OML - Lab Assignment 5

> Ayush Abrol B20AI052

```
#Let's Import
from numpy import *
```

Question 1

```
def fun(x):
    return 0.5*(100*pow(x[0]-pow(x[1],2),2)+pow(2-x[0],2))

def gradf(x):
    f0,n,h1=fun(x),len(x),pow(10,-5)
    g=zeros((n,1),dtype=float)
    for i in range(0,n):
        x1=x.copy()
        x1[i]=x1[i]+h1
        g[i]=(fun(x1)-f0)/h1
    return g
x0, beta1, beta2, r, eps, iter1, B0, countf, countg = array([[4.0],[-4.0]]),pow(10,-4),0.9,0.5,pow(10,-3),0
                                            identity(2,dtype=float),0,0
f0, g0 = fun(x0), gradf(x0)
countf += 1
countg += 1
while linalg.norm(g0)>eps and iter1<1000:
    d0,alpha= dot(linalg.inv(B0),g0),1
    d0,alpha=-dot(linalg.inv(B0),g0),1
    x1=x0+alpha*d0
    f1,g1=fun(x1),gradf(x1)
    countf += 1
    countg += 1
    while (f1>f0+alpha*beta1*g0.T@d0 or\
            dot(g1.T,d0)<beta2*dot(g0.T,d0)) and alpha>pow(10,-5):
        alpha=alpha*r
        x1 = x0 + alpha * d0
        f1, g1 = fun(x1), gradf(x1)
        countf += 1
        countg += 1
    dt1,s1=x1-x0,g1-g0
    x0,g0,iter1=x1,g1,iter1+1

print("x0 : ",x0,"iter : ",iter1,"countf : ",countf,"countg : ",countg)
```

```
x0 :  [[30.22140575]
 [-5.64739222]] iter :  1000 countf :  10243 countg :  10243
```

Question 2

```
def fun(x):
    return 0.5*(100*pow(x[0]-pow(x[1],2),2)+pow(2-x[0],2))

def gradf(x):
    f0,n,h1=fun(x),len(x),pow(10,-5)
    g=zeros((n,1),dtype=float)
    for i in range(0,n):
        x1=x.copy()
        x1[i]=x1[i]+h1
        g[i]=(fun(x1)-f0)/h1
    return g

x0,beta1,beta2,r,eps,iter1,B0,countf,countg=array([[4.0],[-4.0]]),pow(10,-4),0.9,0.5,pow(10,-3),0,\
                                            [[4,pow(2,1/2)],[pow(2,1/2),2]],0,0
f0,g0=fun(x0),gradf(x0)
countf+=1
countg+=1
while linalg.norm(g0)>eps and iter1<1000:
    d0,alpha=-dot(linalg.inv(B0),g0),1

    x1=x0+alpha*d0
    f1,g1=fun(x1),gradf(x1)
    countf += 1
    countg += 1
    while (f1>f0+alpha*beta1*g0.T@d0 or\
            dot(g1.T,d0)<beta2*dot(g0.T,d0)) and alpha>pow(10,-5):
        alpha=alpha*r
        x1 = x0 + alpha * d0
        f1, g1 = fun(x1), gradf(x1)
        countf += 1
        countg += 1
    dt1,s1=x1-x0,g1-g0
    x0,g0,iter1=x1,g1,iter1+1

print("x0 : ",x0,"iter : ",iter1,"countf : ",countf,"countg : ",countg)
```

```
x0 :  [[1.16791338]
 [1.58514763]] iter :  1000 countf :  9326 countg :  9326
```

Question 3

```
def fun(x):
    return 0.5*(100*pow(x[0]-pow(x[1],2),2)+pow(2-x[0],2))

def gradf(x):
    f0,n,h1=fun(x),len(x),pow(10,-5)
    g=zeros((n,1),dtype=float)
    for i in range(0,n):
        x1=x.copy()
        x1[i]=x1[i]+h1
       # print(x,x1)
        g[i]=(fun(x1)-f0)/h1
    return g

x0,beta1,beta2,r,eps,iter1,B0,countf,countg=array([[4.0],[-4.0]]),pow(10,-4),0.9,0.5,pow(10,-3),0,\
                                            identity(2,dtype=float),0,0
f0,g0=fun(x0),gradf(x0)
countf+=1
countg+=1
while linalg.norm(g0)>eps and iter1<1000:
    d0,alpha=-dot(linalg.inv(B0),g0),1

    x1=x0+alpha*d0
    f1,g1=fun(x1),gradf(x1)
    countf += 1
    countg += 1
    while (f1>f0+alpha*beta1*g0.T@d0 or\
            dot(g1.T,d0)<beta2*dot(g0.T,d0)) and alpha>pow(10,-5):
        alpha=alpha*r
        x1 = x0 + alpha * d0
        f1, g1 = fun(x1), gradf(x1)
        countf += 1
        countg += 1
    dt1,s1=x1-x0,g1-g0
    B0=B0+1/(dt1.T@s1)*s1@s1.T-1/(s1.T@B0@s1)*B0@s1@s1.T@B0
    x0,g0,iter1=x1,g1,iter1+1

print("x0 : ",x0,"iter : ",iter1,"countf : ",countf,"countg : ",countg)
```

```
x0 :  [[0.12558541]
 [0.33292398]] iter :  1000 countf :  17810 countg :  17810
```