

```

-----
#####
function broyden(x:Array)
    f1 = (3.0-2.0*x[1])*x[1] - 2.0*x[2] + 1.0
    f2 = (3.0-2.0*x[2])*x[2] - x[1] + 1.0

    return f1^2 + f2^2

end
#####
-----
        else
            Delta = Delta/2.0
        end
        @printf(" %2d  %9.6f  %8.6f  %6d\n",k,f,Delta,nf)
end
-----
objfun= broyden
k      = 0
x      = transpose(x0)
f      = objfun(x)
Delta = Delta0
nf     = 1

@printf(" iter   obj.      Delta   f.ivals\n")
@printf(" %2d   %9.6f  %8.6f  %6d\n",k,f,Delta,nf)

while Delta >= Deltamin && k <= maxiter
-----
        f1 = objfun(y + Delta*D[:,i])
        nf = nf + 1
        if f1 < fy
            y = y + Delta*D[:,i]
            fy= f1
        end
    end
-----
        if fz < fy
            x = copy(z)
            f = fz
        else
            x = copy(y)
            f = fy
        end
-----
        k = k+1
        y = copy(x)
        fy= f
        for i = 1:q
-----
            if f1 < fz
                fz = f1
                z = z + Delta*D[:,i]
            end
        end
-----
        if fy < f
            z = copy(2*y - x)
            fz= objfun(z)
            nf= nf + 1
-----
Delta0 = 0.3
Deltamin = 1.e-5
maxiter = 10000

x0 = [0.0 -1.0]
n = length(x0)
n = 2
q = 2*n
D = zeros(n,q)
for i = 1:n
    D[i, i] = 1.0
    D[i,n+i] = -1.0
end
-----
        for i = 1:q
            f1 = objfun(z + Delta*D[:,i])
            nf = nf + 1
-----

```

1

2

3

4

5

6

7

8

9

10