

```

neldermead <- function(f, x0, alpha, beta, gamma, tol) {
  n <- length(x0)
  step <- 0.1 * max(c(abs(x0), 1))
  x <- matrix(x0, nrow = n + 1, ncol = n, byrow = T) + step * rbind(0, diag(n))
  f.x <- apply(x, 1, f)
  while(diff(range(f.x)) > tol) {
    l <- which.min(f.x)
    f.l <- min(f.x)
    h <- which.max(f.x)
    f.h <- max(f.x)
    xb <- colMeans(x[-h,,drop = F])
    xr <- (1 + alpha) * xb - alpha * x[h,]
    f.xr <- f(xr)
    if (f.xr < f.l) {
      xe <- (1 - gamma) * xb + gamma * xr
      f.xe <- f(xe)
      if (f.xe < f.xr) {
        x[h,] <- xe
        f.x[h] <- f.xe
      } else {
        x[h,] <- xr
        f.x[h] <- f.xr
      }
    } else {
      shrink <- T
      if (f.xr < f.x[h]) {
        x[h,] <- xr
        f.x[h] <- f.xr
        shrink <- F
      } else {
        xc <- beta * xb + (1 - beta) * x[h,]

```

```
f.xc <- f(xc)
if (f.xc < f.x[h]) {
  x[h,] <- xc
  f.x[h] <- f.xc
}
else if ((f.xr >= f.x[h]) && shrink) {
  x <- matrix(x[l,], nrow = n + 1, ncol = n, byrow = T) +
  beta * sweep(x, 2, x[l,])
  f.x <- apply(x, 1, f)
}
}
}
}
return(x[which.min(f.x),])
}
```