Introduction

The gradient-based method with backtracking is a way to find the minimum of a multivariate function. The gradient-based method is an iterative strategy that uses the gradient or negative of the gradient as a search direction, the update rule of gradient-based method is shown in formula (1) and formula (2), where $d^{(k)}$ is the search direction for current iteration and $\alpha^{(k)}$ is the step length for current iteration. The backtracking method enables gradient-based method to change the step length based on the sufficient decrease condition which is shown in formula (3). If the condition is satisfied with current step length, then the update will be performed using current step length, otherwise the step length will be adjusted using formula (4) until the sufficient decrease condition is satisfied. Initial guess of step length, α, step length adjusting parameter, β, and sufficient decrease condition parameter, γ, are self-defined parameters that can affect the convergence efficiency.

$$d^{(k)} = -\nabla f(x^{(k)}) \qquad (1)$$

$$x^{(k+1)} \leftarrow x^{(k)} - \alpha^{(k)} \nabla f\left(x^{(k)}\right) \qquad (2)$$

$$f\left(x^{(k)}\right) - f\left(x^{(k)} + \alpha^{(k)} d^{(k)}\right) \geq -\gamma \alpha^{(k)} \nabla f\left(x^{(k)}\right)^T d^{(k)} \qquad (3)$$

$$\alpha^{(k)} = \beta \alpha^{(k)} \qquad (4)$$

In practice, the algorithm might not reach the actual minimum because of the limited precision of computer. Therefore, a stop criterion needs to be implemented to stop the algorithm at certain point. In the implementation, the algorithm will stop when the number of iterations reaches 1000 or the stopping condition which is shown in formula (5) is met. The gradient-based method with backtracking is implemented using Python and tested with five functions, the code is shown in Appendix A.

$$\|\nabla f(x)\| / (1 + |f(x)|) \leq 10^{-5} \qquad (5)$$

Result

There are five functions used to test the implemented gradient-based method with backtracking. For each testing, the initial step length, α, is set to be 1, the step length adjusting parameter, β, is set to be 0.5, and the sufficient decrease parameter, γ, is set to be 0.3. The result of each testing function is shown below, the detailed output for each function is shown in Appendix B.

Function 1: $f_1(x) = {x_1}^2 + {x_2}^2 + {x_3}^2$
Initial point: $[1,1,1]^T$
Solution found: $[0,0,0]^T$
Iterations used: 1

Function 2: $f_2(x) = {x_1}^2 + 2{x_2}^2 - 2x_1x_2 - 2x_2$
Initial point: $[0,0]^T$
Solution found: $[0.99998474, 0.99999237]^T$

Iterations used: 34

Function 3: $f_3(x) = 100(x_2 - {x_1}^2)^2 + (1 - x_1)^2$
Initial point: $[-1.2,1]^T$
Solution found: $[1.0000106, 1.00002124]^T$
Iterations used: 292

Function 4: $f_4(x) = (x_1 + x_2)^4 + {x_2}^2$
Initial point: $[2, -2]^T$
Solution found: $[\ 1.35714901\mathrm{e}-02, -4.90320909\mathrm{e}-06]^T$
Iterations used: 691

Function 5: $f_5(x) = (x_1 - 1)^2 + (x_2 - 1)^2 + c({x_1}^2 + {x_2}^2 - 0.25)^2$
Initial point: $[1, -1]^T$

With c=1:

Solution found: $[\ 0.56408669, 0.56408569]^T$
Iterations used: 10

With c=10:

Solution found: $[\ 0.40260809, 0.40261189]^T$
Iterations used: 19

With c=100:

Solution found: $[\ 0.35979134, 0.35978779]^T$
Iterations used: 210

Analysis

For quadratic function $x^T Q x$, the convergence speed is related to the condition number of Q and for non-quadratic function, the convergence speed is related to the condition number of the Hessian at the stationary point, the condition number is the ratio of the largest eigenvalue of a matrix and the smallest eigenvalue of a matrix.

Function 1 is a quadratic function, the Q matrix is: $\begin{vmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{vmatrix}$, which gives condition number of 1; function 2 is a quadratic function, the Q matrix is : $\begin{vmatrix} 1 & -1 \\ -1 & 2 \end{vmatrix}$, which gives condition number of 6.9, the Hessian matrix of function 3 at the solution is: $\begin{vmatrix} 802 & -400 \\ -400 & 200 \end{vmatrix}$, which gives condition number of 2508; the Hessian matrix of function 4 at the solution is: $\begin{vmatrix} 0 & 0 \\ 0 & 2 \end{vmatrix}$, which gives condition number of infinite; the Hessian matrix of function 5 with c=1 at the solution is: $\begin{vmatrix} 6 & 2.5 \\ 2.5 & 6 \end{vmatrix}$, which gives condition number of 2.4; the Hessian matrix of function 5 with c=10 at the solution is: $\begin{vmatrix} 17.6 & 12.8 \\ 12.8 & 17.6 \end{vmatrix}$, which gives condition number of 6.1; the Hessian matrix of function 5 with c=100 at the solution is: $\begin{vmatrix} 109.4 & 103.7 \\ 103.7 & 109.4 \end{vmatrix}$, which gives condition number of 37.4.

From the results above, it is obvious that convergence speed is slow if the condition number of Hessian at the solution (stationary point) or matrix Q of quadratic function is large and vice versa, which follows the theory of gradient-based method. Also, for function 5, the value of c affects the convergence speed, with greater c the algorithm will converge slower. The reason is that the value of c affects the Hessian matrix at stationary point, with greater c the Hessian matrix at stationary point will become more ill-conditioned, therefore the condition number will become larger which indicates slower convergence.

Appendix A

```
def backtracking(function,gradient,initial_s,gamma,beta,initial_x
):
    """
    Description:
    Gradient-based method with backtracking to
    find the minimum of a multivariant function
    Parameters:
    function: objective function to be minimized
    gradient: gradient of objective function
    initial_s: initial guess of step length
    gamma: real number between 0 and 1 to
    test if the function is decreased sufficiently
    beta: real number between 0 and 1 to decrease the
    step length if the original one does not make
    sufficient decrease
    initial_x: initial point
    """
    # import library
    import numpy as np
    # set maximum number of iteration
    max_iter=1000
    # set stop criteria
    stop_criteria=1E-5
    # print initial point
    print('Initial point: {}.T'.format(initial_x.reshape(1,-1)))
    # set x to initial point
    x_k=initial_x
    # set s to initial s
    s=initial_s
    # create lists to store result
    d_k_list=[]
    s_list=[]
    x_k_list=[]
    # iterate until maximum iteration is reached
    for iter in range(max_iter):
        # calcualte gradient value
        gradient_k=gradient(x_k)
        # calcualte function value
        function_k=function(x_k)
        # set s to initial s
        s=initial_s
        # check stop criteria
```

```
        if ((np.linalg.norm(gradient_k)/(1+np.abs(function_k))) >
 stop_criteria):
            # descent direction
            d_k=-gradient_k
            # next point
            x_k_plus_one=x_k+s*d_k
            # check if the sufficient decrease condition is met
            while (function(x_k) - function(x_k-s*gradient(x_k)))
 < (-gamma*s*np.dot(gradient_k.T,d_k)):
                # update step length
                s=beta*s
                # re-calculate next point
                x_k_plus_one=x_k+s*d_k
            # update x
            x_k=x_k_plus_one
            # put result in lists
            d_k_list.append(d_k)
            s_list.append(s)
            x_k_list.append(x_k)
        # if the stop criteria is satisfied
        else:
            # print out the solution
            print('Solution found: {}.T'.format(x_k.reshape(1,-1)
))
            # break the iteration
            break
    # print result
    # if iteration less than 15 print each iteration
    if iter < 15:
        for i in range(iter):
            print('=====Iteration{}====='.format(i+1))
            # print search direction
            print('Search direction: {}.T'.format(d_k_list[i].res
hape(1,-1)))
            # print step length
            print('Step length: {}'.format(s_list[i]))
            # print new iterate
            print('New iterate: {}.T'.format(x_k_list[i].reshape(
1,-1)))
    # if iteration is greater than 15
    else:
        # print first 10 iterations
        for i in range(10):
            print('=====Iteration{}====='.format(i+1))
```

```
            # print search direction
            print('Search direction: {}.T'.format(d_k_list[i].res
hape(1,-1)))
            # print step length
            print('Step length: {}'.format(s_list[i]))
            # print new iterate
            print('New iterate: {}.T'.format(x_k_list[i].reshape(
1,-1)))
        print('''
        ..........
        ..........''')
        # print last 5 iterations
        for i in range(5):
            print('=====Iteration{}====='.format(iter-3+i))
            # print search direction
            print('Search direction: {}.T'.format(d_k_list[iter-5
+i].reshape(1,-1)))
            # print step length
            print('Step length: {}'.format(s_list[iter-5+i]))
            # print new iterate
            print('New iterate: {}.T'.format(x_k_list[iter-5+i].r
eshape(1,-1)))
        # print warning if maximum iteration is reached
        if iter == 999:
            print('====================')
            print('Maximum iteration is reached !!!')
```

Function 1:

```
# define function
def function_1(x):
    x1=x[0]
    x2=x[1]
    x3=x[2]
    return x1**2+x2**2+x3**2
# define gradient
def gradient_1(x):
    x1=x[0]
    x2=x[1]
    x3=x[2]
    gradient=np.array([2*x1,2*x2,2*x3]).reshape(-1,1)
    return gradient
# initial point (1,1,1).T
initial=np.array([1,1,1]).reshape(-1,1)
backtracking(function_1,gradient_1,1,0.3,0.5,initial)
```

Function 2:

```
# define function
def function_2(x):
    x1=x[0]
    x2=x[1]
    return x1**2+2*x2**2-2*x1*x2-2*x2
# define gradient
def gradient_2(x):
    x1=x[0]
    x2=x[1]
    gradient=np.array([2*x1-2*x2,4*x2-2*x1-2]).reshape(-1,1)
    return gradient
# initial point (0,0).T
initial=np.array([0,0]).reshape(-1,1)
backtracking(function_2,gradient_2,1,0.3,0.5,initial)
```

Function 3:

```
# define function
def function_3(x):
    x1=x[0]
    x2=x[1]
    return 100*(x2-x1**2)**2+(1-x1)**2
# define gradient
def gradient_3(x):
    x1=x[0]
    x2=x[1]
    gradient=np.array([-400*x1*(x2-x1**2)-2*(1-x1),200*(x2-x1**2)
]).reshape(-1,1)
    return gradient
# initial point (-1.2,1).T
initial=np.array([-1.2,1]).reshape(-1,1)
backtracking(function_3,gradient_3,1,0.3,0.5,initial)
```

Function 4:

```
# define function
def function_4(x):
    x1=x[0]
    x2=x[1]
    return (x1+x2)**4+x2**2
# define gradient
def gradient_4(x):
    x1=x[0]
    x2=x[1]
```

```
    gradient=np.array([4*(x1+x2)**3,4*(x1+x2)**3+2*x2]).reshape(-
1,1)
    return gradient
# initial point (2,-2).T
initial=np.array([2,-2]).reshape(-1,1)
backtracking(function_4,gradient_4,1,0.3,0.5,initial)
```

Function 5 with c=1:

```
# define function
def function_5_1(x):
    x1=x[0]
    x2=x[1]
    return (x1-1)**2+(x2-1)**2+(x1**2+x2**2-0.25)**2
# define gradient
def gradient_5_1(x):
    x1=x[0]
    x2=x[1]
    gradient=np.array([2*(x1-1)+4*x1*(x1**2+x2**2-0.25),
                                2*(x2-1)+4*x2*(x1**2+x2**2-0.
25)]).reshape(-1,1)
    return gradient
# initial point (1,-1).T
initial=np.array([1,-1]).reshape(-1,1)
backtracking(function_5_1,gradient_5_1,1,0.3,0.5,initial)
```

Function 5 with c=10:

```
# define function
def function_5_2(x):
    x1=x[0]
    x2=x[1]
    return (x1-1)**2+(x2-1)**2+10*(x1**2+x2**2-0.25)**2
# define gradient
def gradient_5_2(x):
    x1=x[0]
    x2=x[1]
    gradient=np.array([2*(x1-1)+40*x1*(x1**2+x2**2-0.25),
                                2*(x2-1)+40*x2*(x1**2+x2**2-0
.25)]).reshape(-1,1)
    return gradient
# initial point (1,-1).T
initial=np.array([1,-1]).reshape(-1,1)
backtracking(function_5_2,gradient_5_2,1,0.3,0.5,initial)
```

Function 5 with c=100:

```
# define function
def function_5_3(x):
    x1=x[0]
    x2=x[1]
    return (x1-1)**2+(x2-1)**2+100*(x1**2+x2**2-0.25)**2
# define gradient
def gradient_5_3(x):
    x1=x[0]
    x2=x[1]
    gradient=np.array([2*(x1-1)+400*x1*(x1**2+x2**2-0.25),
                                   2*(x2-1)+400*x2*(x1**2+x2**2-
0.25)]).reshape(-1,1)
    return gradient
# initial point (1,-1).T
initial=np.array([1,-1]).reshape(-1,1)
backtracking(function_5_3,gradient_5_3,1,0.3,0.5,initial)
```

Appendix B

Output with function 1:

```
Initial point: [[1 1 1]].T
Solution found: [[0. 0. 0.]].T
=====Iteration1=====
Search direction: [[-2 -2 -2]].T
Step length: 0.5
New iterate: [[0. 0. 0.]].T
```

Output with function 2:

```
Initial point: [[0 0]].T
Solution found: [[0.99998474 0.99999237]].T
=====Iteration1=====
Search direction: [[0 2]].T
Step length: 0.25
New iterate: [[0.  0.5]].T
=====Iteration2=====
Search direction: [[ 1. -0.]].T
Step length: 0.5
New iterate: [[0.5 0.5]].T
=====Iteration3=====
Search direction: [[-0.  1.]].T
Step length: 0.25
New iterate: [[0.5  0.75]].T
=====Iteration4=====
Search direction: [[ 0.5 -0. ]].T
Step length: 0.5
New iterate: [[0.75 0.75]].T
=====Iteration5=====
Search direction: [[-0.   0.5]].T
Step length: 0.25
New iterate: [[0.75  0.875]].T
=====Iteration6=====
Search direction: [[ 0.25 -0.  ]].T
Step length: 0.5
New iterate: [[0.875 0.875]].T
=====Iteration7=====
Search direction: [[-0.    0.25]].T
Step length: 0.25
New iterate: [[0.875  0.9375]].T
=====Iteration8=====
Search direction: [[ 0.125 -0.   ]].T
Step length: 0.5
```

```
New iterate: [[0.9375 0.9375]].T
=====Iteration9=====
Search direction: [[-0.    0.125]].T
Step length: 0.25
New iterate: [[0.9375  0.96875]].T
=====Iteration10=====
Search direction: [[ 0.0625 -0.    ]].T
Step length: 0.5
New iterate: [[0.96875 0.96875]].T

       ..........
       ..........
=====Iteration30=====
Search direction: [[-0.         0.00012207]].T
Step length: 0.25
New iterate: [[0.99993896 0.99996948]].T
=====Iteration31=====
Search direction: [[ 6.10351562e-05 -0.00000000e+00]].T
Step length: 0.5
New iterate: [[0.99996948 0.99996948]].T
=====Iteration32=====
Search direction: [[-0.00000000e+00  6.10351562e-05]].T
Step length: 0.25
New iterate: [[0.99996948 0.99998474]].T
=====Iteration33=====
Search direction: [[ 3.05175781e-05 -0.00000000e+00]].T
Step length: 0.5
New iterate: [[0.99998474 0.99998474]].T
=====Iteration34=====
Search direction: [[-0.00000000e+00  3.05175781e-05]].T
Step length: 0.25
New iterate: [[0.99998474 0.99999237]].T
```

Output with function 3:

```
Initial point: [[-1.2  1. ]].T
Solution found: [[1.0000106  1.00002124]].T
=====Iteration1=====
Search direction: [[215.6  88. ]].T
Step length: 0.0009765625
New iterate: [[-0.98945312  1.0859375 ]].T
=====Iteration2=====
Search direction: [[-38.33803031 -21.38400269]].T
Step length: 0.0009765625
New iterate: [[-1.02689261  1.06505468]].T
```

```
=====Iteration3=====
Search direction: [[-0.27816415 -2.10925141]].T
Step length: 0.00390625
New iterate: [[-1.02797919  1.05681542]].T
=====Iteration4=====
Search direction: [[ 4.02544229 -0.01484275]].T
Step length: 0.5
New iterate: [[0.98474196 1.04939405]].T
=====Iteration5=====
Search direction: [[ 31.41515509 -15.93546343]].T
Step length: 0.0009765625
New iterate: [[1.01542082 1.03383207]].T
=====Iteration6=====
Search direction: [[ 1.08718639 -0.55052448]].T
Step length: 0.0009765625
New iterate: [[1.01648253 1.03329445]].T
=====Iteration7=====
Search direction: [[-0.00949734 -0.01154359]].T
Step length: 0.015625
New iterate: [[1.01633413 1.03311408]].T
=====Iteration8=====
Search direction: [[ 0.0401058  -0.03580224]].T
Step length: 0.0009765625
New iterate: [[1.0163733  1.03307912]].T
=====Iteration9=====
Search direction: [[-0.00655043 -0.01288708]].T
Step length: 1
New iterate: [[1.00982287 1.02019204]].T
=====Iteration10=====
Search direction: [[ 0.16204633 -0.08996234]].T
Step length: 0.0009765625
New iterate: [[1.00998112 1.02010418]].T

       ..........
       ..........
=====Iteration288=====
Search direction: [[ 1.73867181e-05 -2.19493376e-05]].T
Step length: 0.0009765625
New iterate: [[1.00001327 1.0000266 ]].T
=====Iteration289=====
Search direction: [[-4.80504735e-06 -1.08705808e-05]].T
Step length: 0.5
New iterate: [[1.00001087 1.00002117]].T
=====Iteration290=====
```

```
Search direction: [[-0.00025208  0.00011517]].T
Step length: 0.0009765625
New iterate: [[1.00001062 1.00002128]].T
=====Iteration291=====
Search direction: [[-9.65783514e-06 -5.79557717e-06]].T
Step length: 0.00390625
New iterate: [[1.00001059 1.00002126]].T
=====Iteration292=====
Search direction: [[ 1.15433056e-05 -1.63583100e-05]].T
Step length: 0.0009765625
New iterate: [[1.0000106  1.00002124]].T
```

Output with function 4:

```
Initial point: [[ 2 -2]].T
Solution found: [[ 1.35714901e-02 -4.90320909e-06]].T
=====Iteration1=====
Search direction: [[0 4]].T
Step length: 0.25
New iterate: [[ 2. -1.]].T
=====Iteration2=====
Search direction: [[-4. -2.]].T
Step length: 0.0625
New iterate: [[ 1.75  -1.125]].T
=====Iteration3=====
Search direction: [[-0.9765625  1.2734375]].T
Step length: 0.5
New iterate: [[ 1.26171875 -0.48828125]].T
=====Iteration4=====
Search direction: [[-1.85069847 -0.87413597]].T
Step length: 0.125
New iterate: [[ 1.03038144 -0.59754825]].T
=====Iteration5=====
Search direction: [[-0.3243558   0.87074069]].T
Step length: 0.25
New iterate: [[ 0.94929249 -0.37986307]].T
=====Iteration6=====
Search direction: [[-0.73854964  0.02117651]].T
Step length: 0.5
New iterate: [[ 0.58001767 -0.36927482]].T
=====Iteration7=====
Search direction: [[-0.03743851  0.70111112]].T
Step length: 0.25
New iterate: [[ 0.57065804 -0.19399704]].T
=====Iteration8=====
```

```
Search direction: [[-0.21375289  0.17424119]].T
Step length: 1
New iterate: [[ 0.35690516 -0.01975585]].T
=====Iteration9=====
Search direction: [[-0.15329459 -0.11378289]].T
Step length: 0.5
New iterate: [[ 0.28025787 -0.07664729]].T
=====Iteration10=====
Search direction: [[-0.03376455  0.11953004]].T
Step length: 0.5
New iterate: [[ 0.26337559 -0.01688227]].T

       ..........
       ..........
=====Iteration687=====
Search direction: [[-1.00987376e-05  2.01569716e-07]].T
Step length: 1
New iterate: [[ 1.36116625e-02 -4.94858395e-06]].T
=====Iteration688=====
Search direction: [[-1.00767330e-05 -1.79565062e-07]].T
Step length: 1
New iterate: [[ 1.36015857e-02 -5.12814901e-06]].T
=====Iteration689=====
Search direction: [[-1.00539636e-05  2.02334430e-07]].T
Step length: 1
New iterate: [[ 1.35915318e-02 -4.92581458e-06]].T
=====Iteration690=====
Search direction: [[-1.00321249e-05 -1.80495763e-07]].T
Step length: 1
New iterate: [[ 1.35814996e-02 -5.10631034e-06]].T
=====Iteration691=====
Search direction: [[-1.00095194e-05  2.03101253e-07]].T
Step length: 1
New iterate: [[ 1.35714901e-02 -4.90320909e-06]].T
```

Output with function 5 with c=1:

```
Initial point: [[ 1 -1]].T
Solution found: [[0.56408669 0.56408569]].T
=====Iteration1=====
Search direction: [[-7. 11.]].T
Step length: 0.0625
New iterate: [[ 0.5625 -0.3125]].T
=====Iteration2=====
Search direction: [[0.50585938 2.83007812]].T
```

```
Step length: 0.25
New iterate: [[0.68896484 0.39501953]].T
=====Iteration3=====
Search direction: [[-0.42712114  0.60840468]].T
Step length: 0.25
New iterate: [[0.58218456 0.5471207 ]].T
=====Iteration4=====
Search direction: [[-0.06857142  0.05601467]].T
Step length: 0.25
New iterate: [[0.5650417  0.56112437]].T
=====Iteration5=====
Search direction: [[0.00171264 0.01556643]].T
Step length: 0.125
New iterate: [[0.56525579 0.56307017]].T
=====Iteration6=====
Search direction: [[-0.00453747  0.00321326]].T
Step length: 0.25
New iterate: [[0.56412142 0.56387349]].T
=====Iteration7=====
Search direction: [[0.00033335 0.0012122 ]].T
Step length: 0.125
New iterate: [[0.56416309 0.56402501]].T
=====Iteration8=====
Search direction: [[-0.00030612  0.00018344]].T
Step length: 0.25
New iterate: [[0.56408656 0.56407087]].T
=====Iteration9=====
Search direction: [[4.33237111e-05 9.89335031e-05]].T
Step length: 0.125
New iterate: [[0.56409197 0.56408324]].T
=====Iteration10=====
Search direction: [[-2.11420106e-05  9.82306802e-06]].T
Step length: 0.25
New iterate: [[0.56408669 0.56408569]].T
```

Output with function 5 with c=10:

```
Initial point: [[ 1 -1]].T
Solution found: [[0.40260809 0.40261189]].T
=====Iteration1=====
Search direction: [[-70.  74.]].T
Step length: 0.0078125
New iterate: [[ 0.453125 -0.421875]].T
=====Iteration2=====
Search direction: [[-1.32232666  5.09320068]].T
```

```
Step length: 0.125
New iterate: [[0.28783417 0.21477509]].T
=====Iteration3=====
Search direction: [[2.81771562 2.61016016]].T
Step length: 0.0625
New iterate: [[0.46394139 0.3779101 ]].T
=====Iteration4=====
Search direction: [[-0.9331796  -0.38926338]].T
Step length: 0.03125
New iterate: [[0.43477953 0.36574562]].T
=====Iteration5=====
Search direction: [[-0.1356909   0.20341224]].T
Step length: 0.125
New iterate: [[0.41781817 0.39117215]].T
=====Iteration6=====
Search direction: [[-0.13233786  0.0036503 ]].T
Step length: 0.0625
New iterate: [[0.40954705 0.39140029]].T
=====Iteration7=====
Search direction: [[0.01905408 0.10682851]].T
Step length: 0.0625
New iterate: [[0.41073793 0.39807707]].T
=====Iteration8=====
Search direction: [[-0.08935821 -0.02495443]].T
Step length: 0.03125
New iterate: [[0.40794549 0.39729725]].T
=====Iteration9=====
Search direction: [[-0.02772769  0.02520031]].T
Step length: 0.25
New iterate: [[0.40101357 0.40359732]].T
=====Iteration10=====
Search direction: [[0.0157419 0.0029572]].T
Step length: 0.03125
New iterate: [[0.4015055  0.40368973]].T

       ..........
       ..........
=====Iteration15=====
Search direction: [[0.00078815 0.00025233]].T
Step length: 0.03125
New iterate: [[0.40256387 0.40265501]].T
=====Iteration16=====
Search direction: [[ 0.00024427 -0.00020849]].T
Step length: 0.25
```

```
New iterate: [[0.40262494 0.40260289]].T
=====Iteration17=====
Search direction: [[-1.75040252e-04 -6.55170565e-05]].T
Step length: 0.03125
New iterate: [[0.40261947 0.40260084]].T
=====Iteration18=====
Search direction: [[-5.03795106e-05  4.21362575e-05]].T
Step length: 0.25
New iterate: [[0.40260687 0.40261138]].T
=====Iteration19=====
Search direction: [[3.89129076e-05 1.65337616e-05]].T
Step length: 0.03125
New iterate: [[0.40260809 0.40261189]].T
```

Output with function 5 with c=100:

```
Initial point: [[ 1 -1]].T
Solution found: [[0.35979134 0.35978779]].T
=====Iteration1=====
Search direction: [[-700.  704.]].T
Step length: 0.0009765625
New iterate: [[ 0.31640625 -0.3125    ]].T
=====Iteration2=====
Search direction: [[ 7.97765255 -3.90385437]].T
Step length: 0.0078125
New iterate: [[ 0.37873166 -0.34299886]].T
=====Iteration3=====
Search direction: [[-0.43689437  4.20697682]].T
Step length: 0.0078125
New iterate: [[ 0.37531842 -0.31013186]].T
=====Iteration4=====
Search direction: [[3.1941601  1.01324565]].T
Step length: 0.015625
New iterate: [[ 0.42522717 -0.29429989]].T
=====Iteration5=====
Search direction: [[-1.81523636  4.64052656]].T
Step length: 0.0078125
New iterate: [[ 0.41104564 -0.25804578]].T
=====Iteration6=====
Search direction: [[3.55438672 1.02418882]].T
Step length: 0.0078125
New iterate: [[ 0.43881429 -0.2500443 ]].T
=====Iteration7=====
Search direction: [[0.23067757 3.00819186]].T
Step length: 0.125
```

```
New iterate: [[0.46764898 0.12597968]].T
=====Iteration8=====
Search direction: [[3.95169534 2.52576604]].T
Step length: 0.00390625
New iterate: [[0.48308529 0.13584595]].T
=====Iteration9=====
Search direction: [[0.68107594 1.62911209]].T
Step length: 0.015625
New iterate: [[0.49372711 0.16130083]].T
=====Iteration10=====
Search direction: [[-2.89469436  0.40090153]].T
Step length: 0.0078125
New iterate: [[0.47111231 0.16443287]].T

       ..........
       ..........
=====Iteration206=====
Search direction: [[3.05613652e-06 2.88695869e-05]].T
Step length: 0.0078125
New iterate: [[0.3597918  0.35978736]].T
=====Iteration207=====
Search direction: [[-2.29062228e-05  1.78652437e-06]].T
Step length: 0.0078125
New iterate: [[0.35979162 0.35978737]].T
=====Iteration208=====
Search direction: [[-4.8242631e-06  1.8795916e-05]].T
Step length: 0.015625
New iterate: [[0.35979155 0.35978767]].T
=====Iteration209=====
Search direction: [[-2.70127789e-05 -5.44391122e-06]].T
Step length: 0.0078125
New iterate: [[0.35979134 0.35978762]].T
=====Iteration210=====
Search direction: [[4.19758568e-07 2.10516515e-05]].T
Step length: 0.0078125
New iterate: [[0.35979134 0.35978779]].T
```