MIE 1621 Computational project Junyu Jiang 1002916561

Part 1

(a), (b) & (c)

The algorithm of using gradient-based method with backtracking to solve multivariate function is shown above.

```
import numpy as np
from numpy import linalg as LA

f = lambda x: (x[0]**2 + x[1]**2 + x[2]**2)
dfx1 = lambda x: (2*x[0])
dfx2 = lambda x: (2*x[1])
dfx3 = lambda x: (2*x[2])

x0 = np.array([1,1,1])

#Parameters
alpha = 1
beta = 0.5
count = 1
gamma = 0.5


while LA.norm(np.array([dfx1(x0),dfx2(x0),dfx3(x0)]))/(1+LA.norm(np.array([f(x0)]))) >= 10**(-5):
    print("""

######################
###   iteration {}   ###
######################
""".format(count))
    print("Initial point: ", x0)
    d = - np.array([dfx1(x0),dfx2(x0),dfx3(x0)])

    while (f(x0) - f(x0 + alpha * d)) < (- gamma * alpha * np.dot(np.transpose(np.array([dfx1(x0),dfx2(x0),dfx3(x0)])),d)):
       alpha = beta * alpha
    else:
      alpha = alpha
    x1 = x0 + alpha * d
    x0 = x1

    print( "Search Direction:", d,"step length: ", alpha, "New Iterate:" , x0)
    count +=1
    if count +1 > 1001:
        break
    else:
        continue
```

(d)

1) Function and variable set-up for $f_1(x)= x_1^2+x_2^2+x_3^2$:

```
f = lambda x: (x[0]**2 + x[1]**2 + x[2]**2)
dfx1 = lambda x: (2*x[0])
dfx2 = lambda x: (2*x[1])
dfx3 = lambda x: (2*x[2])

x0 = np.array([1,1,1])

#Parameters
alpha = 1
beta = 0.5
count = 1
gamma = 0.5
```

```
######################
###   iteration 1   ###
######################

Initial point:  [1 1 1]
Search Direction: [-2 -2 -2] step length:  0.5 New Iterate: [0. 0. 0.]
```

It converges at iteration 1

2) Function and variable set-up for $f_2(x) = x_1^2+2x_2^2-2x_1x_2-2x_2$

```
f = lambda x: (x[0]**2 + 2*x[1]**2 - 2*x[0]*x[1] -2*x[1])
dfx1 = lambda x: (2*x[0] - 2*x[1])
dfx2 = lambda x: (4*x[1] - 2*x[0] -2)


x0 = np.array([0,0])

#Parameters
alpha = 1
beta = 0.5
count = 1
gamma = 0.5
```

```
######################
###   iteration 1   ###
######################

Initial point:  [0 0]
Search Direction: [0 2] step length:  0.25 New Iterate: [0.  0.5]


######################
###   iteration 2   ###
######################

Initial point:  [0.  0.5]
Search Direction: [ 1. -0.] step length:  0.25 New Iterate: [0.25 0.5 ]


######################
###   iteration 3   ###
######################

Initial point:  [0.25 0.5 ]
Search Direction: [0.5 0.5] step length:  0.25 New Iterate: [0.375 0.625]


######################
###   iteration 4   ###
######################

Initial point:  [0.375 0.625]
Search Direction: [0.5  0.25] step length:  0.25 New Iterate: [0.5    0.6875]


######################
###   iteration 5   ###
######################

Initial point:  [0.5    0.6875]
Search Direction: [0.375 0.25 ] step length:  0.25 New Iterate: [0.59375 0.75   ]


######################
###   iteration 6   ###
######################

Initial point:  [0.59375 0.75   ]
Search Direction: [0.3125 0.1875] step length:  0.25 New Iterate: [0.671875 0.796875]


######################
###   iteration 7   ###
######################

Initial point:  [0.671875 0.796875]
Search Direction: [0.25    0.15625] step length:  0.25 New Iterate: [0.734375  0.8359375]
```

```
######################
###   iteration 8   ###
######################

Initial point:  [0.734375  0.8359375]
Search Direction: [0.203125 0.125   ] step length:  0.25 New Iterate: [0.78515625 0.8671875 ]


######################
###   iteration 9   ###
######################

Initial point:  [0.78515625 0.8671875 ]
Search Direction: [0.1640625 0.1015625] step length:  0.25 New Iterate: [0.82617188 0.89257812]


######################
###   iteration 10   ###
######################

Initial point:  [0.82617188 0.89257812]
Search Direction: [0.1328125  0.08203125] step length:  0.25 New Iterate: [0.859375   0.91308594]
```

```
######################
###   iteration 48   ###
######################

Initial point:  [0.99994473 0.99996584]
Search Direction: [4.22235057e-05 2.60955617e-05] step length:  0.25 New Iterate: [0.99995528 0.99997236]


######################
###   iteration 49   ###
######################

Initial point:  [0.99995528 0.99997236]
Search Direction: [3.41595337e-05 2.11117529e-05] step length:  0.25 New Iterate: [0.99996382 0.99997764]


######################
###   iteration 50   ###
######################

Initial point:  [0.99996382 0.99997764]
Search Direction: [2.76356433e-05 1.70797668e-05] step length:  0.25 New Iterate: [0.99997073 0.99998191]


######################
###   iteration 51   ###
######################

Initial point:  [0.99997073 0.99998191]
Search Direction: [2.23577051e-05 1.38178216e-05] step length:  0.25 New Iterate: [0.99997632 0.99998537]


######################
###   iteration 52   ###
######################

Initial point:  [0.99997632 0.99998537]
Search Direction: [1.80877634e-05 1.11788525e-05] step length:  0.25 New Iterate: [0.99998084 0.99998816]
```

It converges at iteration 52

3) Function and variable set-up for $f_3(x) = 100(x_2 - x_1^2)^2 + (1-x_1)^2$

```
f = lambda x: (100*(x[1]-x[0]**2)**2 + (1-x[0])**2)
dfx1 = lambda x: (400*x[0]*x[1] - 400*x[0]**3 + 2*x[0]-2)
dfx2 = lambda x: (200*x[1] - 200*x[0]**2)


x0 = np.array([-1.2,1])
```

```
######################
###   iteration 1   ###
######################

Initial point:  [-1.2  1. ]
Search Direction: [-206.8   88. ] step length:  5e-324 New Iterate: [-1.2  1. ]


######################
###   iteration 2   ###
######################

Initial point:  [-1.2  1. ]
Search Direction: [-206.8   88. ] step length:  5e-324 New Iterate: [-1.2  1. ]


######################
###   iteration 3   ###
######################

Initial point:  [-1.2  1. ]
Search Direction: [-206.8   88. ] step length:  5e-324 New Iterate: [-1.2  1. ]


######################
###   iteration 4   ###
######################

Initial point:  [-1.2  1. ]
Search Direction: [-206.8   88. ] step length:  5e-324 New Iterate: [-1.2  1. ]


######################
###   iteration 5   ###
######################

Initial point:  [-1.2  1. ]
Search Direction: [-206.8   88. ] step length:  5e-324 New Iterate: [-1.2  1. ]


######################
###   iteration 6   ###
######################

Initial point:  [-1.2  1. ]
Search Direction: [-206.8   88. ] step length:  5e-324 New Iterate: [-1.2  1. ]


######################
###   iteration 7   ###
######################

Initial point:  [-1.2  1. ]
Search Direction: [-206.8   88. ] step length:  5e-324 New Iterate: [-1.2  1. ]
```

```
######################
###   iteration 8   ###
######################

Initial point:  [-1.2  1. ]
Search Direction: [-206.8   88. ] step length:  5e-324 New Iterate: [-1.2  1. ]


######################
###   iteration 9   ###
######################

Initial point:  [-1.2  1. ]
Search Direction: [-206.8   88. ] step length:  5e-324 New Iterate: [-1.2  1. ]


######################
###   iteration 10   ###
######################

Initial point:  [-1.2  1. ]
Search Direction: [-206.8   88. ] step length:  5e-324 New Iterate: [-1.2  1. ]
```

```
######################
###   iteration 996   ###
######################

Initial point:  [-1.2  1. ]
Search Direction: [-206.8   88. ] step length:  5e-324 New Iterate: [-1.2  1. ]


######################
###   iteration 997   ###
######################

Initial point:  [-1.2  1. ]
Search Direction: [-206.8   88. ] step length:  5e-324 New Iterate: [-1.2  1. ]


######################
###   iteration 998   ###
######################

Initial point:  [-1.2  1. ]
Search Direction: [-206.8   88. ] step length:  5e-324 New Iterate: [-1.2  1. ]


######################
###   iteration 999   ###
######################

Initial point:  [-1.2  1. ]
Search Direction: [-206.8   88. ] step length:  5e-324 New Iterate: [-1.2  1. ]


######################
###   iteration 1000   ###
######################

Initial point:  [-1.2  1. ]
Search Direction: [-206.8   88. ] step length:  5e-324 New Iterate: [-1.2  1. ]
```

4) Function and variable set-up for $f_4(x) = (x_1+x_2)^4 + x_2^2$

```python
f = lambda x: ((x[0]+x[1])**4 + x[1]**2)
dfx1 = lambda x: (4*(x[0]+x[1])**3)
dfx2 = lambda x: (4*(x[0]+x[1])**3 + 2*x[1])


x0 = np.array([2,-2])

#Parameters
alpha =1
beta = 0.5
count = 1
gamma = 0.5
```

```
######################
###   iteration 1   ###
######################

Initial point:  [ 2 -2]
Search Direction: [0 4] step length:  0.25 New Iterate: [ 2. -1.]


######################
###   iteration 2   ###
######################

Initial point:  [ 2. -1.]
Search Direction: [-4. -2.] step length:  0.03125 New Iterate: [ 1.875  -1.0625]


######################
###   iteration 3   ###
######################

Initial point:  [ 1.875  -1.0625]
Search Direction: [-2.14550781 -0.02050781] step length:  0.03125 New Iterate: [ 1.80795288 -1.06314087]


######################
###   iteration 4   ###
######################

Initial point:  [ 1.80795288 -1.06314087]
Search Direction: [-1.65272276  0.47355898] step length:  0.03125 New Iterate: [ 1.75630529 -1.04834215]


######################
###   iteration 5   ###
######################

Initial point:  [ 1.75630529 -1.04834215]
Search Direction: [-1.41935796  0.67732634] step length:  0.03125 New Iterate: [ 1.71195036 -1.0271757 ]


######################
###   iteration 6   ###
######################

Initial point:  [ 1.71195036 -1.0271757 ]
Search Direction: [-1.28440807  0.76994334] step length:  0.03125 New Iterate: [ 1.67181261 -1.00311497]


######################
###   iteration 7   ###
######################

Initial point:  [ 1.67181261 -1.00311497]
Search Direction: [-1.19605004  0.81017991] step length:  0.03125 New Iterate: [ 1.63443604 -0.97779685]
```

```
######################
###   iteration 8   ###
######################

Initial point:  [ 1.63443604 -0.97779685]
Search Direction: [-1.13250568  0.82308802] step length:  0.03125 New Iterate: [ 1.59904524 -0.95207535]


######################
###   iteration 9   ###
######################

Initial point:  [ 1.59904524 -0.95207535]
Search Direction: [-1.08320884  0.82094186] step length:  0.03125 New Iterate: [ 1.56519496 -0.92642092]


######################
###   iteration 10   ###
######################

Initial point:  [ 1.56519496 -0.92642092]
Search Direction: [-1.04256173  0.81028011] step length:  0.03125 New Iterate: [ 1.53261491 -0.90109966]
```

```
######################
###   iteration 996   ###
######################

Initial point:  [ 0.06952085 -0.00067161]
Search Direction: [-1.30544145e-03  3.77816462e-05] step length:  0.03125 New Iterate: [ 0.06948005 -0.00067043]


######################
###   iteration 997   ###
######################

Initial point:  [ 0.06948005 -0.00067043]
Search Direction: [-1.30318938e-03  3.76723657e-05] step length:  0.03125 New Iterate: [ 0.06943933 -0.00066925]


######################
###   iteration 998   ###
######################

Initial point:  [ 0.06943933 -0.00066925]
Search Direction: [-1.30094370e-03  3.75635235e-05] step length:  0.03125 New Iterate: [ 0.06939868 -0.00066808]


######################
###   iteration 999   ###
######################

Initial point:  [ 0.06939868 -0.00066808]
Search Direction: [-1.29870439e-03  3.74551172e-05] step length:  0.03125 New Iterate: [ 0.06935809 -0.00066691]


######################
###   iteration 1000   ###
######################

Initial point:  [ 0.06935809 -0.00066691]
Search Direction: [-1.29647141e-03  3.73471446e-05] step length:  0.03125 New Iterate: [ 0.06931758 -0.00066574]
```

Iteration maximum is reached.

5) Function and variable set-up for $f_5(x) = (x_1-1)^2 + (x_2-1)^2 + c(x_1^2 + x_2^2-0.25)^2$

When c=1 :

```
c = 1
f = lambda x: ((x[0]-1)**2 + (x[1]-1)**2 + c*(x[0]**2+x[1]**2-0.25)**2)
dfx1 = lambda x: (2*(x[0]-1) + 4*c*x[0]*(x[0]**2+x[1]**2-0.25))
dfx2 = lambda x: (2*(x[1]-1) + 4*c*x[1]*(x[0]**2+x[1]**2-0.25))


x0 = np.array([1,-1])

#Parameters
alpha = 1
beta = 0.5
count = 1
gamma = 0.1
```

```
######################
###   iteration 1   ###
######################

Initial point:  [ 1 -1]
Search Direction: [-7. 11.] step length:  0.125 New Iterate: [0.125 0.375]


######################
###   iteration 2   ###
######################

Initial point:  [0.125 0.375]
Search Direction: [1.796875 1.390625] step length:  0.125 New Iterate: [0.34960938 0.54882812]


######################
###   iteration 3   ###
######################

Initial point:  [0.34960938 0.54882812]
Search Direction: [1.05823761 0.52159089] step length:  0.125 New Iterate: [0.48188908 0.61402699]


######################
###   iteration 4   ###
######################

Initial point:  [0.48188908 0.61402699]
Search Direction: [ 0.34375453 -0.11040147] step length:  0.125 New Iterate: [0.52485839 0.6002268 ]


######################
###   iteration 5   ###
######################

Initial point:  [0.52485839 0.6002268 ]
Search Direction: [ 0.14042977 -0.12660007] step length:  0.125 New Iterate: [0.54241211 0.58440179]


######################
###   iteration 6   ###
######################

Initial point:  [0.54241211 0.58440179]
Search Direction: [ 0.07826348 -0.07050366] step length:  0.125 New Iterate: [0.55219505 0.57558884]


######################
###   iteration 7   ###
######################

Initial point:  [0.55219505 0.57558884]
Search Direction: [ 0.04253066 -0.04039768] step length:  0.125 New Iterate: [0.55751138 0.57053913]
```

```
######################
###   iteration 8   ###
######################

Initial point:  [0.55751138 0.57053913]
Search Direction: [ 0.0234352  -0.02275252] step length:  0.125 New Iterate: [0.56044078 0.56769506]


######################
###   iteration 9   ###
######################

Initial point:  [0.56044078 0.56769506]
Search Direction: [ 0.01296494 -0.012755  ] step length:  0.125 New Iterate: [0.5620614  0.56610068]


######################
###   iteration 10   ###
######################

Initial point:  [0.5620614  0.56610068]
Search Direction: [ 0.00719332 -0.0071281 ] step length:  0.125 New Iterate: [0.56296056 0.56520967]
```

```
######################
###   iteration 17   ###
######################

Initial point:  [0.56405344 0.56412046]
Search Direction: [ 0.00011883 -0.00011881] step length:  0.125 New Iterate: [0.56406829 0.56410561]


######################
###   iteration 18   ###
######################

Initial point:  [0.56406829 0.56410561]
Search Direction: [ 6.61622730e-05 -6.61567078e-05] step length:  0.125 New Iterate: [0.56407656 0.56409734]


######################
###   iteration 19   ###
######################

Initial point:  [0.56407656 0.56409734]
Search Direction: [ 3.68388582e-05 -3.68371328e-05] step length:  0.125 New Iterate: [0.56408116 0.56409273]


######################
###   iteration 20   ###
######################

Initial point:  [0.56408116 0.56409273]
Search Direction: [ 2.05118812e-05 -2.05113463e-05] step length:  0.125 New Iterate: [0.56408373 0.56409017]


######################
###   iteration 21   ###
######################

Initial point:  [0.56408373 0.56409017]
Search Direction: [ 1.14210708e-05 -1.14209049e-05] step length:  0.125 New Iterate: [0.56408516 0.56408874]
```

It converges at iteration 21.

When c = 10:

```
c = 10
f = lambda x: ((x[0]-1)**2 + (x[1]-1)**2 + c*(x[0]**2+x[1]**2-0.25)**2)
dfx1 = lambda x: (2*(x[0]-1) + 4*c*x[0]*(x[0]**2+x[1]**2-0.25))
dfx2 = lambda x: (2*(x[1]-1) + 4*c*x[1]*(x[0]**2+x[1]**2-0.25))


x0 = np.array([1,-1])

#Parameters
alpha = 1
beta = 0.5
count = 1
gamma = 0.1
```

```
######################
###   iteration 1   ###
######################

Initial point:  [ 1 -1]
Search Direction: [-70.  74.] step length:  0.015625 New Iterate: [-0.09375  0.15625]


######################
###   iteration 2   ###
######################

Initial point:  [-0.09375  0.15625]
Search Direction: [1.37451172 3.04248047] step length:  0.015625 New Iterate: [-0.07227325  0.20378876]


######################
###   iteration 3   ###
######################

Initial point:  [-0.07227325  0.20378876]
Search Direction: [1.55697444 3.24919834] step length:  0.015625 New Iterate: [-0.04794553  0.25455748]


######################
###   iteration 4   ###
######################

Initial point:  [-0.04794553  0.25455748]
Search Direction: [1.74511828 3.35324503] step length:  0.015625 New Iterate: [-0.02067806  0.30695193]


######################
###   iteration 5   ###
######################

Initial point:  [-0.02067806  0.30695193]
Search Direction: [1.91286025 3.2935314 ] step length:  0.015625 New Iterate: [0.00921039 0.35841336]


######################
###   iteration 6   ###
######################

Initial point:  [0.00921039 0.35841336]
Search Direction: [2.02632514 3.0244175 ] step length:  0.015625 New Iterate: [0.04087172 0.40566989]


######################
###   iteration 7   ###
######################

Initial point:  [0.04087172 0.40566989]
Search Direction: [2.05519553 2.54784008] step length:  0.015625 New Iterate: [0.07298415 0.44547989]
```

```
########################
###   iteration 8   ###
########################

Initial point:  [0.07298415 0.44547989]
Search Direction: [1.98896767 1.93266097] step length:  0.015625 New Iterate: [0.10406177 0.47567772]


########################
###   iteration 9   ###
########################

Initial point:  [0.10406177 0.47567772]
Search Direction: [1.84558008 1.29412966] step length:  0.015625 New Iterate: [0.13289895 0.49589849]


########################
###   iteration 10   ###
########################

Initial point:  [0.13289895 0.49589849]
Search Direction: [1.66202496 0.73888162] step length:  0.015625 New Iterate: [0.15886809 0.50744352]
```

```
######################
###   iteration 146   ###
######################

Initial point:  [0.40260645 0.40261359]
Search Direction: [ 1.77511231e-05 -1.77507552e-05] step length:  0.015625 New Iterate: [0.40260672 0.40261332]


######################
###   iteration 147   ###
######################

Initial point:  [0.40260672 0.40261332]
Search Direction: [ 1.63732937e-05 -1.63729808e-05] step length:  0.015625 New Iterate: [0.40260698 0.40261306]


######################
###   iteration 148   ###
######################

Initial point:  [0.40260698 0.40261306]
Search Direction: [ 1.51024115e-05 -1.51021452e-05] step length:  0.015625 New Iterate: [0.40260722 0.40261282]


######################
###   iteration 149   ###
######################

Initial point:  [0.40260722 0.40261282]
Search Direction: [ 1.39301749e-05 -1.39299483e-05] step length:  0.015625 New Iterate: [0.40260743 0.40261261]


######################
###   iteration 150   ###
######################

Initial point:  [0.40260743 0.40261261]
Search Direction: [ 1.28489270e-05 -1.28487343e-05] step length:  0.015625 New Iterate: [0.40260763 0.40261241]
```

It converges at iteration 150.

When c = 100:

```
c = 100
f = lambda x: ((x[0]-1)**2 + (x[1]-1)**2 + c*(x[0]**2+x[1]**2-0.25)**2)
dfx1 = lambda x: (2*(x[0]-1) + 4*c*x[0]*(x[0]**2+x[1]**2-0.25))
dfx2 = lambda x: (2*(x[1]-1) + 4*c*x[1]*(x[0]**2+x[1]**2-0.25))


x0 = np.array([1,-1])

#Parameters
alpha = 1
beta = 0.5
count = 1
gamma = 0.1
```

```
######################
###   iteration 1   ###
######################

Initial point:  [ 1 -1]
Search Direction: [-700.  704.] step length:  0.001953125 New Iterate: [-0.3671875  0.375    ]


######################
###   iteration 2   ###
######################

Initial point:  [-0.3671875  0.375    ]
Search Direction: [ 6.47258759 -2.56774902] step length:  0.001953125 New Iterate: [-0.35454573  0.36998487


######################
###   iteration 3   ###
######################

Initial point:  [-0.35454573  0.36998487]
Search Direction: [ 4.49479267 -0.60343155] step length:  0.001953125 New Iterate: [-0.34576684  0.36880629


######################
###   iteration 4   ###
######################

Initial point:  [-0.34576684  0.36880629]
Search Direction: [3.46228702 0.44027653] step length:  0.001953125 New Iterate: [-0.33900456  0.3696662 ]


######################
###   iteration 5   ###
######################

Initial point:  [-0.33900456  0.3696662 ]
Search Direction: [2.89187905 1.02745395] step length:  0.001953125 New Iterate: [-0.33335635  0.37167295]


######################
###   iteration 6   ###
######################

Initial point:  [-0.33335635  0.37167295]
Search Direction: [2.56900468 1.36559288] step length:  0.001953125 New Iterate: [-0.32833877  0.37434012]


######################
###   iteration 7   ###
######################

Initial point:  [-0.32833877  0.37434012]
Search Direction: [2.3857158  1.56024414] step length:  0.001953125 New Iterate: [-0.32367917  0.37738747]

######################
###   iteration 8   ###
######################

Initial point:  [-0.32367917  0.37738747]
Search Direction: [2.28347933 1.66948275] step length:  0.001953125 New Iterate: [-0.31921925  0.38064818]


######################
###   iteration 9   ###
######################

Initial point:  [-0.31921925  0.38064818]
Search Direction: [2.2290674  1.72685203] step length:  0.001953125 New Iterate: [-0.3148656   0.38402094]


######################
###   iteration 10   ###
######################

Initial point:  [-0.3148656   0.38402094]
Search Direction: [2.20307926 1.75231753] step length:  0.001953125 New Iterate: [-0.31056271  0.38744344]
```

```
######################
###   iteration 996   ###
######################

Initial point:  [0.35977592 0.35980314]
Search Direction: [ 7.56628490e-05 -7.56570004e-05] step length:  0.001953125 New Iterate: [0.35977607 0.35980299]


######################
###   iteration 997   ###
######################

Initial point:  [0.35977607 0.35980299]
Search Direction: [ 7.48413429e-05 -7.48356206e-05] step length:  0.001953125 New Iterate: [0.35977621 0.35980285]


######################
###   iteration 998   ###
######################

Initial point:  [0.35977621 0.35980285]
Search Direction: [ 7.40287566e-05 -7.40231578e-05] step length:  0.001953125 New Iterate: [0.35977636 0.3598027 ]


######################
###   iteration 999   ###
######################

Initial point:  [0.35977636 0.3598027 ]
Search Direction: [ 7.32249932e-05 -7.32195154e-05] step length:  0.001953125 New Iterate: [0.3597765  0.35980256]


######################
###   iteration 1000   ###
######################

Initial point:  [0.3597765  0.35980256]
Search Direction: [ 7.24299570e-05 -7.24245975e-05] step length:  0.001953125 New Iterate: [0.35977664 0.35980242]
```

Iteration maximum is reached.

Larger c will make the problem be slower due to more iterations will be made.

e) All the problems above can converge to the optimal solution except for f_3. The problem for f_3 is that alpha becomes too small after backtracking. Therefore, it can only change the value of x a little after one iteration.

Part 2:

a) Newton's method to minimize $100x_1^4 + 0.01x_2^4$

```
f = lambda x: (100*x[0]**4 + 0.01*x[1]**4)
dfx1 = lambda x: (400*x[0]**3)
dfx2 = lambda x: (0.04*x[1]**3)
hx11 = lambda x: (1200*x[0]**2)
hx12 = lambda x: (0)
hx21 = lambda x: (0)
hx22 = lambda x: (0.12*x[1]**2)

#parameters
alpha = 1
count = 1

#initial point
x0 = np.array([1,1])


while LA.norm(np.array([[dfx1(x0),dfx2(x0)]])) > 10**(-6):

    print("""

#######################
###   iteration {}   ###
#######################
""".format(count))
    print("Initial point: ", x0)
    d = -np.dot(np.array([dfx1(x0),dfx2(x0)]), inv(np.array([[hx11(x0),hx12(x0)],
                        [hx21(x0),hx22(x0)]])))

    x0 = x0 + alpha * d
    count += 1
    print("Search Direction:", d,"New Iterate:" , x0)
```

```
#####################
###   iteration 1   ###
#####################

Initial point:  [1 1]
Search Direction: [-0.33333333 -0.33333333] New Iterate: [0.66666667 0.66666667]


#####################
###   iteration 2   ###
#####################

Initial point:  [0.66666667 0.66666667]
Search Direction: [-0.22222222 -0.22222222] New Iterate: [0.44444444 0.44444444]


#####################
###   iteration 3   ###
#####################

Initial point:  [0.44444444 0.44444444]
Search Direction: [-0.14814815 -0.14814815] New Iterate: [0.2962963 0.2962963]


#####################
###   iteration 4   ###
#####################

Initial point:  [0.2962963 0.2962963]
Search Direction: [-0.09876543 -0.09876543] New Iterate: [0.19753086 0.19753086]


#####################
###   iteration 5   ###
#####################

Initial point:  [0.19753086 0.19753086]
Search Direction: [-0.06584362 -0.06584362] New Iterate: [0.13168724 0.13168724]


#####################
###   iteration 6   ###
#####################

Initial point:  [0.13168724 0.13168724]
Search Direction: [-0.04389575 -0.04389575] New Iterate: [0.0877915 0.0877915]


#####################
###   iteration 7   ###
#####################

Initial point:  [0.0877915 0.0877915]
Search Direction: [-0.02926383 -0.02926383] New Iterate: [0.05852766 0.05852766]
```

```
######################
###   iteration 8   ###
######################

Initial point:  [0.05852766 0.05852766]
Search Direction: [-0.01950922 -0.01950922] New Iterate: [0.03901844 0.03901844]


######################
###   iteration 9   ###
######################

Initial point:  [0.03901844 0.03901844]
Search Direction: [-0.01300615 -0.01300615] New Iterate: [0.02601229 0.02601229]


######################
###   iteration 10   ###
######################

Initial point:  [0.02601229 0.02601229]
Search Direction: [-0.00867076 -0.00867076] New Iterate: [0.01734153 0.01734153]
```

```
######################
###   iteration 13   ###
######################

Initial point:  [0.00770735 0.00770735]
Search Direction: [-0.00256912 -0.00256912] New Iterate: [0.00513823 0.00513823]


######################
###   iteration 14   ###
######################

Initial point:  [0.00513823 0.00513823]
Search Direction: [-0.00171274 -0.00171274] New Iterate: [0.00342549 0.00342549]


######################
###   iteration 15   ###
######################

Initial point:  [0.00342549 0.00342549]
Search Direction: [-0.00114183 -0.00114183] New Iterate: [0.00228366 0.00228366]


######################
###   iteration 16   ###
######################

Initial point:  [0.00228366 0.00228366]
Search Direction: [-0.00076122 -0.00076122] New Iterate: [0.00152244 0.00152244]


######################
###   iteration 17   ###
######################

Initial point:  [0.00152244 0.00152244]
Search Direction: [-0.00050748 -0.00050748] New Iterate: [0.00101496 0.00101496]
```

It converges at iteration 17.

Hessians are always positive definite.

b) Gradient-based method with backtracking to minimize $100x_1^4 + 0.01x_2^4$:

```
import numpy as np
from numpy import linalg as LA
#import sys

#stdoutOrigin=sys.stdout
#sys.stdout = open("log.txt", "w")

f = lambda x: (100*x[0]**4 + 0.01*x[1]**4)
dfx1 = lambda x: (400*x[0]**3)
dfx2 = lambda x: (0.04*x[1]**3)


x0 = np.array([1,1])

#Parameters
alpha = 1
beta = 0.5
count = 1
gamma = 0.5


while LA.norm(np.array([[dfx1(x0),dfx2(x0)]])) > 10**(-6):

    print("""

#######################
###   iteration {}   ###
#######################
""".format(count))
    print("Initial point: ", x0)
    d = - np.array([dfx1(x0),dfx2(x0)])
    while (f(x0) - f(x0 + alpha * d)) < (- gamma * alpha * np.dot(np.array([dfx1(x0),dfx2(x0)]),np.transpose(d))):
        alpha = beta * alpha
    else:
        alpha = alpha
    x1 = x0 + alpha * d
    x0 = x1

    print("Search Direction:", d,"step length: ", alpha, "New Iterate:" , x0)

    count +=1
    if count +1 > 1001:
        break
    else:
        continue
```

```
#######################
###   iteration 1   ###
#######################

Initial point:  [ 1 -1]
Sear

#######################
###   iteration 1   ###
#######################

Initial point:  [1 1]
Search Direction: [-4.e+02 -4.e-02] step length:  0.0009765625 New Iterate: [0.609375   0.99996094]


#######################
###   iteration 2   ###
#######################

Initial point:  [0.609375   0.99996094]
Search Direction: [-9.05136108e+01 -3.99953127e-02] step length:  0.0009765625 New Iterate: [0.5209828  0.99992188]


#######################
###   iteration 3   ###
#######################

Initial point:  [0.5209828  0.99992188]
Search Direction: [-5.65627027e+01 -3.99906263e-02] step length:  0.0009765625 New Iterate: [0.46574579 0.99988283]


#######################
###   iteration 4   ###
#######################

Initial point:  [0.46574579 0.99988283]
Search Direction: [-4.04116700e+01 -3.99859408e-02] step length:  0.0009765625 New Iterate: [0.42628127 0.99984378]


#######################
###   iteration 5   ###
#######################

Initial point:  [0.42628127 0.99984378]
Search Direction: [-30.9848025   -0.03998126] step length:  0.0009765625 New Iterate: [0.39602267 0.99980473]


#######################
###   iteration 6   ###
#######################

Initial point:  [0.39602267 0.99980473]
Search Direction: [-24.84392064  -0.03997657] step length:  0.0009765625 New Iterate: [0.37176103 0.99976569]
```

```
#######################
###   iteration 7   ###
#######################

Initial point:  [0.37176103 0.99976569]
Search Direction: [-20.55188091  -0.03997189] step length:  0.0009765625 New Iterate: [0.35169083 0.99972666]


#######################
###   iteration 8   ###
#######################

Initial point:  [0.35169083 0.99972666]
Search Direction: [-17.39975504  -0.03996721] step length:  0.0009765625 New Iterate: [0.33469888 0.99968763]


#######################
###   iteration 9   ###
#######################

Initial point:  [0.33469888 0.99968763]
Search Direction: [-14.99763517  -0.03996253] step length:  0.0009765625 New Iterate: [0.32005276 0.9996486 ]


#######################
###   iteration 10   ###
#######################
```

```
#######################
###   iteration 996   ###
#######################

Initial point:  [0.03574476 0.96325938]
Search Direction: [-0.01826825 -0.03575113] step length:  0.0009765625 New Iterate: [0.03572692 0.96322447]


#######################
###   iteration 997   ###
#######################

Initial point:  [0.03572692 0.96322447]
Search Direction: [-0.01824091 -0.03574724] step length:  0.0009765625 New Iterate: [0.0357091  0.96318956]


#######################
###   iteration 998   ###
#######################

Initial point:  [0.0357091  0.96318956]
Search Direction: [-0.01821364 -0.03574335] step length:  0.0009765625 New Iterate: [0.03569132 0.96315466]


#######################
###   iteration 999   ###
#######################

Initial point:  [0.03569132 0.96315466]
Search Direction: [-0.01818644 -0.03573947] step length:  0.0009765625 New Iterate: [0.03567356 0.96311975]


#######################
###   iteration 1000   ###
#######################

Initial point:  [0.03567356 0.96311975]
Search Direction: [-0.0181593  -0.03573558] step length:  0.0009765625 New Iterate: [0.03565582 0.96308486]
```

Iteration maximum is reached. It is not an efficient method to optimize this function since it needs a great number of runs to get the optimal solution.

c) Newton's method to minimize $\sqrt{x_1^2+1}+\sqrt{x_2^2+1}$

```
f = lambda x: (np.sqrt(x[0]**2+1) + np.sqrt(x[1]**2+1))
dfx1 = lambda x: (x[0]/np.sqrt(x[0]**2+1))
dfx2 = lambda x: (x[0]/np.sqrt(x[1]**2+1))
hx11 = lambda x: (1/(x[0]**2+1)**(3/2))
hx12 = lambda x: (0)
hx21 = lambda x: (0)
hx22 = lambda x: ((1/(x[1]**2+1)**(3/2)))

#parameters
alpha = 1
count = 1

#initial point
x0 = np.array([1,1])


while LA.norm(np.array([[dfx1(x0),dfx2(x0)]])) > 10**(-8):

    print("""

######################
###   iteration {}   ###
######################
""".format(count))
    print("Initial point: ", x0)
    d = -np.dot(np.array([dfx1(x0),dfx2(x0)]), inv(np.array([[hx11(x0),hx12(x0)],
                        [hx21(x0),hx22(x0)]])))

    x0 = x0 + alpha * d
    count += 1
    print("Search Direction:", d,"New Iterate:" , x0)
```

The algorithm starts an infinite loop and cannot converge to an optimal solution. Here is a screen shot of the output:

```
########################
###   iteration 1458   ###
########################

Initial point:  [-1. -1.]
Search Direction: [2. 2.] New Iterate: [1. 1.]


########################
###   iteration 1459   ###
########################

Initial point:  [1. 1.]
Search Direction: [-2. -2.] New Iterate: [-1. -1.]


########################
###   iteration 1460   ###
########################

Initial point:  [-1. -1.]
Search Direction: [2. 2.] New Iterate: [1. 1.]
```

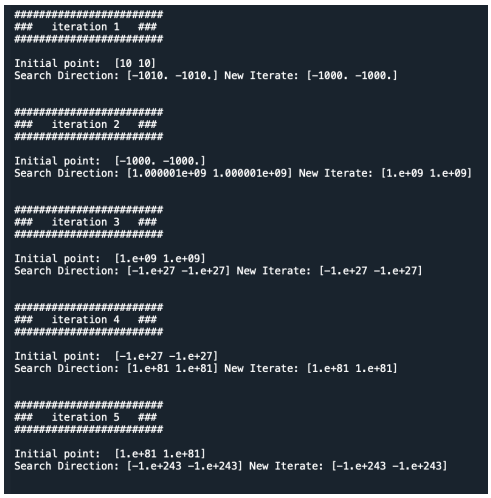
d) Gradient-based method with backtracking to minimize $\sqrt{x_1^2+1}+\sqrt{x_2^2+1}$

```
f = lambda x: (np.sqrt(x[0]**2+1) + np.sqrt(x[1]**2+1))
dfx1 = lambda x: (x[0]/np.sqrt(x[0]**2+1))
dfx2 = lambda x: (x[0]/np.sqrt(x[1]**2+1))

x0 = np.array([1,1])

#Parameters
alpha = 1
beta = 0.5
count = 1
gamma = 0.5


while LA.norm(np.array([dfx1(x0),dfx2(x0)]))/(1+LA.norm(np.array([f(x0)]))) > 10**(-5):

    print("""

######################
###   iteration {}   ###
######################
""".format(count))
    print("Initial point: ", x0)
    d = - np.array([dfx1(x0),dfx2(x0)])    #steepest descent method
    while (f(x0) - f(x0 + alpha * d)) < (- gamma * alpha * np.dot(np.transpose(np.array([dfx1(x0),dfx2(x0)])),d)):
        alpha = beta * alpha
    else:
      alpha = alpha

    x1 = x0 + alpha * d
    x0 = x1

    print("Search Direction:", d,"step length: ", alpha, "New Iterate:" , x0)
    count +=1
    if count +1 > 1001:
        break
    else:
        continue
```

```
######################
###   iteration 1   ###
######################

Initial point:  [1 1]
Search Direction: [-0.70710678 -0.70710678] step length:  1 New Iterate:
[0.29289322 0.29289322]


######################
###   iteration 2   ###
######################

Initial point:  [0.29289322 0.29289322]
Search Direction: [-0.28108464 -0.28108464] step length:  1 New Iterate:
[0.01180858 0.01180858]


######################
###   iteration 3   ###
######################

Initial point:  [0.01180858 0.01180858]
Search Direction: [-0.01180776 -0.01180776] step length:  1 New Iterate:
[8.23223459e-07 8.23223459e-07]
```

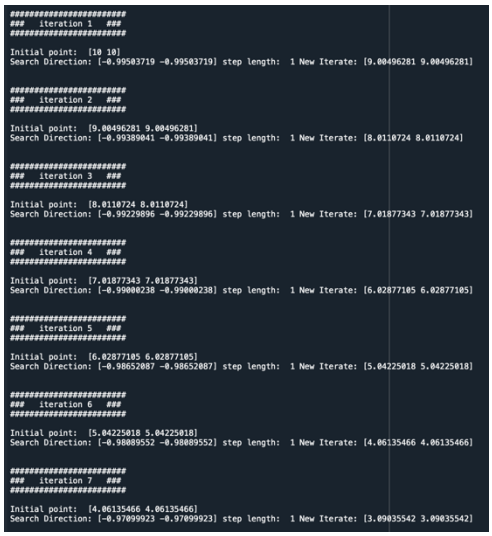
The algorithm converges at iteration 3.

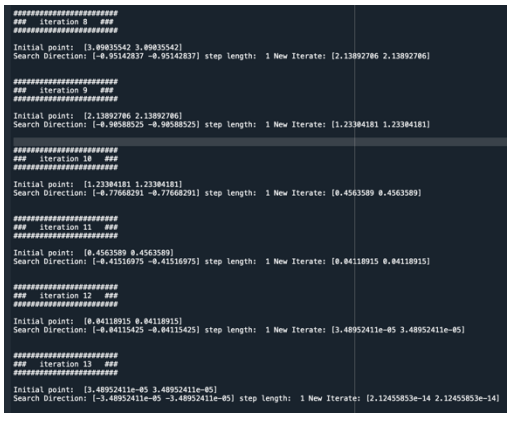
e) Newton's method result ($x_0 = [10, 10]^T$):



The result shows that it is not converged, but diverged when a larger initial point is typed in.

Gradient-based method with backing result ($x_0 = [10, 10]^T$):





It converges at iteration 13.

f) Newton's method with backtracking result ($x_0 = [10, 10]^T$):

```
f = lambda x: (np.sqrt(x[0]**2+1) + np.sqrt(x[1]**2+1))
dfx1 = lambda x: (x[0]/np.sqrt(x[0]**2+1))
dfx2 = lambda x: (x[0]/np.sqrt(x[1]**2+1))
hx11 = lambda x: (1/(x[0]**2+1)**(3/2))
hx12 = lambda x: (0)
hx21 = lambda x: (0)
hx22 = lambda x: ((1/(x[1]**2+1)**(3/2)))

x0 = np.array([10,10])

#Parameters
alpha = 1
beta = 0.5
count = 1
gamma = 0.5


while LA.norm(np.array([[dfx1(x0),dfx2(x0)]])) > 10**(-8):

    print("""

######################
###   iteration {}   ###
######################
""".format(count))
    print("Initial point: ", x0)
    d =  -np.dot(np.array([dfx1(x0),dfx2(x0)]), inv(np.array([[hx11(x0),hx12(x0)],
                    [hx21(x0),hx22(x0)]])))      #newtons method

    while (f(x0) - f(x0 + alpha * d)) < (- gamma * alpha * np.dot(np.transpose(np.array([dfx1(x0),dfx2(x0)])),d)):
        alpha = beta * alpha
    else:
        alpha = alpha
    x1 = x0 + alpha * d
    x0 = x1

    print("Search Direction:", d,"step length: ", alpha, "New Iterate:" , x0)
    count +=1
    if count +1 > 1001:
        break
    else:
        continue
```

```
######################
###   iteration 1   ###
######################

Initial point:  [10 10]
Search Direction: [-1010. -1010.] step length:  0.0078125 New Iterate: [2.109375 2.109375]


######################
###   iteration 2   ###
######################

Initial point:  [2.109375 2.109375]
Search Direction: [-11.49496078 -11.49496078] step length:  0.0078125 New Iterate: [2.01957062 2.01957062]


######################
###   iteration 3   ###
######################

Initial point:  [2.01957062 2.01957062]
Search Direction: [-10.2567236 -10.2567236] step length:  0.0078125 New Iterate: [1.93943997 1.93943997]


######################
###   iteration 4   ###
######################

Initial point:  [1.93943997 1.93943997]
Search Direction: [-9.23450256 -9.23450256] step length:  0.0078125 New Iterate: [1.86729541 1.86729541]


######################
###   iteration 5   ###
######################

Initial point:  [1.86729541 1.86729541]
Search Direction: [-8.37816644 -8.37816644] step length:  0.0078125 New Iterate: [1.80184099 1.80184099]


######################
###   iteration 6   ###
######################

Initial point:  [1.80184099 1.80184099]
Search Direction: [-7.65175371 -7.65175371] step length:  0.0078125 New Iterate: [1.74206166 1.74206166]


######################
###   iteration 7   ###
######################

Initial point:  [1.74206166 1.74206166]
Search Direction: [-7.02883354 -7.02883354] step length:  0.0078125 New Iterate: [1.6871489 1.6871489]


######################
###   iteration 8   ###
######################

Initial point:  [1.6871489 1.6871489]
Search Direction: [-6.48957002 -6.48957002] step length:  0.0078125 New Iterate: [1.63644914 1.63644914]


######################
###   iteration 9   ###
######################

Initial point:  [1.63644914 1.63644914]
Search Direction: [-6.01880391 -6.01880391] step length:  0.0078125 New Iterate: [1.58942723 1.58942723]


######################
###   iteration 10   ###
######################

Initial point:  [1.58942723 1.58942723]
Search Direction: [-5.60476374 -5.60476374] step length:  0.0078125 New Iterate: [1.54564001 1.54564001]
```

```
######################
###   iteration 996   ###
######################

Initial point:  [0.00036793 0.00036793]
Search Direction: [-0.00036793 -0.00036793] step length:  0.0078125 New Iterate: [0.00036505 0.00036505]


######################
###   iteration 997   ###
######################

Initial point:  [0.00036505 0.00036505]
Search Direction: [-0.00036505 -0.00036505] step length:  0.0078125 New Iterate: [0.0003622 0.0003622]


######################
###   iteration 998   ###
######################

Initial point:  [0.0003622 0.0003622]
Search Direction: [-0.0003622 -0.0003622] step length:  0.0078125 New Iterate: [0.00035937 0.00035937]


######################
###   iteration 999   ###
######################

Initial point:  [0.00035937 0.00035937]
Search Direction: [-0.00035937 -0.00035937] step length:  0.0078125 New Iterate: [0.00035656 0.00035656]


######################
###   iteration 1000   ###
######################

Initial point:  [0.00035656 0.00035656]
Search Direction: [-0.00035656 -0.00035656] step length:  0.0078125 New Iterate: [0.00035378 0.00035378]
```

Iteration maximum is reached

Here is a summary for (c), (d), (e) and (f):

Method	**Initial Point**	**# of iteration**
Newton's	(1,1)	Diverged (error)
Gradient-based with backtracking	(1,1)	3
Newton's	(10,10)	Diverged
Gradient-based with backtracking	(10,10)	13
Newton's with backtracking	(10,10)	1000+