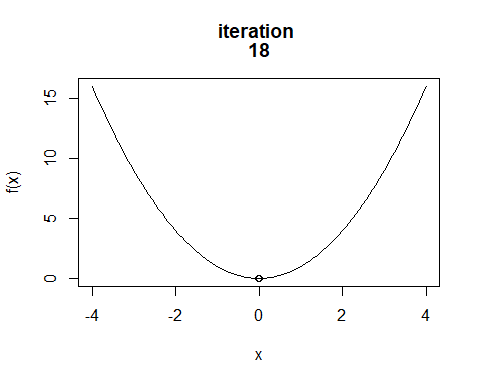
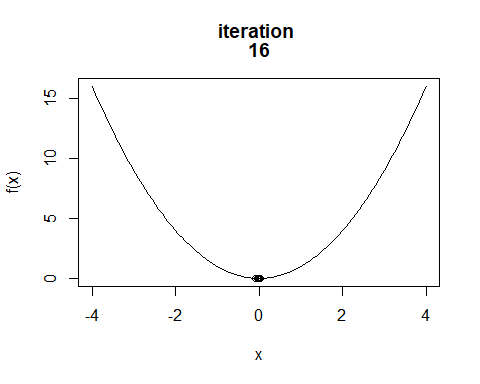
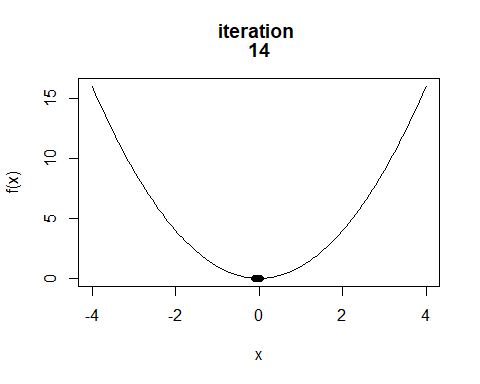
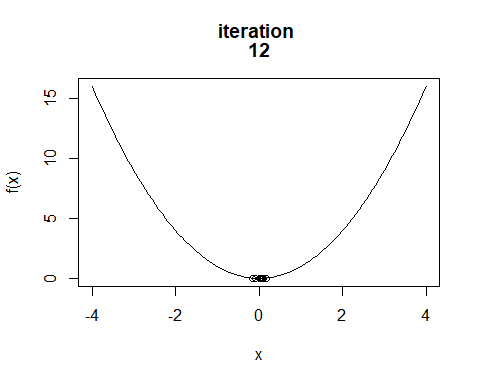
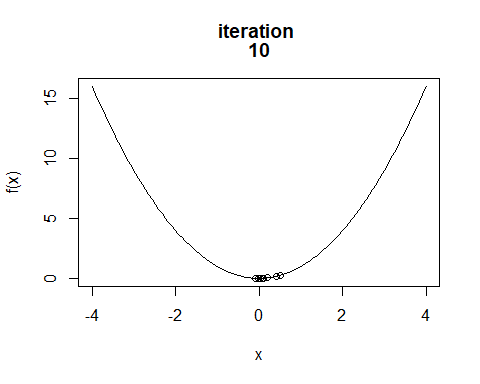
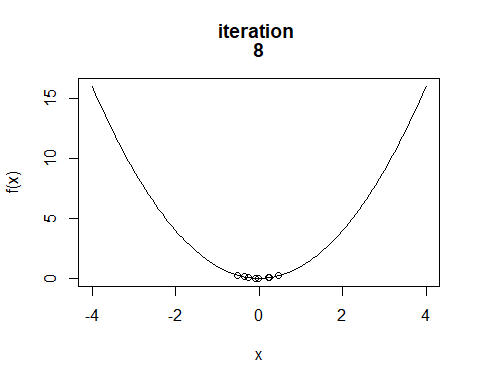
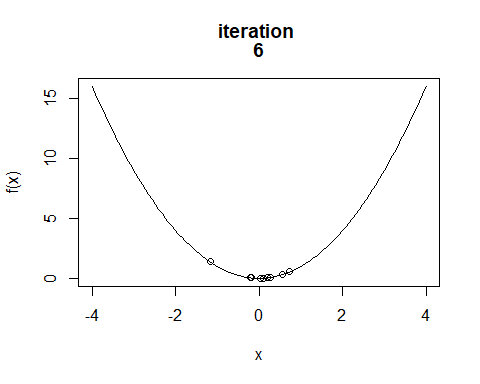
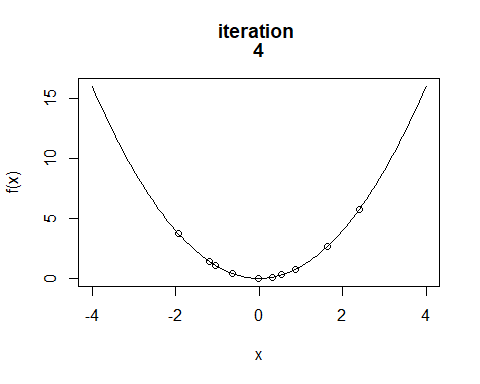
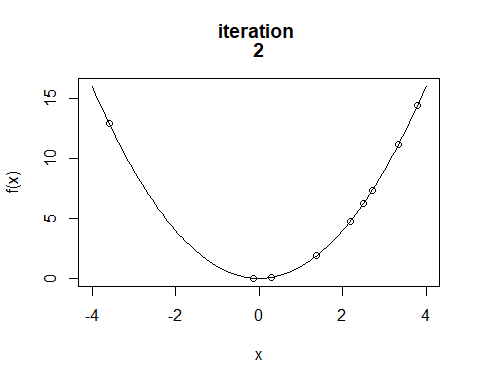
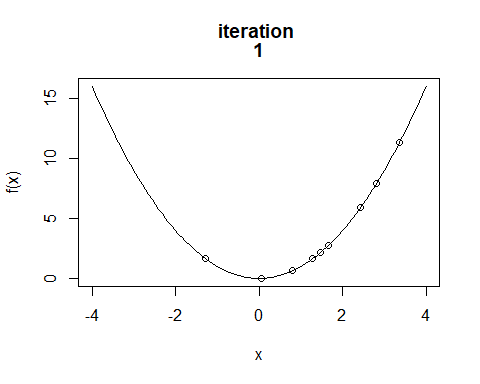
April 11, 2019

Trying out 2D first

set.seed(123)  
#Simple function  
f <- function(x){  
 return(x^2)  
}  
  
curve(f(x),-5,5)

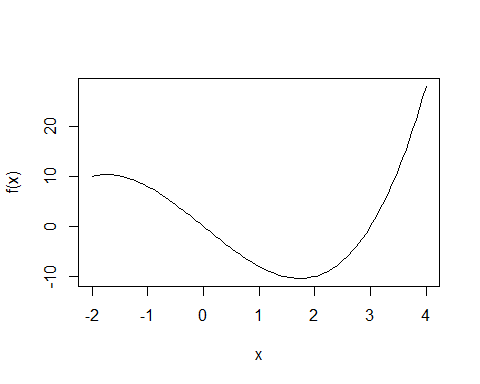


swarm <- function(S,lower, upper,omega,phiP,phiG){  
 iterations <- 100  
 best\_swarm <- lower  
 #particle positions  
 x <- rep(0,S)  
 #particle best positions  
 p <- rep(0,S)  
 #particle velocities  
 v <- rep(0,S)  
   
 #for each particle i   
 for (i in 1:S){  
 #initial position  
 x[i] <- runif(1,lower,upper)  
 p[i] <- x[i]  
   
 if(f(p[i]) < f(best\_swarm)){  
 best\_swarm <- p[i]  
 }  
   
 #initial velocity  
 v[i] <- runif(1,-abs(upper-lower),abs(upper-lower))  
 }  
 #print(x)  
 #print(p)  
 #print(v)  
 #print(best\_swarm)  
 for(j in 1:iterations){  
   
 for (i in 1:S){  
 rp <- runif(1,0,1)  
 rg <- runif(1,0,1)  
   
 v[i] <- omega\*v[i] + phiP\*rp\*(p[i]-x[i]) + phiG\*rg\*(best\_swarm-x[i])  
   
 x[i] <- x[i] + v[i]  
   
 if (f(x[i]) < f(p[i])){  
 p[i] <- x[i]  
   
 #update best position in the swarm  
 if (f(p[i]) < f(best\_swarm)){  
 best\_swarm <- p[i]  
 }  
 }  
 }  
 if((j%%2 == 0 && j < 20) || j == 1){  
 curve(f,lower,upper,main = c("iteration ",j))  
 points(x,f(x))  
 }  
   
 }  
 cat("best x value: ",best\_swarm,"\n")  
 cat("minimum: ",f(best\_swarm),"\n")  
 cat("\n")  
}  
  
swarm(10,-4,4,.5,.5,.5)

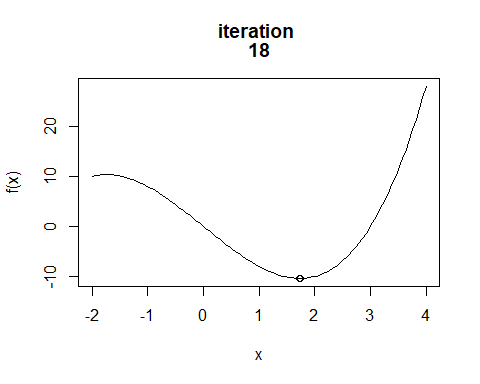
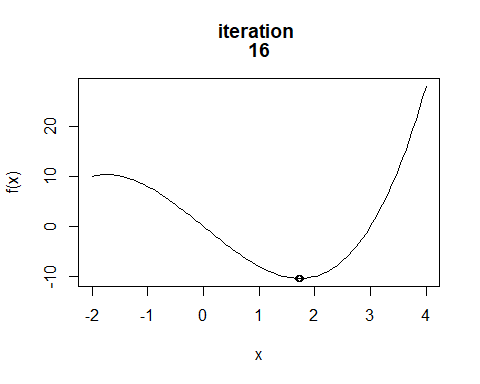
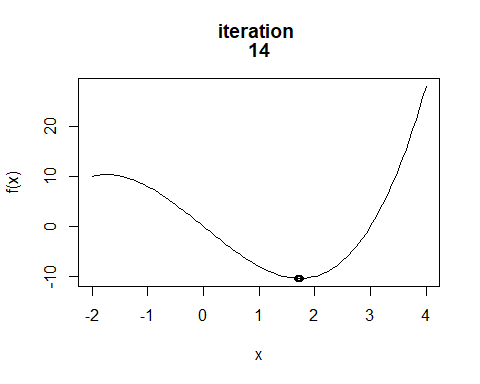
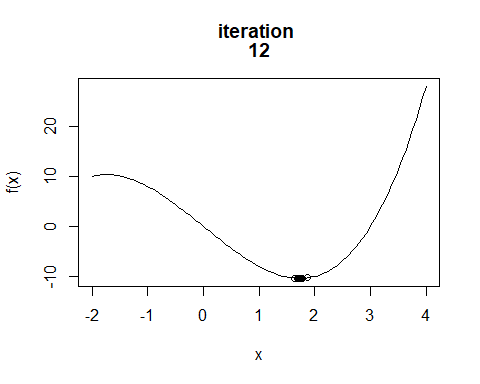
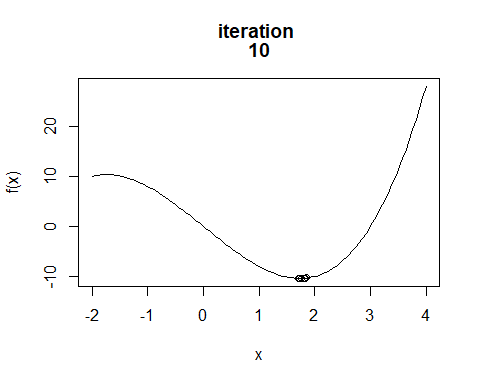
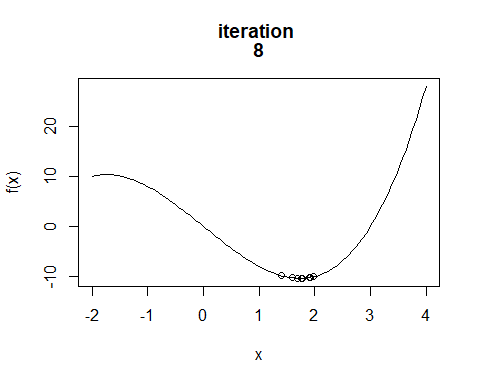
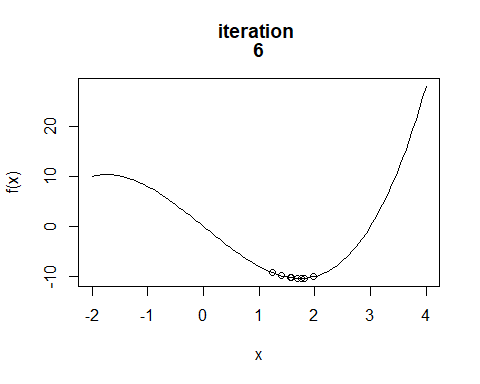
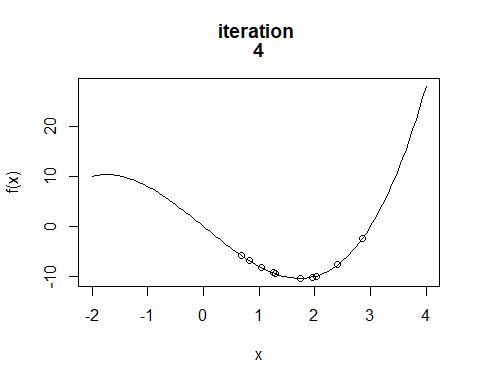
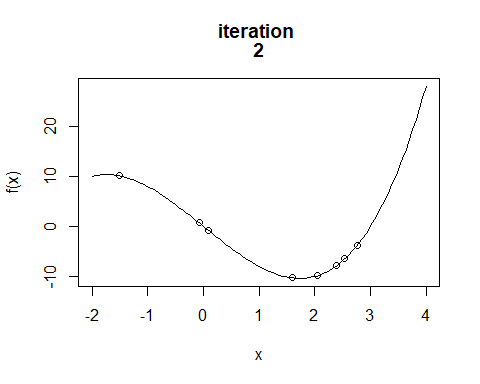
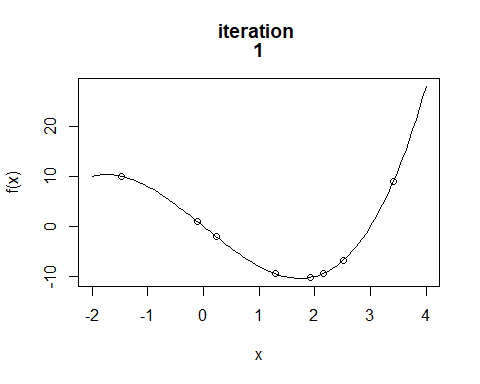


## best x value: -8.493187e-17   
## minimum: 7.213423e-33

#more complex function  
f <- function(x){  
 return(x^3 - 9\*x)  
}  
curve(f,-2,4)

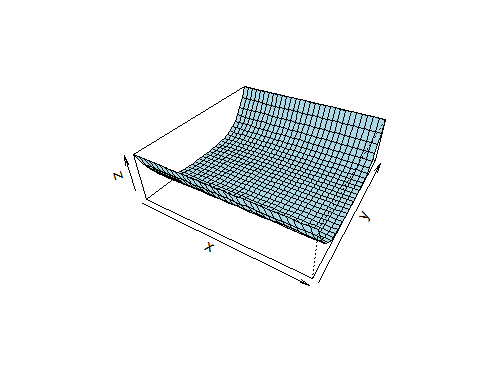


swarm(10,-2,4,.5,.5,.5)



## best x value: 1.732051   
## minimum: -10.3923

set.seed(1111)  
  
  
swarm\_nd <- function(S,n,lower, upper,omega,phiP,phiG){  
 iterations <- 100  
 best\_swarm <- lower  
 #particle positions  
 x <- matrix(0,nrow = n,ncol = S)   
 #particle best positions  
 p <- matrix(0,nrow = n,ncol = S)   
 #particle velocities  
 v <- matrix(0,nrow = n,ncol = S)   
   
 #for each particle i   
 for (i in 1:S){  
 #initial position  
 for (k in 1:n){  
 x[k,i] <- runif(1,lower[k],upper[k])  
 }  
 p[,i] <- x[,i]  
   
 if(f(p[,i]) < f(best\_swarm)){  
 best\_swarm <- p[,i]  
 }  
   
 #initial velocity  
 for (k in 1:n){  
 v[k,i] <- runif(1,-abs(upper[k]-lower[k]),abs(upper[k]-lower[k]))  
 }  
 }  
 #print(x)  
 #print(p)  
 #print(v)  
 #print(best\_swarm)  
 for(j in 1:iterations){  
   
 for (i in 1:S){  
   
 for (d in 1:n){  
 rp <- runif(1,0,1)  
 rg <- runif(1,0,1)  
   
 v[,i] <- omega\*v[,i] + phiP\*rp\*(p[,i]-x[,i]) + phiG\*rg\*(best\_swarm-x[,i])  
   
 }  
   
   
 x[,i] <- x[,i] + v[,i]  
   
 if (f(x[,i]) < f(p[,i])){  
 p[,i] <- x[,i]  
   
 #update best position in the swarm  
 if (f(p[,i]) < f(best\_swarm)){  
 best\_swarm <- p[,i]  
 }  
 }  
 }  
  
 }  
 cat("best x,y value: ",best\_swarm,"\n")  
 cat(2\*best\_swarm[1]^2 + 4\*best\_swarm[2]^2,"\n")  
 cat("minimum: ",f(best\_swarm),"\n")  
 cat("\n")  
 return(best\_swarm)  
}  
  
f <- function(x,y){  
 return((4 - x^2 - 2\*y^2)^2)  
}  
x <- seq(-4,4,length = 30)  
y <- seq(-100,100,length = 30)  
z <- outer(x,y,f)  
op <- par(bg= 'white')  
persp(x, y, z, theta = 30, phi = 35, expand = 0.3, col = "lightblue")



f <- function(x){  
 return((4 - x[1]^2 - 2\*x[2]^2)^2)  
}  
best\_swarm <- swarm\_nd(100,2,c(-4,-4),c(4,100),.01,.01,.01)

## best x,y value: -1.959869 -0.2895273   
## 8.017478   
## minimum: 7.636615e-05