BOĞAZIÇI UNIVERSITY

NONLINEAR MODELS IN OPERATIONS RESEARCH
IE 440

Homework 3

Authors:
M. Akın Elden
Yunus Emre Karataş
Y. Harun Kıvrıl
Sefa Kayraklık

18 November 2019

Department of Industrial Engineering
Boğaziçi University

1 Introduction

The project is implemented using Python as the programming language. First the given function is converted to a lambda function using "sympy" package. Then its plotted between the points [0,15] and [10,50].

The source code used to import required dependencies, converting function to a lambda expression:

```
import pandas as pd
import numpy as np
from sympy import Symbol, cos, sin, lambdify

x1  = Symbol('x1')
x2 = Symbol('x2')
function = (5*x1-x2)**4+((x1-2)**2)+(x1-2*x2)+12
f = lambdify([[x1,x2]], function, 'numpy')
```

The graph of the function:

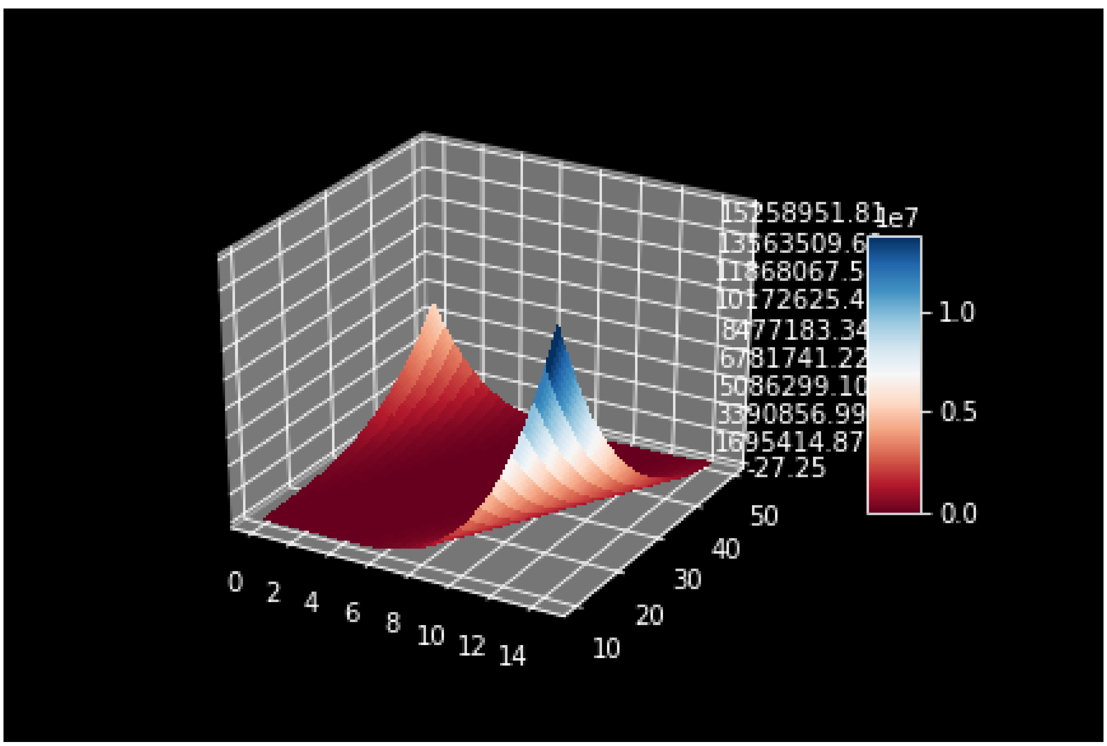

Figure 1: The graph of the given function

The source used to plot the graph of the function:

```
from pylab import meshgrid,cm,imshow,contour,clabel,colorbar,axis,title,show
from mpl_toolkits.mplot3d import Axes3D
from matplotlib import cm
```

```
from matplotlib.ticker import LinearLocator, FormatStrFormatter
import matplotlib.pyplot as plt

# plot the function
x = np.arange(0,15,0.5)
y = np.arange(10,50,0.5)
X,Y = meshgrid(x, y) # grid of point
Z = f([X,Y]) # evaluation of the function on the grid

fig = plt.figure()
ax = fig.gca(projection='3d')
surf = ax.plot_surface(X, Y, Z, rstride=1, cstride=1,
                    cmap=cm.RdBu,linewidth=0, antialiased=False)

ax.zaxis.set_major_locator(LinearLocator(10))
ax.zaxis.set_major_formatter(FormatStrFormatter('%.02f'))

fig.colorbar(surf, shrink=0.5, aspect=5)

plt.show()
```

Exact Line Search

Since Cyclic coordinate search and Hook & Jeeves method requires an exact line search, an algorithm is used to implement that search. Bisection method is used to find the approximately optimum point in the interval [-100, 100].

The source code to implement exact line search:

```
def BisectionMethod(f, a=-100,b=100,epsilon=0.005) :
    iteration=0
    while (b - a) >= epsilon:
        x_1 = (a + b) / 2
        fx_1 = f(x_1)
        if f(x_1 + epsilon) <= fx_1:
            a = x_1
        else:
            b = x_1
        iteration+=1
    x_star = (a+b)/2
    return x_star

def ExactLineSearch(f, x0, d):
    alpha = Symbol('alpha')
    function_alpha = f(np.array(x0)+alpha*np.array(d))
    f_alp = lambdify(alpha, function_alpha, 'numpy')
    alp_star = BisectionMethod(f_alp)
    return alp_star

def np_str(x_k):
    # Used to convert numpy array to string with determined format
```

```
    return np.array2string(x_k, precision=2, separator=',')
```

2 Cyclic Coordinate Search

Cyclic coordinate search is used to find a local optimum point. Since the function is in 2 dimensional space, each iteration consists of 2 steps in the directions of unit vectors. In each step the optimal step length is found using exact line search. When the magnitude of the difference between two consecutive iterates is less than a given ε value, the algorithm ends.

Source code of the algorithm:

```
def CyclicCoordinateSearch(f, x0, epsilon):
    x0 = np.array(x0)
    x_array = [x0]
    k = 0
    n = len(x0)
    res_array = []
    while(True):
        y0 = np.copy(x_array[k])
        for j in range(n):
            d = np.zeros(n)
            d[j] = 1
            alpha = ExactLineSearch(f, y0, d)
            y1 = y0 + alpha*d
            res_array.append([k, np_str(x_array[k]), f(x_array[k]),j, str(d),np_str(y0),
                alpha, np_str(y1)])
            y0 = y1
        x_array.append(y1)
        k += 1
        if(np.linalg.norm(x_array[k]-x_array[k-1]) < epsilon):
            res_array.append([k, np_str(x_array[k]), f(x_array[k])])
            result_table = pd.DataFrame(res_array, columns=['k' ,'x^k', 'fx^k', 'j','d^j'
                ,'y^j','a^j', 'y^j+1'])
            return result_table
```

Solution set 1:

- $x^{(0)} = [0, 0]$
- $\varepsilon = 0.01$

Output of the solution set 1:

k	$x^{(k)}$	$f(x^{(k)})$	j	$d_j^{(k)}$	y_j	$\alpha_j^{(k)}$	y_{j+1}
0	[0 0]	16.000	0	[1. 0.]	[0 0]	0.102	[0.1 0.]
0	[0,0]	16.0000	0	[1. 0.]	[0,0]	0.1022	[0.1,0.]
0	[0,0]	16.0000	1	[0. 1.]	[0.1,0.]	1.3016	[0.1,1.3]
1	[0.1,1.3]	13.4909	0	[1. 0.]	[0.1,1.3]	0.2518	[0.35,1.3]
1	[0.1,1.3]	13.4909	1	[0. 1.]	[0.35,1.3]	1.2589	[0.35,2.56]
2	[0.35,2.56]	10.3328	0	[1. 0.]	[0.35,2.56]	0.2457	[0.6 ,2.56]
2	[0.35,2.56]	10.3328	1	[0. 1.]	[0.6 ,2.56]	1.2283	[0.6 ,3.79]
3	[0.6 ,3.79]	7.3734	0	[1. 0.]	[0.6 ,3.79]	0.2365	[0.84,3.79]
3	[0.6 ,3.79]	7.3734	1	[0. 1.]	[0.84,3.79]	1.1826	[0.84,4.97]
4	[0.84,4.97]	4.6383	0	[1. 0.]	[0.84,4.97]	0.2274	[1.06,4.97]
4	[0.84,4.97]	4.6383	1	[0. 1.]	[1.06,4.97]	1.1368	[1.06,6.11]
...	...	...	...	...	...	...	...
226	[5.92,30.38]	-27.101	0	[1. 0.]	[5.92,30.38]	0.005	[5.92,30.38]
226	[5.92,30.38]	-27.101	1	[0. 1.]	[5.92,30.38]	0.023	[5.92,30.4]
227	[5.92,30.4]	-27.106	0	[1. 0.]	[5.92,30.4]	0.005	[5.93,30.4]
227	[5.92,30.4]	-27.106	1	[0. 1.]	[5.93,30.4]	0.023	[5.93,30.42]
228	[5.93,30.42]	-27.112	0	[1. 0.]	[5.93,30.42]	0.005	[5.93,30.42]
228	[5.93,30.42]	-27.112	1	[0. 1.]	[5.93,30.42]	0.023	[5.93,30.45]
229	[5.93,30.45]	-27.117	0	[1. 0.]	[5.93,30.45]	0.005	[5.94,30.45]
229	[5.93,30.45]	-27.117	1	[0. 1.]	[5.94,30.45]	0.023	[5.94,30.47]
230	[5.94,30.47]	-27.122	0	[1. 0.]	[5.94,30.47]	0.002	[5.94,30.47]
230	[5.94,30.47]	-27.122	1	[0. 1.]	[5.94,30.47]	0.008	[5.94,30.48]
231	[5.94,30.48]	-27.124	NaN	None	None	NaN	None

$$x^* = (5.94, 30.48)$$
$$f(x^*) = -27.124$$

Solution set 2:

- $x^{(0)}$ = [10, 35]
- ε = 0.01

Output of the solution set 2:

k	$x^{(k)}$	$f(x^{(k)})$	j	$d_j^{(k)}$	y_j	$\alpha_j^{(k)}$	y_{j+1}
0	[10,35]	50641.000	0.000	[1. 0.]	[10,35]	-3.166	[6.83,35.]
0	[10,35]	50641.000	1.000	[0. 1.]	[6.83,35.]	-0.041	[6.83,34.96]
1	[6.83,34.96]	-27.329	0.000	[1. 0.]	[6.83,34.96]	-0.008	[6.83,34.96]
1	[6.83,34.96]	-27.329	1.000	[0. 1.]	[6.83,34.96]	-0.038	[6.83,34.92]
2	[6.83,34.92]	-27.334	0.000	[1. 0.]	[6.83,34.92]	-0.008	[6.82,34.92]
2	[6.83,34.92]	-27.334	1.000	[0. 1.]	[6.82,34.92]	-0.038	[6.82,34.88]
3	[6.82,34.88]	-27.339	0.000	[1. 0.]	[6.82,34.88]	-0.008	[6.81,34.88]
3	[6.82,34.88]	-27.339	1.000	[0. 1.]	[6.81,34.88]	-0.038	[6.81,34.84]
4	[6.81,34.84]	-27.344	0.000	[1. 0.]	[6.81,34.84]	-0.008	[6.8 ,34.84]
4	[6.81,34.84]	-27.344	1.000	[0. 1.]	[6.8 ,34.84]	-0.038	[6.8 ,34.81]
...	...	...	...	...	...	...	...
69	[6.5 ,33.28]	-27.440	0.000	[1. 0.]	[6.5 ,33.28]	-0.005	[6.49,33.28]
69	[6.5 ,33.28]	-27.440	1.000	[0. 1.]	[6.49,33.28]	-0.023	[6.49,33.26]
70	[6.49,33.26]	-27.440	0.000	[1. 0.]	[6.49,33.26]	-0.005	[6.49,33.26]
70	[6.49,33.26]	-27.440	1.000	[0. 1.]	[6.49,33.26]	-0.023	[6.49,33.23]
71	[6.49,33.23]	-27.440	0.000	[1. 0.]	[6.49,33.23]	-0.005	[6.48,33.23]
71	[6.49,33.23]	-27.440	1.000	[0. 1.]	[6.48,33.23]	-0.023	[6.48,33.21]
72	[6.48,33.21]	-27.440	0.000	[1. 0.]	[6.48,33.21]	-0.005	[6.48,33.21]
72	[6.48,33.21]	-27.440	1.000	[0. 1.]	[6.48,33.21]	-0.023	[6.48,33.19]
73	[6.48,33.19]	-27.440	0.000	[1. 0.]	[6.48,33.19]	-0.002	[6.48,33.19]
73	[6.48,33.19]	-27.440	1.000	[0. 1.]	[6.48,33.19]	-0.008	[6.48,33.18]
74	[6.48,33.18]	-27.440	NaN	None	None	NaN	None

$$x^* = (6.48, 33.18)$$
$$f(x^*) = -27.440$$

***NOTE:** The complete tables of the outputs are available in the Appendix section.*

The conclusion:

The algorithm found two different points as the local optimum point however the function values, $f(x^*)$, for two different sets are very close to each other.

3 Hook & Jeeves Method

The Hook & Jeeves method is a modified version of the cyclic coordinate search. Since the cyclic coordinate method starts to cycle around a stationary point and makes so much steps to reach the point with a given precision, The Hook & Jeeves method avoids cycling and iterates much less steps to reach with a given precision.

The Hook & Jeeves method has two types of search; the first one is the exploratory moves which determine the promising direction by performing a single iteration of cyclic coordinate search and the second one is the pattern moves which determine how long the step length should be. These two searches use the exact line search that is mentioned above. And the method runs until the distance between two consecutive x^k's is less than a given ε value.

Source code of the algorithm:

```
def HookJeevesMethod(f, x0, epsilon):
    x0 = np.array(x0)
    x_array = [x0]
    x_temp = []
    k = 0
    n = len(x0)
    res_array = []
    while(True):
        # explorotary moves
        y0 = np.copy(x_array[k])
        for j in range(n):
            d_e = np.zeros(n)
            d_e[j] = 1
            alpha_e = ExactLineSearch(f, y0, d_e)
            y1 = y0 + alpha_e*d_e
            y0 = y1
        x_temp.append(y1)
        # pattern moves
        d_p = x_temp[k] - x_array[k]
        alpha_p = ExactLineSearch(f, x_array[k], d_p)
        y1 = x_array[k] + alpha_p*d_p
        x_array.append(y1)
        res_array.append([k, np_str(x_array[k]), f(x_array[k]), np_str(x_temp[k]), str(
            d_p), alpha_p, np_str(x_array[k+1])])
        k += 1
        if(np.linalg.norm(x_array[k]-x_array[k-1]) < epsilon):
            res_array.append([k, np_str(x_array[k]), f(x_array[k])])
            result_table = pd.DataFrame(res_array, columns=['k' ,'x^k', 'fx^k', 'x^temp',
                'd^k','a^k', 'x^k+1'])
            return result_table
```

Solution set 1:

- $x^{(0)} = [0, 0]$
- $\varepsilon = 0.01$

Output of the solution set 1:

k	$x^{(k)}$	$f(x^{(k)})$	x_{temp}	$d^{(k)}$	$\alpha^{(k)}$	$x^{(k+1)}$
0	[0,0]	16.000	[0.102,1.302]	[0.10223389 1.30157471]	1.225	[0.125,1.595]
1	[0.125,1.595]	13.330	[0.411,2.844]	[0.28533936 1.24969482]	12.340	[3.646,17.016]
2	[3.646,17.016]	-13.491	[3.288,17.231]	[-0.35858154 0.21514893]	0.902	[3.323,17.21]
3	[3.323,17.21]	-17.221	[3.324,17.413]	[0.00152588 0.20294189]	1.048	[3.325,17.423]
4	[3.325,17.423]	-17.357	[3.369,17.635]	[0.04425049 0.21209717]	67.143	[6.296,31.663]
5	[6.296,31.663]	-26.577	[6.175,31.668]	[-0.12054443 0.00457764]	0.972	[6.178,31.668]
6	[6.178,31.668]	-27.336	[6.177,31.675]	[-0.00152588 0.00762939]	0.627	[6.178,31.673]
7	[6.178,31.673]	-27.336	None	None	NaN	None

$$x^* = 6.178, 31.673$$
$$f(x^*) = -27.336$$

Solution set 2:

- $x^{(0)}$ = [10, 35]
- ε = 0.01

Output of the solution set 2:

k	$x^{(k)}$	$f(x^{(k)})$	x_{temp}	$d^{(k)}$	$\alpha^{(k)}$	$x^{(k+1)}$
0	[10,35]	50641.000	[6.834,34.959]	[-3.16619873 -0.04119873]	0.999	[6.836,34.959]
1	[6.836,34.959]	-27.327	[6.828,34.93]	[-0.00762939 -0.0289917]	7.597	[6.778,34.739]
2	[6.778,34.739]	-27.351	[6.782,34.703]	[0.00457764 -0.03509521]	0.880	[6.782,34.708]
3	[6.782,34.708]	-27.361	[6.777,34.676]	[-0.00457764 -0.03204346]	4.585	[6.761,34.561]
4	[6.761,34.561]	-27.368	[6.747,34.526]	[-0.01373291 -0.03509521]	1.808	[6.736,34.497]
5	[6.736,34.497]	-27.383	[6.734,34.462]	[-0.00152588 -0.03509521]	1.021	[6.734,34.461]
6	[6.734,34.461]	-27.386	[6.73 ,34.439]	[-0.00457764 -0.02288818]	51.164	[6.5 ,33.29]
7	[6.5 ,33.29]	-27.441	[6.495,33.268]	[-0.00457764 -0.02288818]	0.002	[6.5 ,33.29]
8	[6.5 ,33.29]	-27.441	None	None	NaN	None

$$x^* = 6.5, 33.29$$
$$f(x^*) = -27.441$$

The conclusion:

The method converges a zero-gradient point; in this case, a local minimum point due to the convexity of the given function as seen in the Figure 1, because the given function is differentiable. It finds two close points for two different sets, but the function values at these points are so much close to each other. It is expected to obtain the same point for different starting points if the precision ε is sufficiently small.

4 Simplex Search

Simplex Search is based on using the geometric structure of simplex to converge to optimality. The main idea of the search is replacing the worst point in a given simplex with a better one. In order to find a better point, reflection, expansion and contraction methods are used. The complete algorithm is given below.

```

x = np.zeros(shape=(3,2))
x[0] = np.array([-2,15])#initial
x[1] = np.array([-8,10])#initial
x[2] = np.array([0,0])#initial

def compute_f_values(a):
    f_values=np.zeros(a.shape[0])
    for i in range(a.shape[0]):
        f_values[i]=f(a[i])
    return f_values

def SimplexSearch(x, epsilon = 0.005, alpha = 1, beta = 0.5, gamma = 2, cond = True, _type = None):
    res = []
    while(cond):
        sum_value = 0
        f_values = compute_f_values(x)#function values of x_matrix
        x_h = x[np.argmax(f_values)] #the worst point
        x_l = x[np.argmin(f_values)] #the best point
        x_mean = np.mean(np.delete(x, np.argmax(f_values), 0), axis=0) #delete x_h and take the mean

        x_r = x_mean + alpha*(x_mean-x_h) #reflection

        if f(x_l) > f(x_r): #the reflected point x_r happens to be better than the current best
            x_e = x_mean + gamma*(x_r-x_mean) #Expansion

            if f(x_r) > f(x_e): #the expanded point x_e happens to be better than the current best x_r
                x[np.argmax(f_values)] = x_e
                _type = "E"

            else:           #the expanded point is not better than x_r so we replace x_h with x_r
                x[np.argmax(f_values)] = x_r
                _type = "R"
        else:
            if np.max(compute_f_values(np.delete(x, np.argmax(f_values), 0))) >= f(x_r):
                x[np.argmax(f_values)] = x_r
                _type = "R"
            else:
                _type = "C"
```

```
                if f(x_h) > f(x_r):
                    x_h_prime = x_r

                else:
                    x_h_prime = x_h
                x_c = x_mean + beta*(x_h_prime-x_mean) #contraction

                if f(x_c) <= f(x_h):
                    x[np.argmax(f_values)] = x_c
                else:
                    for i in range(x.shape[0]):
                        x[i] = x[i] + 0.5*(x_l-x[i]) #shrink operation

        for i in range(x.shape[0]):
            sum_value += (f(x[i]) - f(x_mean))**2
        cond = np.sqrt(sum_value) >= epsilon
        res.append([np_str(x_mean), np_str(x_h), np_str(x_l), np_str(x[np.argmax(f_values)]), f(x[np.argmax(f_values)]), _type])

    return pd.DataFrame(res, columns=["x_mean", "x_h", "x_l", "x_new", "f(x_new)", "type"])
```

Solution set 1:

- $x^{(0)}$ = [-2, 15]
- $x^{(1)}$ = [-8, 10]
- $x^{(2)}$ = [0, 0]
- $\varepsilon = 0.01$

Output of the solution set 1:

Iteration	$\bar{x}$	x_h	x_1	x_{new}	$f(x_{new})$	type
0	[-1. , 7.5]	[2.5 ,6.25]	[0.,0.]	[2.5 ,6.25]	1528.129	C
1	[1.25 ,3.125]	[-0.375, 9.062]	[0.,0.]	[-0.375, 9.062]	14310.216	C
2	[1.25 ,3.125]	[0.438,6.094]	[0.,0.]	[0.438,6.094]	235.522	C
3	[0.219,3.047]	[1.359,4.648]	[0.,0.]	[1.359,4.648]	25.778	C
4	[0.68 ,2.324]	[0.559,4.209]	[0.,0.]	[0.559,4.209]	10.239	C
5	[0.279,2.104]	[0.819,3.376]	[0.559,4.209]	[0.819,3.376]	7.729	C
6	[0.689,3.793]	[2.067,11.378]	[0.819,3.376]	[2.067,11.378]	-7.498	E
7	[1.443,7.377]	[2.328,10.546]	[2.067,11.378]	[2.328,10.546]	-5.232	R
8	[2.197,10.962]	[4.953,26.133]	[2.067,11.378]	[4.953,26.133]	-23.098	E
9	[3.51 ,18.755]	[2.919,14.651]	[4.953,26.133]	[2.919,14.651]	-13.538	C
10	[3.936,20.392]	[5.805,29.405]	[4.953,26.133]	[5.805,29.405]	-26.506	R
11	[5.379,27.769]	[6.609,34.328]	[5.805,29.405]	[6.609,34.328]	-26.100	C
12	[6.207,31.867]	[6.834,34.734]	[5.805,29.405]	[6.834,34.734]	-27.164	C
13	[6.32 ,32.069]	[6.464,33.199]	[6.834,34.734]	[6.464,33.199]	-27.411	C
14	[6.649,33.966]	[6.227,31.686]	[6.464,33.199]	[6.227,31.686]	-27.184	C
15	[6.346,32.442]	[6.59 ,33.588]	[6.464,33.199]	[6.59 ,33.588]	-27.352	C
16	[6.527,33.393]	[6.677,34.247]	[6.464,33.199]	[6.677,34.247]	-27.391	C
17	[6.571,33.723]	[6.58 ,33.656]	[6.464,33.199]	[6.58 ,33.656]	-27.428	C
18	[6.522,33.427]	[6.368,32.607]	[6.58 ,33.656]	[6.368,32.607]	-27.421	R
19	[6.474,33.131]	[6.469,33.165]	[6.58 ,33.656]	[6.469,33.165]	-27.437	C
20	[6.525,33.41]	[6.446,33.009]	[6.469,33.165]	[6.446,33.009]	-27.437	C
21	[6.458,33.087]	[6.519,33.371]	[6.469,33.165]	[6.519,33.371]	-27.439	C

$$x^* = (6.458, 33.087)$$
$$f(x^*) = -27.439$$

Solution set 2:

- $x^{(0)}$ = [-10, -10]
- $x^{(1)}$ = [-25, 45]
- $x^{(2)}$ = [20, -1]
- ε = 0.01

Output of the solution set 2:

Iteration	$\bar{x}$	x_h	x_1	x_{new}	$f(x_{new})$	type
0	[5. ,-5.5]	[-10. , 19.75]	[-10.,-10.]	[-10. , 19.75]	23668939.629	C
1	[-10. , 4.875]	[5. ,1.938]	[-10.,-10.]	[5. ,1.938]	282917.296	C
2	[-2.5 ,-4.031]	[1.25 ,-15.922]	[5. ,1.938]	[1.25 ,-15.922]	241708.391	C
3	[3.125,-6.992]	[-3.438,-8.496]	[1.25 ,-15.922]	[-3.438,-8.496]	5761.495	C
4	[-1.094,-12.209]	[-7.188,-26.355]	[-3.438,-8.496]	[-7.188,-26.355]	8571.987	R
5	[-5.312,-17.426]	[-2.031,-16.674]	[-3.438,-8.496]	[-2.031,-16.674]	1864.018	C
6	[-2.734,-12.585]	[1.719,1.186]	[-2.031,-16.674]	[1.719,1.186]	3023.403	R
7	[-0.156,-7.744]	[-1.797,-8.12]	[-2.031,-16.674]	[-1.797,-8.12]	41.418	C
8	[-1.914,-12.397]	[-5.547,-25.979]	[-1.797,-8.12]	[-5.547,-25.979]	124.851	R
9	[-3.672,-17.05]	[-2.852,-16.862]	[-1.797,-8.12]	[-2.852,-16.862]	112.389	C
10	[-2.324,-12.491]	[-3.936,-19.235]	[-1.797,-8.12]	[-3.936,-19.235]	81.804	C
11	[-2.866,-13.678]	[-2.859,-15.27]	[-1.797,-8.12]	[-2.859,-15.27]	64.194	C
12	[-2.328,-11.695]	[0.887,3.386]	[-1.797,-8.12]	[0.887,3.386]	8.577	E
13	[-0.455,-2.367]	[4.354,23.438]	[0.887,3.386]	[4.354,23.438]	-17.211	E
14	[2.621,13.412]	[7.038,34.944]	[4.354,23.438]	[7.038,34.944]	-25.464	R
15	[5.696,29.191]	[3.292,16.288]	[7.038,34.944]	[3.292,16.288]	-15.616	C
16	[5.696,29.191]	[8.1 ,42.093]	[7.038,34.944]	[8.1 ,42.093]	-20.432	R
17	[7.569,38.518]	[5.961,30.978]	[7.038,34.944]	[5.961,30.978]	-26.419	C
18	[6.5 ,32.961]	[5.699,28.395]	[5.961,30.978]	[5.699,28.395]	-25.404	C
19	[6.5 ,32.961]	[7.3 ,37.527]	[5.961,30.978]	[7.3 ,37.527]	-26.549	R
20	[6.631,34.253]	[6.834,34.598]	[7.3 ,37.527]	[6.834,34.598]	-26.959	C
21	[7.067,36.063]	[6.514,33.52]	[6.834,34.598]	[6.514,33.52]	-27.336	C
22	[6.674,34.059]	[6.361,32.325]	[6.514,33.52]	[6.361,32.325]	-27.195	C
23	[6.438,32.923]	[6.636,33.76]	[6.514,33.52]	[6.636,33.76]	-27.279	C
24	[6.575,33.64]	[6.468,32.983]	[6.514,33.52]	[6.468,32.983]	-27.363	C
25	[6.491,33.252]	[6.564,33.506]	[6.468,32.983]	[6.564,33.506]	-27.398	C
26	[6.516,33.244]	[6.515,33.382]	[6.564,33.506]	[6.515,33.382]	-27.440	C
27	[6.539,33.444]	[6.61 ,33.906]	[6.515,33.382]	[6.61 ,33.906]	-27.414	R
28	[6.563,33.644]	[6.563,33.575]	[6.515,33.382]	[6.563,33.575]	-27.432	C
29	[6.539,33.479]	[6.503,33.265]	[6.515,33.382]	[6.503,33.265]	-27.433	C
30	[6.509,33.324]	[6.455,33.073]	[6.515,33.382]	[6.455,33.073]	-27.439	R
31	[6.485,33.228]	[6.494,33.246]	[6.515,33.382]	[6.494,33.246]	-27.439	C

$$x^* = (6.485, 33.228)$$
$$f(x^*) = -27.439$$

The conclusion:
The algorithm found two different points as the local optimum point however the function values, $f(x^*)$, for two different sets are the same

5 Appendix

- The complete script file:

```
# %% [markdown]
# # Homework 3

# %%
import pandas as pd
import numpy as np
from sympy import Symbol, cos, sin, lambdify


# %%
x1  = Symbol('x1')
x2 = Symbol('x2')
function = (5*x1-x2)**4+((x1-2)**2)+(x1-2*x2)+12
f = lambdify([[x1,x2]], function, 'numpy')


# %%
from pylab import meshgrid,cm,imshow,contour,clabel,colorbar,axis,title,show
from mpl_toolkits.mplot3d import Axes3D
from matplotlib import cm
from matplotlib.ticker import LinearLocator, FormatStrFormatter
import matplotlib.pyplot as plt

# plot the function
x = np.arange(0,15,0.5)
y = np.arange(10,50,0.5)
X,Y = meshgrid(x, y) # grid of point
Z = f([X,Y]) # evaluation of the function on the grid

fig = plt.figure()
ax = fig.gca(projection='3d')
surf = ax.plot_surface(X, Y, Z, rstride=1, cstride=1,
                    cmap=cm.RdBu,linewidth=0, antialiased=False)

ax.zaxis.set_major_locator(LinearLocator(10))
ax.zaxis.set_major_formatter(FormatStrFormatter('%.02f'))

fig.colorbar(surf, shrink=0.5, aspect=5)
```

```python
plt.savefig("graph.png")
plt.show()

# %% [markdown]
# ## Exact Line Search

# %%
def BisectionMethod(f, a=-100,b=100,epsilon=0.005) :
    iteration=0
    while (b - a) >= epsilon:
        x_1 = (a + b) / 2
        fx_1 = f(x_1)
        if f(x_1 + epsilon) <= fx_1:
            a = x_1
        else:
            b = x_1
        iteration+=1
    x_star = (a+b)/2
    return x_star


# %%
def ExactLineSearch(f, x0, d):
    alpha = Symbol('alpha')
    function_alpha = f(np.array(x0)+alpha*np.array(d))
    f_alp = lambdify(alpha, function_alpha, 'numpy')
    alp_star = BisectionMethod(f_alp)
    return alp_star

# %% [markdown]
# ## Cyclic Coordinate Search

# %%
def np_str(x_k):
    '''
    Used to convert numpy array to string with determined format
    '''
    return np.array2string(x_k, precision=3, separator=',')


# %%
def CyclicCoordinateSearch(f, x0, epsilon):
    x0 = np.array(x0)
    x_array = [x0]
    k = 0
    n = len(x0)
    res_array = []
    while(True):
        y0 = np.copy(x_array[k])
        for j in range(n):
            d = np.zeros(n)
            d[j] = 1
            alpha = ExactLineSearch(f, y0, d)
```

```
            y1 = y0 + alpha*d
            res_array.append([k, np_str(x_array[k]), f(x_array[k]),j, str(d),np_str(y0),
                alpha, np_str(y1)])
            y0 = y1
        x_array.append(y1)
        k += 1
        if(np.linalg.norm(x_array[k]-x_array[k-1]) < epsilon):
            res_array.append([k, np_str(x_array[k]), f(x_array[k])])
            result_table = pd.DataFrame(res_array, columns=['k' ,'x^k', 'fx^k', 'j','d^j'
                ,'y^j','a^j', 'y^j+1'])
            return result_table

# %% [markdown]
# **Solution set 1:**
# *   x^0 = \[0, 0\]
# *   Epsilon = 0.01

# %%
output1 = CyclicCoordinateSearch(f,[0,0],0.01)
output1

# %% [markdown]
# **Solution set 2:**
# *   x^0 = \[10, 35\]
# *   Epsilon = 0.01

# %%
output2 = CyclicCoordinateSearch(f,[10,35],0.01)
output2


# %%
print(output2[-11:].to_latex(index=False,float_format='%.3f'))

# %% [markdown]
# ## Hook & Jeeves Method

# %%
def HookJeevesMethod(f, x0, epsilon):
    x0 = np.array(x0)
    x_array = [x0]
    x_temp = []
    k = 0
    n = len(x0)
    res_array = []
    while(True):
        # explorotary moves
        y0 = np.copy(x_array[k])
        for j in range(n):
            d_e = np.zeros(n)
            d_e[j] = 1
            alpha_e = ExactLineSearch(f, y0, d_e)
            y1 = y0 + alpha_e*d_e
            y0 = y1
```

```
        x_temp.append(y1)
        # pattern moves
        d_p = x_temp[k] - x_array[k]
        alpha_p = ExactLineSearch(f, x_array[k], d_p)
        y1 = x_array[k] + alpha_p*d_p
        x_array.append(y1)
        res_array.append([k, np_str(x_array[k]), f(x_array[k]), np_str(x_temp[k]), str(
            d_p), alpha_p, np_str(x_array[k+1])])
        k += 1
        if(np.linalg.norm(x_array[k]-x_array[k-1]) < epsilon):
            res_array.append([k, np_str(x_array[k]), f(x_array[k])])
            result_table = pd.DataFrame(res_array, columns=['k' ,'x^k', 'fx^k', 'x^temp',
                'd^k','a^k', 'x^k+1'])
            return result_table


# %%
HJMoutput1 = HookJeevesMethod(f,[0,0],0.01)
HJMoutput1

# %%
HJMoutput2 = HookJeevesMethod(f,[10,35],0.01)
HJMoutput2

# %% [markdown]
# ## Simplex Search

# %%

def compute_f_values(a):
    f_values=np.zeros(a.shape[0])
    for i in range(a.shape[0]):
        f_values[i]=f(a[i])
    return f_values

def SimplexSearch(x, epsilon = 0.005, alpha = 1, beta = 0.5, gamma = 2, cond = True,
    _type = None):
    res = []
    while(cond):
        sum_value = 0
        f_values = compute_f_values(x)#function values of x_matrix
        x_h = x[np.argmax(f_values)] #the worst point
        x_l = x[np.argmin(f_values)] #the best point
        x_mean = np.mean(np.delete(x, np.argmax(f_values), 0), axis=0) #delete x_h and
            take the mean

        x_r = x_mean + alpha*(x_mean-x_h) #reflection

        if f(x_l) > f(x_r): #the reflected point x_r happens to be better than the
            current best
            x_e = x_mean + gamma*(x_r-x_mean) #Expansion

            if f(x_r) > f(x_e): #the expanded point x_e happens to be better than the
                current best x_r
```

```
                x[np.argmax(f_values)] = x_e
                _type = "E"

            else:           #the expanded point is not better than x_r so we replace x_h
                 with x_r
                x[np.argmax(f_values)] = x_r
                _type = "R"
        else:
            if np.max(compute_f_values(np.delete(x, np.argmax(f_values), 0))) >= f(x_r):
                x[np.argmax(f_values)] = x_r
                _type = "R"
            else:
                _type = "C"
                if f(x_h) > f(x_r):
                    x_h_prime = x_r

                else:
                    x_h_prime = x_h
                x_c = x_mean + beta*(x_h_prime-x_mean) #contraction

                if f(x_c) <= f(x_h):
                    x[np.argmax(f_values)] = x_c
                else:
                    for i in range(x.shape[0]):
                        x[i] = x[i] + 0.5*(x_l-x[i]) #shrink operation

        for i in range(x.shape[0]):
            sum_value += (f(x[i]) - f(x_mean))**2
        cond = np.sqrt(sum_value) >= epsilon
        res.append([np_str(x_mean), np_str(x_h), np_str(x_l), np_str(x[np.argmax(
            f_values)]), f(x[np.argmax(f_values)]), _type])

    return pd.DataFrame(res, columns=["x_mean", "x_h", "x_l", "x_new", "f(x_new)", "type
        "])


# %%
x = np.zeros(shape=(3,2))
x[0] = np.array([-2,15])#initial
x[1] = np.array([-8,10])#initial
x[2] = np.array([0,0])#initial

simplex_result = SimplexSearch(x.copy())
simplex_result

# %%
x_2 = np.zeros(shape=(3,2))
x_2[0] = np.array([-10,-10])#initial
x_2[1] = np.array([-25,45])#initial
x_2[2] = np.array([20,-1])#initial
simplex_result_2 = SimplexSearch(x_2.copy())
simplex_result_2
```

- The complete output table of the cyclic coordinate search set 1:

k	$x^{(k)}$	$f(x^{(k)})$	j	$d_j^{(k)}$	y_j	$\alpha_j^{(k)}$	y_{j+1}
0	[0,0]	16.000	0.0	[1. 0.]	[0,0]	0.102	[0.102,0.]
0	[0,0]	16.000	1.0	[0. 1.]	[0.102,0.]	1.302	[0.102,1.302]
1	[0.102,1.302]	13.491	0.0	[1. 0.]	[0.102,1.302]	0.252	[0.354,1.302]
1	[0.102,1.302]	13.491	1.0	[0. 1.]	[0.354,1.302]	1.259	[0.354,2.56]
2	[0.354,2.56]	10.333	0.0	[1. 0.]	[0.354,2.56]	0.246	[0.6 ,2.56]
2	[0.354,2.56]	10.333	1.0	[0. 1.]	[0.6 ,2.56]	1.228	[0.6 ,3.789]
3	[0.6 ,3.789]	7.373	0.0	[1. 0.]	[0.6 ,3.789]	0.237	[0.836,3.789]
3	[0.6 ,3.789]	7.373	1.0	[0. 1.]	[0.836,3.789]	1.183	[0.836,4.971]
4	[0.836,4.971]	4.638	0.0	[1. 0.]	[0.836,4.971]	0.227	[1.064,4.971]
4	[0.836,4.971]	4.638	1.0	[0. 1.]	[1.064,4.971]	1.137	[1.064,6.108]
5	[1.064,6.108]	2.115	0.0	[1. 0.]	[1.064,6.108]	0.212	[1.276,6.108]
5	[1.064,6.108]	2.115	1.0	[0. 1.]	[1.276,6.108]	1.060	[1.276,7.169]
6	[1.276,7.169]	-0.147	0.0	[1. 0.]	[1.276,7.169]	0.185	[1.46 ,7.169]
6	[1.276,7.169]	-0.147	1.0	[0. 1.]	[1.46 ,7.169]	0.923	[1.46 ,8.092]
7	[1.46 ,8.092]	-2.042	0.0	[1. 0.]	[1.46 ,8.092]	0.114	[1.575,8.092]
7	[1.46 ,8.092]	-2.042	1.0	[0. 1.]	[1.575,8.092]	0.572	[1.575,8.664]
8	[1.575,8.664]	-3.182	0.0	[1. 0.]	[1.575,8.664]	0.102	[1.677,8.664]
8	[1.575,8.664]	-3.182	1.0	[0. 1.]	[1.677,8.664]	0.511	[1.677,9.175]
9	[1.677,9.175]	-4.179	0.0	[1. 0.]	[1.677,9.175]	0.096	[1.773,9.175]
9	[1.677,9.175]	-4.179	1.0	[0. 1.]	[1.773,9.175]	0.481	[1.773,9.656]
10	[1.773,9.656]	-5.097	0.0	[1. 0.]	[1.773,9.656]	0.090	[1.863,9.656]
10	[1.773,9.656]	-5.097	1.0	[0. 1.]	[1.863,9.656]	0.450	[1.863,10.106]
11	[1.863,10.106]	-5.940	0.0	[1. 0.]	[1.863,10.106]	0.084	[1.947,10.106]
11	[1.863,10.106]	-5.940	1.0	[0. 1.]	[1.947,10.106]	0.420	[1.947,10.526]
12	[1.947,10.526]	-6.711	0.0	[1. 0.]	[1.947,10.526]	0.081	[2.028,10.526]
12	[1.947,10.526]	-6.711	1.0	[0. 1.]	[2.028,10.526]	0.404	[2.028,10.93]
13	[2.028,10.93]	-7.441	0.0	[1. 0.]	[2.028,10.93]	0.078	[2.106,10.93]
13	[2.028,10.93]	-7.441	1.0	[0. 1.]	[2.106,10.93]	0.389	[2.106,11.319]
14	[2.106,11.319]	-8.131	0.0	[1. 0.]	[2.106,11.319]	0.075	[2.18 ,11.319]
14	[2.106,11.319]	-8.131	1.0	[0. 1.]	[2.18 ,11.319]	0.374	[2.18 ,11.693]
15	[2.18 ,11.693]	-8.782	0.0	[1. 0.]	[2.18 ,11.693]	0.072	[2.252,11.693]
15	[2.18 ,11.693]	-8.782	1.0	[0. 1.]	[2.252,11.693]	0.359	[2.252,12.051]
16	[2.252,12.051]	-9.397	0.0	[1. 0.]	[2.252,12.051]	0.069	[2.321,12.051]

16	[2.252,12.051]	-9.397	1.0	[0. 1.]	[2.321,12.051]	0.343	[2.321,12.395]
17	[2.321,12.395]	-9.975	0.0	[1. 0.]	[2.321,12.395]	0.066	[2.386,12.395]
17	[2.321,12.395]	-9.975	1.0	[0. 1.]	[2.386,12.395]	0.328	[2.386,12.723]
18	[2.386,12.723]	-10.519	0.0	[1. 0.]	[2.386,12.723]	0.066	[2.452,12.723]
18	[2.386,12.723]	-10.519	1.0	[0. 1.]	[2.452,12.723]	0.328	[2.452,13.051]
19	[2.452,13.051]	-11.055	0.0	[1. 0.]	[2.452,13.051]	0.063	[2.515,13.051]
19	[2.452,13.051]	-11.055	1.0	[0. 1.]	[2.515,13.051]	0.313	[2.515,13.364]
20	[2.515,13.364]	-11.557	0.0	[1. 0.]	[2.515,13.364]	0.060	[2.574,13.364]
20	[2.515,13.364]	-11.557	1.0	[0. 1.]	[2.574,13.364]	0.298	[2.574,13.661]
21	[2.574,13.661]	-12.028	0.0	[1. 0.]	[2.574,13.661]	0.060	[2.634,13.661]
21	[2.574,13.661]	-12.028	1.0	[0. 1.]	[2.634,13.661]	0.298	[2.634,13.959]
22	[2.634,13.959]	-12.492	0.0	[1. 0.]	[2.634,13.959]	0.056	[2.69 ,13.959]
22	[2.634,13.959]	-12.492	1.0	[0. 1.]	[2.69 ,13.959]	0.282	[2.69 ,14.241]
23	[2.69 ,14.241]	-12.925	0.0	[1. 0.]	[2.69 ,14.241]	0.056	[2.747,14.241]
23	[2.69 ,14.241]	-12.925	1.0	[0. 1.]	[2.747,14.241]	0.282	[2.747,14.523]
24	[2.747,14.523]	-13.352	0.0	[1. 0.]	[2.747,14.523]	0.053	[2.8 ,14.523]
24	[2.747,14.523]	-13.352	1.0	[0. 1.]	[2.8 ,14.523]	0.267	[2.8 ,14.79]
25	[2.8 ,14.79]	-13.750	0.0	[1. 0.]	[2.8 ,14.79]	0.053	[2.853,14.79]
25	[2.8 ,14.79]	-13.750	1.0	[0. 1.]	[2.853,14.79]	0.267	[2.853,15.057]
26	[2.853,15.057]	-14.143	0.0	[1. 0.]	[2.853,15.057]	0.050	[2.904,15.057]
26	[2.853,15.057]	-14.143	1.0	[0. 1.]	[2.904,15.057]	0.252	[2.904,15.309]
27	[2.904,15.309]	-14.507	0.0	[1. 0.]	[2.904,15.309]	0.050	[2.954,15.309]
27	[2.904,15.309]	-14.507	1.0	[0. 1.]	[2.954,15.309]	0.252	[2.954,15.561]
28	[2.954,15.561]	-14.867	0.0	[1. 0.]	[2.954,15.561]	0.050	[3.004,15.561]
28	[2.954,15.561]	-14.867	1.0	[0. 1.]	[3.004,15.561]	0.252	[3.004,15.813]
29	[3.004,15.813]	-15.222	0.0	[1. 0.]	[3.004,15.813]	0.047	[3.052,15.813]
29	[3.004,15.813]	-15.222	1.0	[0. 1.]	[3.052,15.813]	0.237	[3.052,16.049]
30	[3.052,16.049]	-15.550	0.0	[1. 0.]	[3.052,16.049]	0.047	[3.099,16.049]
30	[3.052,16.049]	-15.550	1.0	[0. 1.]	[3.099,16.049]	0.237	[3.099,16.286]
31	[3.099,16.286]	-15.874	0.0	[1. 0.]	[3.099,16.286]	0.047	[3.146,16.286]
31	[3.099,16.286]	-15.874	1.0	[0. 1.]	[3.146,16.286]	0.237	[3.146,16.522]
32	[3.146,16.522]	-16.194	0.0	[1. 0.]	[3.146,16.522]	0.044	[3.191,16.522]
32	[3.146,16.522]	-16.194	1.0	[0. 1.]	[3.191,16.522]	0.221	[3.191,16.743]
33	[3.191,16.743]	-16.488	0.0	[1. 0.]	[3.191,16.743]	0.044	[3.235,16.743]
33	[3.191,16.743]	-16.488	1.0	[0. 1.]	[3.235,16.743]	0.221	[3.235,16.965]
34	[3.235,16.965]	-16.779	0.0	[1. 0.]	[3.235,16.965]	0.044	[3.279,16.965]

34	[3.235,16.965]	-16.779	1.0	[0. 1.]	[3.279,16.965]	0.221	[3.279,17.186]
35	[3.279,17.186]	-17.066	0.0	[1. 0.]	[3.279,17.186]	0.041	[3.32 ,17.186]
35	[3.279,17.186]	-17.066	1.0	[0. 1.]	[3.32 ,17.186]	0.206	[3.32 ,17.392]
36	[3.32 ,17.392]	-17.330	0.0	[1. 0.]	[3.32 ,17.392]	0.041	[3.362,17.392]
36	[3.32 ,17.392]	-17.330	1.0	[0. 1.]	[3.362,17.392]	0.206	[3.362,17.598]
37	[3.362,17.598]	-17.590	0.0	[1. 0.]	[3.362,17.598]	0.041	[3.403,17.598]
37	[3.362,17.598]	-17.590	1.0	[0. 1.]	[3.403,17.598]	0.206	[3.403,17.804]
38	[3.403,17.804]	-17.847	0.0	[1. 0.]	[3.403,17.804]	0.041	[3.444,17.804]
38	[3.403,17.804]	-17.847	1.0	[0. 1.]	[3.444,17.804]	0.206	[3.444,18.01]
39	[3.444,18.01]	-18.101	0.0	[1. 0.]	[3.444,18.01]	0.038	[3.482,18.01]
39	[3.444,18.01]	-18.101	1.0	[0. 1.]	[3.482,18.01]	0.191	[3.482,18.201]
40	[3.482,18.201]	-18.333	0.0	[1. 0.]	[3.482,18.201]	0.038	[3.52 ,18.201]
40	[3.482,18.201]	-18.333	1.0	[0. 1.]	[3.52 ,18.201]	0.191	[3.52 ,18.391]
41	[3.52 ,18.391]	-18.561	0.0	[1. 0.]	[3.52 ,18.391]	0.038	[3.558,18.391]
41	[3.52 ,18.391]	-18.561	1.0	[0. 1.]	[3.558,18.391]	0.191	[3.558,18.582]
42	[3.558,18.582]	-18.787	0.0	[1. 0.]	[3.558,18.582]	0.038	[3.596,18.582]
42	[3.558,18.582]	-18.787	1.0	[0. 1.]	[3.596,18.582]	0.191	[3.596,18.773]
43	[3.596,18.773]	-19.010	0.0	[1. 0.]	[3.596,18.773]	0.035	[3.632,18.773]
43	[3.596,18.773]	-19.010	1.0	[0. 1.]	[3.632,18.773]	0.175	[3.632,18.948]
44	[3.632,18.948]	-19.213	0.0	[1. 0.]	[3.632,18.948]	0.035	[3.667,18.948]
44	[3.632,18.948]	-19.213	1.0	[0. 1.]	[3.667,18.948]	0.175	[3.667,19.124]
45	[3.667,19.124]	-19.413	0.0	[1. 0.]	[3.667,19.124]	0.035	[3.702,19.124]
45	[3.667,19.124]	-19.413	1.0	[0. 1.]	[3.702,19.124]	0.175	[3.702,19.299]
46	[3.702,19.299]	-19.610	0.0	[1. 0.]	[3.702,19.299]	0.035	[3.737,19.299]
46	[3.702,19.299]	-19.610	1.0	[0. 1.]	[3.737,19.299]	0.175	[3.737,19.475]
47	[3.737,19.475]	-19.806	0.0	[1. 0.]	[3.737,19.475]	0.032	[3.769,19.475]
47	[3.737,19.475]	-19.806	1.0	[0. 1.]	[3.769,19.475]	0.160	[3.769,19.635]
48	[3.769,19.635]	-19.982	0.0	[1. 0.]	[3.769,19.635]	0.032	[3.801,19.635]
48	[3.769,19.635]	-19.982	1.0	[0. 1.]	[3.801,19.635]	0.160	[3.801,19.795]
49	[3.801,19.795]	-20.156	0.0	[1. 0.]	[3.801,19.795]	0.032	[3.833,19.795]
49	[3.801,19.795]	-20.156	1.0	[0. 1.]	[3.833,19.795]	0.160	[3.833,19.955]
50	[3.833,19.955]	-20.328	0.0	[1. 0.]	[3.833,19.955]	0.032	[3.865,19.955]
50	[3.833,19.955]	-20.328	1.0	[0. 1.]	[3.865,19.955]	0.160	[3.865,20.116]
51	[3.865,20.116]	-20.498	0.0	[1. 0.]	[3.865,20.116]	0.032	[3.897,20.116]
51	[3.865,20.116]	-20.498	1.0	[0. 1.]	[3.897,20.116]	0.160	[3.897,20.276]
52	[3.897,20.276]	-20.665	0.0	[1. 0.]	[3.897,20.276]	0.032	[3.929,20.276]

52	[3.897,20.276]	-20.665	1.0	[0. 1.]	[3.929,20.276]	0.160	[3.929,20.436]
53	[3.929,20.436]	-20.831	0.0	[1. 0.]	[3.929,20.436]	0.029	[3.958,20.436]
53	[3.929,20.436]	-20.831	1.0	[0. 1.]	[3.958,20.436]	0.145	[3.958,20.581]
54	[3.958,20.581]	-20.979	0.0	[1. 0.]	[3.958,20.581]	0.029	[3.987,20.581]
54	[3.958,20.581]	-20.979	1.0	[0. 1.]	[3.987,20.581]	0.145	[3.987,20.726]
55	[3.987,20.726]	-21.126	0.0	[1. 0.]	[3.987,20.726]	0.029	[4.016,20.726]
55	[3.987,20.726]	-21.126	1.0	[0. 1.]	[4.016,20.726]	0.145	[4.016,20.871]
56	[4.016,20.871]	-21.271	0.0	[1. 0.]	[4.016,20.871]	0.029	[4.045,20.871]
56	[4.016,20.871]	-21.271	1.0	[0. 1.]	[4.045,20.871]	0.145	[4.045,21.016]
57	[4.045,21.016]	-21.414	0.0	[1. 0.]	[4.045,21.016]	0.029	[4.074,21.016]
57	[4.045,21.016]	-21.414	1.0	[0. 1.]	[4.074,21.016]	0.145	[4.074,21.161]
58	[4.074,21.161]	-21.556	0.0	[1. 0.]	[4.074,21.161]	0.029	[4.103,21.161]
58	[4.074,21.161]	-21.556	1.0	[0. 1.]	[4.103,21.161]	0.145	[4.103,21.306]
59	[4.103,21.306]	-21.695	0.0	[1. 0.]	[4.103,21.306]	0.026	[4.129,21.306]
59	[4.103,21.306]	-21.695	1.0	[0. 1.]	[4.129,21.306]	0.130	[4.129,21.436]
60	[4.129,21.436]	-21.819	0.0	[1. 0.]	[4.129,21.436]	0.026	[4.155,21.436]
60	[4.129,21.436]	-21.819	1.0	[0. 1.]	[4.155,21.436]	0.130	[4.155,21.565]
61	[4.155,21.565]	-21.941	0.0	[1. 0.]	[4.155,21.565]	0.026	[4.181,21.565]
61	[4.155,21.565]	-21.941	1.0	[0. 1.]	[4.181,21.565]	0.130	[4.181,21.695]
62	[4.181,21.695]	-22.062	0.0	[1. 0.]	[4.181,21.695]	0.026	[4.207,21.695]
62	[4.181,21.695]	-22.062	1.0	[0. 1.]	[4.207,21.695]	0.130	[4.207,21.825]
63	[4.207,21.825]	-22.182	0.0	[1. 0.]	[4.207,21.825]	0.026	[4.233,21.825]
63	[4.207,21.825]	-22.182	1.0	[0. 1.]	[4.233,21.825]	0.130	[4.233,21.954]
64	[4.233,21.954]	-22.300	0.0	[1. 0.]	[4.233,21.954]	0.026	[4.259,21.954]
64	[4.233,21.954]	-22.300	1.0	[0. 1.]	[4.259,21.954]	0.130	[4.259,22.084]
65	[4.259,22.084]	-22.417	0.0	[1. 0.]	[4.259,22.084]	0.026	[4.285,22.084]
65	[4.259,22.084]	-22.417	1.0	[0. 1.]	[4.285,22.084]	0.130	[4.285,22.214]
66	[4.285,22.214]	-22.533	0.0	[1. 0.]	[4.285,22.214]	0.026	[4.311,22.214]
66	[4.285,22.214]	-22.533	1.0	[0. 1.]	[4.311,22.214]	0.130	[4.311,22.343]
67	[4.311,22.343]	-22.647	0.0	[1. 0.]	[4.311,22.343]	0.023	[4.333,22.343]
67	[4.311,22.343]	-22.647	1.0	[0. 1.]	[4.333,22.343]	0.114	[4.333,22.458]
68	[4.333,22.458]	-22.747	0.0	[1. 0.]	[4.333,22.458]	0.023	[4.356,22.458]
68	[4.333,22.458]	-22.747	1.0	[0. 1.]	[4.356,22.458]	0.114	[4.356,22.572]
69	[4.356,22.572]	-22.845	0.0	[1. 0.]	[4.356,22.572]	0.023	[4.379,22.572]
69	[4.356,22.572]	-22.845	1.0	[0. 1.]	[4.379,22.572]	0.114	[4.379,22.687]
70	[4.379,22.687]	-22.943	0.0	[1. 0.]	[4.379,22.687]	0.023	[4.402,22.687]

70	[4.379,22.687]	-22.943	1.0	[0. 1.]	[4.402,22.687]	0.114	[4.402,22.801]
71	[4.402,22.801]	-23.040	0.0	[1. 0.]	[4.402,22.801]	0.023	[4.425,22.801]
71	[4.402,22.801]	-23.040	1.0	[0. 1.]	[4.425,22.801]	0.114	[4.425,22.916]
72	[4.425,22.916]	-23.135	0.0	[1. 0.]	[4.425,22.916]	0.023	[4.448,22.916]
72	[4.425,22.916]	-23.135	1.0	[0. 1.]	[4.448,22.916]	0.114	[4.448,23.03]
73	[4.448,23.03]	-23.230	0.0	[1. 0.]	[4.448,23.03]	0.023	[4.471,23.03]
73	[4.448,23.03]	-23.230	1.0	[0. 1.]	[4.471,23.03]	0.114	[4.471,23.145]
74	[4.471,23.145]	-23.323	0.0	[1. 0.]	[4.471,23.145]	0.023	[4.494,23.145]
74	[4.471,23.145]	-23.323	1.0	[0. 1.]	[4.494,23.145]	0.114	[4.494,23.259]
75	[4.494,23.259]	-23.415	0.0	[1. 0.]	[4.494,23.259]	0.023	[4.517,23.259]
75	[4.494,23.259]	-23.415	1.0	[0. 1.]	[4.517,23.259]	0.114	[4.517,23.373]
76	[4.517,23.373]	-23.507	0.0	[1. 0.]	[4.517,23.373]	0.020	[4.536,23.373]
76	[4.517,23.373]	-23.507	1.0	[0. 1.]	[4.536,23.373]	0.099	[4.536,23.473]
77	[4.536,23.473]	-23.585	0.0	[1. 0.]	[4.536,23.473]	0.020	[4.556,23.473]
77	[4.536,23.473]	-23.585	1.0	[0. 1.]	[4.556,23.473]	0.099	[4.556,23.572]
78	[4.556,23.572]	-23.662	0.0	[1. 0.]	[4.556,23.572]	0.020	[4.576,23.572]
78	[4.556,23.572]	-23.662	1.0	[0. 1.]	[4.576,23.572]	0.099	[4.576,23.671]
79	[4.576,23.671]	-23.739	0.0	[1. 0.]	[4.576,23.671]	0.020	[4.596,23.671]
79	[4.576,23.671]	-23.739	1.0	[0. 1.]	[4.596,23.671]	0.099	[4.596,23.77]
80	[4.596,23.77]	-23.815	0.0	[1. 0.]	[4.596,23.77]	0.020	[4.616,23.77]
80	[4.596,23.77]	-23.815	1.0	[0. 1.]	[4.616,23.77]	0.099	[4.616,23.869]
81	[4.616,23.869]	-23.890	0.0	[1. 0.]	[4.616,23.869]	0.020	[4.636,23.869]
81	[4.616,23.869]	-23.890	1.0	[0. 1.]	[4.636,23.869]	0.099	[4.636,23.969]
82	[4.636,23.969]	-23.965	0.0	[1. 0.]	[4.636,23.969]	0.020	[4.655,23.969]
82	[4.636,23.969]	-23.965	1.0	[0. 1.]	[4.655,23.969]	0.099	[4.655,24.068]
83	[4.655,24.068]	-24.038	0.0	[1. 0.]	[4.655,24.068]	0.020	[4.675,24.068]
83	[4.655,24.068]	-24.038	1.0	[0. 1.]	[4.675,24.068]	0.099	[4.675,24.167]
84	[4.675,24.167]	-24.111	0.0	[1. 0.]	[4.675,24.167]	0.020	[4.695,24.167]
84	[4.675,24.167]	-24.111	1.0	[0. 1.]	[4.695,24.167]	0.099	[4.695,24.266]
85	[4.695,24.266]	-24.183	0.0	[1. 0.]	[4.695,24.266]	0.020	[4.715,24.266]
85	[4.695,24.266]	-24.183	1.0	[0. 1.]	[4.715,24.266]	0.099	[4.715,24.365]
86	[4.715,24.365]	-24.254	0.0	[1. 0.]	[4.715,24.365]	0.017	[4.732,24.365]
86	[4.715,24.365]	-24.254	1.0	[0. 1.]	[4.732,24.365]	0.084	[4.732,24.449]
87	[4.732,24.449]	-24.314	0.0	[1. 0.]	[4.732,24.449]	0.017	[4.749,24.449]
87	[4.732,24.449]	-24.314	1.0	[0. 1.]	[4.749,24.449]	0.084	[4.749,24.533]
88	[4.749,24.533]	-24.373	0.0	[1. 0.]	[4.749,24.533]	0.017	[4.765,24.533]

88	[4.749,24.533]	-24.373	1.0	[0. 1.]	[4.765,24.533]	0.084	[4.765,24.617]
89	[4.765,24.617]	-24.431	0.0	[1. 0.]	[4.765,24.617]	0.017	[4.782,24.617]
89	[4.765,24.617]	-24.431	1.0	[0. 1.]	[4.782,24.617]	0.084	[4.782,24.701]
90	[4.782,24.701]	-24.489	0.0	[1. 0.]	[4.782,24.701]	0.017	[4.799,24.701]
90	[4.782,24.701]	-24.489	1.0	[0. 1.]	[4.799,24.701]	0.084	[4.799,24.785]
91	[4.799,24.785]	-24.547	0.0	[1. 0.]	[4.799,24.785]	0.017	[4.816,24.785]
91	[4.799,24.785]	-24.547	1.0	[0. 1.]	[4.816,24.785]	0.084	[4.816,24.869]
92	[4.816,24.869]	-24.604	0.0	[1. 0.]	[4.816,24.869]	0.017	[4.832,24.869]
92	[4.816,24.869]	-24.604	1.0	[0. 1.]	[4.832,24.869]	0.084	[4.832,24.953]
93	[4.832,24.953]	-24.660	0.0	[1. 0.]	[4.832,24.953]	0.017	[4.849,24.953]
93	[4.832,24.953]	-24.660	1.0	[0. 1.]	[4.849,24.953]	0.084	[4.849,25.037]
94	[4.849,25.037]	-24.716	0.0	[1. 0.]	[4.849,25.037]	0.017	[4.866,25.037]
94	[4.849,25.037]	-24.716	1.0	[0. 1.]	[4.866,25.037]	0.084	[4.866,25.121]
95	[4.866,25.121]	-24.771	0.0	[1. 0.]	[4.866,25.121]	0.017	[4.883,25.121]
95	[4.866,25.121]	-24.771	1.0	[0. 1.]	[4.883,25.121]	0.084	[4.883,25.204]
96	[4.883,25.204]	-24.825	0.0	[1. 0.]	[4.883,25.204]	0.017	[4.9 ,25.204]
96	[4.883,25.204]	-24.825	1.0	[0. 1.]	[4.9 ,25.204]	0.084	[4.9 ,25.288]
97	[4.9 ,25.288]	-24.879	0.0	[1. 0.]	[4.9 ,25.288]	0.017	[4.916,25.288]
97	[4.9 ,25.288]	-24.879	1.0	[0. 1.]	[4.916,25.288]	0.084	[4.916,25.372]
98	[4.916,25.372]	-24.933	0.0	[1. 0.]	[4.916,25.372]	0.017	[4.933,25.372]
98	[4.916,25.372]	-24.933	1.0	[0. 1.]	[4.933,25.372]	0.084	[4.933,25.456]
99	[4.933,25.456]	-24.986	0.0	[1. 0.]	[4.933,25.456]	0.017	[4.95 ,25.456]
99	[4.933,25.456]	-24.986	1.0	[0. 1.]	[4.95 ,25.456]	0.084	[4.95,25.54]
100	[4.95,25.54]	-25.038	0.0	[1. 0.]	[4.95,25.54]	0.014	[4.964,25.54]
100	[4.95,25.54]	-25.038	1.0	[0. 1.]	[4.964,25.54]	0.069	[4.964,25.609]
101	[4.964,25.609]	-25.080	0.0	[1. 0.]	[4.964,25.609]	0.014	[4.977,25.609]
101	[4.964,25.609]	-25.080	1.0	[0. 1.]	[4.977,25.609]	0.069	[4.977,25.677]
102	[4.977,25.677]	-25.122	0.0	[1. 0.]	[4.977,25.677]	0.014	[4.991,25.677]
102	[4.977,25.677]	-25.122	1.0	[0. 1.]	[4.991,25.677]	0.069	[4.991,25.746]
103	[4.991,25.746]	-25.164	0.0	[1. 0.]	[4.991,25.746]	0.014	[5.005,25.746]
103	[4.991,25.746]	-25.164	1.0	[0. 1.]	[5.005,25.746]	0.069	[5.005,25.815]
104	[5.005,25.815]	-25.205	0.0	[1. 0.]	[5.005,25.815]	0.014	[5.019,25.815]
104	[5.005,25.815]	-25.205	1.0	[0. 1.]	[5.019,25.815]	0.069	[5.019,25.883]
105	[5.019,25.883]	-25.246	0.0	[1. 0.]	[5.019,25.883]	0.014	[5.032,25.883]
105	[5.019,25.883]	-25.246	1.0	[0. 1.]	[5.032,25.883]	0.069	[5.032,25.952]
106	[5.032,25.952]	-25.287	0.0	[1. 0.]	[5.032,25.952]	0.014	[5.046,25.952]

106	[5.032,25.952]	-25.287	1.0	[0. 1.]	[5.046,25.952]	0.069	[5.046,26.021]
107	[5.046,26.021]	-25.327	0.0	[1. 0.]	[5.046,26.021]	0.014	[5.06 ,26.021]
107	[5.046,26.021]	-25.327	1.0	[0. 1.]	[5.06 ,26.021]	0.069	[5.06 ,26.089]
108	[5.06 ,26.089]	-25.366	0.0	[1. 0.]	[5.06 ,26.089]	0.014	[5.074,26.089]
108	[5.06 ,26.089]	-25.366	1.0	[0. 1.]	[5.074,26.089]	0.069	[5.074,26.158]
109	[5.074,26.158]	-25.406	0.0	[1. 0.]	[5.074,26.158]	0.014	[5.087,26.158]
109	[5.074,26.158]	-25.406	1.0	[0. 1.]	[5.087,26.158]	0.069	[5.087,26.227]
110	[5.087,26.227]	-25.445	0.0	[1. 0.]	[5.087,26.227]	0.014	[5.101,26.227]
110	[5.087,26.227]	-25.445	1.0	[0. 1.]	[5.101,26.227]	0.069	[5.101,26.295]
111	[5.101,26.295]	-25.483	0.0	[1. 0.]	[5.101,26.295]	0.014	[5.115,26.295]
111	[5.101,26.295]	-25.483	1.0	[0. 1.]	[5.115,26.295]	0.069	[5.115,26.364]
112	[5.115,26.364]	-25.522	0.0	[1. 0.]	[5.115,26.364]	0.014	[5.128,26.364]
112	[5.115,26.364]	-25.522	1.0	[0. 1.]	[5.128,26.364]	0.069	[5.128,26.433]
113	[5.128,26.433]	-25.559	0.0	[1. 0.]	[5.128,26.433]	0.014	[5.142,26.433]
113	[5.128,26.433]	-25.559	1.0	[0. 1.]	[5.142,26.433]	0.069	[5.142,26.501]
114	[5.142,26.501]	-25.597	0.0	[1. 0.]	[5.142,26.501]	0.014	[5.156,26.501]
114	[5.142,26.501]	-25.597	1.0	[0. 1.]	[5.156,26.501]	0.069	[5.156,26.57]
115	[5.156,26.57]	-25.634	0.0	[1. 0.]	[5.156,26.57]	0.014	[5.17,26.57]
115	[5.156,26.57]	-25.634	1.0	[0. 1.]	[5.17,26.57]	0.069	[5.17 ,26.639]
116	[5.17 ,26.639]	-25.671	0.0	[1. 0.]	[5.17 ,26.639]	0.014	[5.183,26.639]
116	[5.17 ,26.639]	-25.671	1.0	[0. 1.]	[5.183,26.639]	0.069	[5.183,26.707]
117	[5.183,26.707]	-25.707	0.0	[1. 0.]	[5.183,26.707]	0.011	[5.194,26.707]
117	[5.183,26.707]	-25.707	1.0	[0. 1.]	[5.194,26.707]	0.053	[5.194,26.761]
118	[5.194,26.761]	-25.735	0.0	[1. 0.]	[5.194,26.761]	0.011	[5.205,26.761]
118	[5.194,26.761]	-25.735	1.0	[0. 1.]	[5.205,26.761]	0.053	[5.205,26.814]
119	[5.205,26.814]	-25.763	0.0	[1. 0.]	[5.205,26.814]	0.011	[5.215,26.814]
119	[5.205,26.814]	-25.763	1.0	[0. 1.]	[5.215,26.814]	0.053	[5.215,26.868]
120	[5.215,26.868]	-25.790	0.0	[1. 0.]	[5.215,26.868]	0.011	[5.226,26.868]
120	[5.215,26.868]	-25.790	1.0	[0. 1.]	[5.226,26.868]	0.053	[5.226,26.921]
121	[5.226,26.921]	-25.818	0.0	[1. 0.]	[5.226,26.921]	0.011	[5.237,26.921]
121	[5.226,26.921]	-25.818	1.0	[0. 1.]	[5.237,26.921]	0.053	[5.237,26.974]
122	[5.237,26.974]	-25.845	0.0	[1. 0.]	[5.237,26.974]	0.011	[5.247,26.974]
122	[5.237,26.974]	-25.845	1.0	[0. 1.]	[5.247,26.974]	0.053	[5.247,27.028]
123	[5.247,27.028]	-25.872	0.0	[1. 0.]	[5.247,27.028]	0.011	[5.258,27.028]
123	[5.247,27.028]	-25.872	1.0	[0. 1.]	[5.258,27.028]	0.053	[5.258,27.081]
124	[5.258,27.081]	-25.898	0.0	[1. 0.]	[5.258,27.081]	0.011	[5.269,27.081]

124	[5.258,27.081]	-25.898	1.0	[0. 1.]	[5.269,27.081]	0.053	[5.269,27.135]
125	[5.269,27.135]	-25.925	0.0	[1. 0.]	[5.269,27.135]	0.011	[5.28 ,27.135]
125	[5.269,27.135]	-25.925	1.0	[0. 1.]	[5.28 ,27.135]	0.053	[5.28 ,27.188]
126	[5.28 ,27.188]	-25.951	0.0	[1. 0.]	[5.28 ,27.188]	0.011	[5.29 ,27.188]
126	[5.28 ,27.188]	-25.951	1.0	[0. 1.]	[5.29 ,27.188]	0.053	[5.29 ,27.242]
127	[5.29 ,27.242]	-25.977	0.0	[1. 0.]	[5.29 ,27.242]	0.011	[5.301,27.242]
127	[5.29 ,27.242]	-25.977	1.0	[0. 1.]	[5.301,27.242]	0.053	[5.301,27.295]
128	[5.301,27.295]	-26.003	0.0	[1. 0.]	[5.301,27.295]	0.011	[5.312,27.295]
128	[5.301,27.295]	-26.003	1.0	[0. 1.]	[5.312,27.295]	0.053	[5.312,27.348]
129	[5.312,27.348]	-26.028	0.0	[1. 0.]	[5.312,27.348]	0.011	[5.322,27.348]
129	[5.312,27.348]	-26.028	1.0	[0. 1.]	[5.322,27.348]	0.053	[5.322,27.402]
130	[5.322,27.402]	-26.053	0.0	[1. 0.]	[5.322,27.402]	0.011	[5.333,27.402]
130	[5.322,27.402]	-26.053	1.0	[0. 1.]	[5.333,27.402]	0.053	[5.333,27.455]
131	[5.333,27.455]	-26.078	0.0	[1. 0.]	[5.333,27.455]	0.011	[5.344,27.455]
131	[5.333,27.455]	-26.078	1.0	[0. 1.]	[5.344,27.455]	0.053	[5.344,27.509]
132	[5.344,27.509]	-26.103	0.0	[1. 0.]	[5.344,27.509]	0.011	[5.354,27.509]
132	[5.344,27.509]	-26.103	1.0	[0. 1.]	[5.354,27.509]	0.053	[5.354,27.562]
133	[5.354,27.562]	-26.128	0.0	[1. 0.]	[5.354,27.562]	0.011	[5.365,27.562]
133	[5.354,27.562]	-26.128	1.0	[0. 1.]	[5.365,27.562]	0.053	[5.365,27.615]
134	[5.365,27.615]	-26.152	0.0	[1. 0.]	[5.365,27.615]	0.011	[5.376,27.615]
134	[5.365,27.615]	-26.152	1.0	[0. 1.]	[5.376,27.615]	0.053	[5.376,27.669]
135	[5.376,27.669]	-26.176	0.0	[1. 0.]	[5.376,27.669]	0.011	[5.386,27.669]
135	[5.376,27.669]	-26.176	1.0	[0. 1.]	[5.386,27.669]	0.053	[5.386,27.722]
136	[5.386,27.722]	-26.200	0.0	[1. 0.]	[5.386,27.722]	0.011	[5.397,27.722]
136	[5.386,27.722]	-26.200	1.0	[0. 1.]	[5.397,27.722]	0.053	[5.397,27.776]
137	[5.397,27.776]	-26.224	0.0	[1. 0.]	[5.397,27.776]	0.011	[5.408,27.776]
137	[5.397,27.776]	-26.224	1.0	[0. 1.]	[5.408,27.776]	0.053	[5.408,27.829]
138	[5.408,27.829]	-26.247	0.0	[1. 0.]	[5.408,27.829]	0.011	[5.418,27.829]
138	[5.408,27.829]	-26.247	1.0	[0. 1.]	[5.418,27.829]	0.053	[5.418,27.882]
139	[5.418,27.882]	-26.271	0.0	[1. 0.]	[5.418,27.882]	0.008	[5.426,27.882]
139	[5.418,27.882]	-26.271	1.0	[0. 1.]	[5.426,27.882]	0.038	[5.426,27.921]
140	[5.426,27.921]	-26.287	0.0	[1. 0.]	[5.426,27.921]	0.008	[5.434,27.921]
140	[5.426,27.921]	-26.287	1.0	[0. 1.]	[5.434,27.921]	0.038	[5.434,27.959]
141	[5.434,27.959]	-26.303	0.0	[1. 0.]	[5.434,27.959]	0.008	[5.441,27.959]
141	[5.434,27.959]	-26.303	1.0	[0. 1.]	[5.441,27.959]	0.038	[5.441,27.997]
142	[5.441,27.997]	-26.320	0.0	[1. 0.]	[5.441,27.997]	0.008	[5.449,27.997]

142	[5.441,27.997]	-26.320	1.0	[0. 1.]	[5.449,27.997]	0.038	[5.449,28.035]
143	[5.449,28.035]	-26.336	0.0	[1. 0.]	[5.449,28.035]	0.008	[5.457,28.035]
143	[5.449,28.035]	-26.336	1.0	[0. 1.]	[5.457,28.035]	0.038	[5.457,28.073]
144	[5.457,28.073]	-26.352	0.0	[1. 0.]	[5.457,28.073]	0.008	[5.464,28.073]
144	[5.457,28.073]	-26.352	1.0	[0. 1.]	[5.464,28.073]	0.038	[5.464,28.111]
145	[5.464,28.111]	-26.368	0.0	[1. 0.]	[5.464,28.111]	0.008	[5.472,28.111]
145	[5.464,28.111]	-26.368	1.0	[0. 1.]	[5.472,28.111]	0.038	[5.472,28.149]
146	[5.472,28.149]	-26.383	0.0	[1. 0.]	[5.472,28.149]	0.008	[5.479,28.149]
146	[5.472,28.149]	-26.383	1.0	[0. 1.]	[5.479,28.149]	0.038	[5.479,28.188]
147	[5.479,28.188]	-26.399	0.0	[1. 0.]	[5.479,28.188]	0.008	[5.487,28.188]
147	[5.479,28.188]	-26.399	1.0	[0. 1.]	[5.487,28.188]	0.038	[5.487,28.226]
148	[5.487,28.226]	-26.414	0.0	[1. 0.]	[5.487,28.226]	0.008	[5.495,28.226]
148	[5.487,28.226]	-26.414	1.0	[0. 1.]	[5.495,28.226]	0.038	[5.495,28.264]
149	[5.495,28.264]	-26.430	0.0	[1. 0.]	[5.495,28.264]	0.008	[5.502,28.264]
149	[5.495,28.264]	-26.430	1.0	[0. 1.]	[5.502,28.264]	0.038	[5.502,28.302]
150	[5.502,28.302]	-26.445	0.0	[1. 0.]	[5.502,28.302]	0.008	[5.51 ,28.302]
150	[5.502,28.302]	-26.445	1.0	[0. 1.]	[5.51 ,28.302]	0.038	[5.51,28.34]
151	[5.51,28.34]	-26.460	0.0	[1. 0.]	[5.51,28.34]	0.008	[5.518,28.34]
151	[5.51,28.34]	-26.460	1.0	[0. 1.]	[5.518,28.34]	0.038	[5.518,28.378]
152	[5.518,28.378]	-26.475	0.0	[1. 0.]	[5.518,28.378]	0.008	[5.525,28.378]
152	[5.518,28.378]	-26.475	1.0	[0. 1.]	[5.525,28.378]	0.038	[5.525,28.416]
153	[5.525,28.416]	-26.490	0.0	[1. 0.]	[5.525,28.416]	0.008	[5.533,28.416]
153	[5.525,28.416]	-26.490	1.0	[0. 1.]	[5.533,28.416]	0.038	[5.533,28.455]
154	[5.533,28.455]	-26.505	0.0	[1. 0.]	[5.533,28.455]	0.008	[5.54 ,28.455]
154	[5.533,28.455]	-26.505	1.0	[0. 1.]	[5.54 ,28.455]	0.038	[5.54 ,28.493]
155	[5.54 ,28.493]	-26.520	0.0	[1. 0.]	[5.54 ,28.493]	0.008	[5.548,28.493]
155	[5.54 ,28.493]	-26.520	1.0	[0. 1.]	[5.548,28.493]	0.038	[5.548,28.531]
156	[5.548,28.531]	-26.534	0.0	[1. 0.]	[5.548,28.531]	0.008	[5.556,28.531]
156	[5.548,28.531]	-26.534	1.0	[0. 1.]	[5.556,28.531]	0.038	[5.556,28.569]
157	[5.556,28.569]	-26.549	0.0	[1. 0.]	[5.556,28.569]	0.008	[5.563,28.569]
157	[5.556,28.569]	-26.549	1.0	[0. 1.]	[5.563,28.569]	0.038	[5.563,28.607]
158	[5.563,28.607]	-26.563	0.0	[1. 0.]	[5.563,28.607]	0.008	[5.571,28.607]
158	[5.563,28.607]	-26.563	1.0	[0. 1.]	[5.571,28.607]	0.038	[5.571,28.645]
159	[5.571,28.645]	-26.577	0.0	[1. 0.]	[5.571,28.645]	0.008	[5.579,28.645]
159	[5.571,28.645]	-26.577	1.0	[0. 1.]	[5.579,28.645]	0.038	[5.579,28.683]
160	[5.579,28.683]	-26.592	0.0	[1. 0.]	[5.579,28.683]	0.008	[5.586,28.683]

160	[5.579,28.683]	-26.592	1.0	[0. 1.]	[5.586,28.683]	0.038	[5.586,28.722]
161	[5.586,28.722]	-26.606	0.0	[1. 0.]	[5.586,28.722]	0.008	[5.594,28.722]
161	[5.586,28.722]	-26.606	1.0	[0. 1.]	[5.594,28.722]	0.038	[5.594,28.76]
162	[5.594,28.76]	-26.619	0.0	[1. 0.]	[5.594,28.76]	0.008	[5.602,28.76]
162	[5.594,28.76]	-26.619	1.0	[0. 1.]	[5.602,28.76]	0.038	[5.602,28.798]
163	[5.602,28.798]	-26.633	0.0	[1. 0.]	[5.602,28.798]	0.008	[5.609,28.798]
163	[5.602,28.798]	-26.633	1.0	[0. 1.]	[5.609,28.798]	0.038	[5.609,28.836]
164	[5.609,28.836]	-26.647	0.0	[1. 0.]	[5.609,28.836]	0.008	[5.617,28.836]
164	[5.609,28.836]	-26.647	1.0	[0. 1.]	[5.617,28.836]	0.038	[5.617,28.874]
165	[5.617,28.874]	-26.660	0.0	[1. 0.]	[5.617,28.874]	0.008	[5.624,28.874]
165	[5.617,28.874]	-26.660	1.0	[0. 1.]	[5.624,28.874]	0.038	[5.624,28.912]
166	[5.624,28.912]	-26.674	0.0	[1. 0.]	[5.624,28.912]	0.008	[5.632,28.912]
166	[5.624,28.912]	-26.674	1.0	[0. 1.]	[5.632,28.912]	0.038	[5.632,28.951]
167	[5.632,28.951]	-26.687	0.0	[1. 0.]	[5.632,28.951]	0.008	[5.64 ,28.951]
167	[5.632,28.951]	-26.687	1.0	[0. 1.]	[5.64 ,28.951]	0.038	[5.64 ,28.989]
168	[5.64 ,28.989]	-26.700	0.0	[1. 0.]	[5.64 ,28.989]	0.008	[5.647,28.989]
168	[5.64 ,28.989]	-26.700	1.0	[0. 1.]	[5.647,28.989]	0.038	[5.647,29.027]
169	[5.647,29.027]	-26.713	0.0	[1. 0.]	[5.647,29.027]	0.008	[5.655,29.027]
169	[5.647,29.027]	-26.713	1.0	[0. 1.]	[5.655,29.027]	0.038	[5.655,29.065]
170	[5.655,29.065]	-26.726	0.0	[1. 0.]	[5.655,29.065]	0.008	[5.663,29.065]
170	[5.655,29.065]	-26.726	1.0	[0. 1.]	[5.663,29.065]	0.038	[5.663,29.103]
171	[5.663,29.103]	-26.739	0.0	[1. 0.]	[5.663,29.103]	0.008	[5.67 ,29.103]
171	[5.663,29.103]	-26.739	1.0	[0. 1.]	[5.67 ,29.103]	0.038	[5.67 ,29.141]
172	[5.67 ,29.141]	-26.752	0.0	[1. 0.]	[5.67 ,29.141]	0.005	[5.675,29.141]
172	[5.67 ,29.141]	-26.752	1.0	[0. 1.]	[5.675,29.141]	0.023	[5.675,29.164]
173	[5.675,29.164]	-26.759	0.0	[1. 0.]	[5.675,29.164]	0.005	[5.679,29.164]
173	[5.675,29.164]	-26.759	1.0	[0. 1.]	[5.679,29.164]	0.023	[5.679,29.187]
174	[5.679,29.187]	-26.767	0.0	[1. 0.]	[5.679,29.187]	0.005	[5.684,29.187]
174	[5.679,29.187]	-26.767	1.0	[0. 1.]	[5.684,29.187]	0.023	[5.684,29.21]
175	[5.684,29.21]	-26.774	0.0	[1. 0.]	[5.684,29.21]	0.005	[5.688,29.21]
175	[5.684,29.21]	-26.774	1.0	[0. 1.]	[5.688,29.21]	0.023	[5.688,29.233]
176	[5.688,29.233]	-26.782	0.0	[1. 0.]	[5.688,29.233]	0.005	[5.693,29.233]
176	[5.688,29.233]	-26.782	1.0	[0. 1.]	[5.693,29.233]	0.023	[5.693,29.256]
177	[5.693,29.256]	-26.789	0.0	[1. 0.]	[5.693,29.256]	0.005	[5.698,29.256]
177	[5.693,29.256]	-26.789	1.0	[0. 1.]	[5.698,29.256]	0.023	[5.698,29.279]
178	[5.698,29.279]	-26.797	0.0	[1. 0.]	[5.698,29.279]	0.005	[5.702,29.279]

178	[5.698,29.279]	-26.797	1.0	[0. 1.]	[5.702,29.279]	0.023	[5.702,29.301]
179	[5.702,29.301]	-26.804	0.0	[1. 0.]	[5.702,29.301]	0.005	[5.707,29.301]
179	[5.702,29.301]	-26.804	1.0	[0. 1.]	[5.707,29.301]	0.023	[5.707,29.324]
180	[5.707,29.324]	-26.811	0.0	[1. 0.]	[5.707,29.324]	0.005	[5.711,29.324]
180	[5.707,29.324]	-26.811	1.0	[0. 1.]	[5.711,29.324]	0.023	[5.711,29.347]
181	[5.711,29.347]	-26.819	0.0	[1. 0.]	[5.711,29.347]	0.005	[5.716,29.347]
181	[5.711,29.347]	-26.819	1.0	[0. 1.]	[5.716,29.347]	0.023	[5.716,29.37]
182	[5.716,29.37]	-26.826	0.0	[1. 0.]	[5.716,29.37]	0.005	[5.721,29.37]
182	[5.716,29.37]	-26.826	1.0	[0. 1.]	[5.721,29.37]	0.023	[5.721,29.393]
183	[5.721,29.393]	-26.833	0.0	[1. 0.]	[5.721,29.393]	0.005	[5.725,29.393]
183	[5.721,29.393]	-26.833	1.0	[0. 1.]	[5.725,29.393]	0.023	[5.725,29.416]
184	[5.725,29.416]	-26.840	0.0	[1. 0.]	[5.725,29.416]	0.005	[5.73 ,29.416]
184	[5.725,29.416]	-26.840	1.0	[0. 1.]	[5.73 ,29.416]	0.023	[5.73 ,29.439]
185	[5.73 ,29.439]	-26.847	0.0	[1. 0.]	[5.73 ,29.439]	0.005	[5.734,29.439]
185	[5.73 ,29.439]	-26.847	1.0	[0. 1.]	[5.734,29.439]	0.023	[5.734,29.462]
186	[5.734,29.462]	-26.854	0.0	[1. 0.]	[5.734,29.462]	0.005	[5.739,29.462]
186	[5.734,29.462]	-26.854	1.0	[0. 1.]	[5.739,29.462]	0.023	[5.739,29.485]
187	[5.739,29.485]	-26.861	0.0	[1. 0.]	[5.739,29.485]	0.005	[5.743,29.485]
187	[5.739,29.485]	-26.861	1.0	[0. 1.]	[5.743,29.485]	0.023	[5.743,29.507]
188	[5.743,29.507]	-26.868	0.0	[1. 0.]	[5.743,29.507]	0.005	[5.748,29.507]
188	[5.743,29.507]	-26.868	1.0	[0. 1.]	[5.748,29.507]	0.023	[5.748,29.53]
189	[5.748,29.53]	-26.875	0.0	[1. 0.]	[5.748,29.53]	0.005	[5.753,29.53]
189	[5.748,29.53]	-26.875	1.0	[0. 1.]	[5.753,29.53]	0.023	[5.753,29.553]
190	[5.753,29.553]	-26.882	0.0	[1. 0.]	[5.753,29.553]	0.005	[5.757,29.553]
190	[5.753,29.553]	-26.882	1.0	[0. 1.]	[5.757,29.553]	0.023	[5.757,29.576]
191	[5.757,29.576]	-26.889	0.0	[1. 0.]	[5.757,29.576]	0.005	[5.762,29.576]
191	[5.757,29.576]	-26.889	1.0	[0. 1.]	[5.762,29.576]	0.023	[5.762,29.599]
192	[5.762,29.599]	-26.895	0.0	[1. 0.]	[5.762,29.599]	0.005	[5.766,29.599]
192	[5.762,29.599]	-26.895	1.0	[0. 1.]	[5.766,29.599]	0.023	[5.766,29.622]
193	[5.766,29.622]	-26.902	0.0	[1. 0.]	[5.766,29.622]	0.005	[5.771,29.622]
193	[5.766,29.622]	-26.902	1.0	[0. 1.]	[5.771,29.622]	0.023	[5.771,29.645]
194	[5.771,29.645]	-26.909	0.0	[1. 0.]	[5.771,29.645]	0.005	[5.775,29.645]
194	[5.771,29.645]	-26.909	1.0	[0. 1.]	[5.775,29.645]	0.023	[5.775,29.668]
195	[5.775,29.668]	-26.916	0.0	[1. 0.]	[5.775,29.668]	0.005	[5.78 ,29.668]
195	[5.775,29.668]	-26.916	1.0	[0. 1.]	[5.78 ,29.668]	0.023	[5.78 ,29.691]
196	[5.78 ,29.691]	-26.922	0.0	[1. 0.]	[5.78 ,29.691]	0.005	[5.785,29.691]

196	[5.78 ,29.691]	-26.922	1.0	[0. 1.]	[5.785,29.691]	0.023	[5.785,29.713]
197	[5.785,29.713]	-26.929	0.0	[1. 0.]	[5.785,29.713]	0.005	[5.789,29.713]
197	[5.785,29.713]	-26.929	1.0	[0. 1.]	[5.789,29.713]	0.023	[5.789,29.736]
198	[5.789,29.736]	-26.935	0.0	[1. 0.]	[5.789,29.736]	0.005	[5.794,29.736]
198	[5.789,29.736]	-26.935	1.0	[0. 1.]	[5.794,29.736]	0.023	[5.794,29.759]
199	[5.794,29.759]	-26.942	0.0	[1. 0.]	[5.794,29.759]	0.005	[5.798,29.759]
199	[5.794,29.759]	-26.942	1.0	[0. 1.]	[5.798,29.759]	0.023	[5.798,29.782]
200	[5.798,29.782]	-26.948	0.0	[1. 0.]	[5.798,29.782]	0.005	[5.803,29.782]
200	[5.798,29.782]	-26.948	1.0	[0. 1.]	[5.803,29.782]	0.023	[5.803,29.805]
201	[5.803,29.805]	-26.955	0.0	[1. 0.]	[5.803,29.805]	0.005	[5.807,29.805]
201	[5.803,29.805]	-26.955	1.0	[0. 1.]	[5.807,29.805]	0.023	[5.807,29.828]
202	[5.807,29.828]	-26.961	0.0	[1. 0.]	[5.807,29.828]	0.005	[5.812,29.828]
202	[5.807,29.828]	-26.961	1.0	[0. 1.]	[5.812,29.828]	0.023	[5.812,29.851]
203	[5.812,29.851]	-26.967	0.0	[1. 0.]	[5.812,29.851]	0.005	[5.817,29.851]
203	[5.812,29.851]	-26.967	1.0	[0. 1.]	[5.817,29.851]	0.023	[5.817,29.874]
204	[5.817,29.874]	-26.974	0.0	[1. 0.]	[5.817,29.874]	0.005	[5.821,29.874]
204	[5.817,29.874]	-26.974	1.0	[0. 1.]	[5.821,29.874]	0.023	[5.821,29.897]
205	[5.821,29.897]	-26.980	0.0	[1. 0.]	[5.821,29.897]	0.005	[5.826,29.897]
205	[5.821,29.897]	-26.980	1.0	[0. 1.]	[5.826,29.897]	0.023	[5.826,29.919]
206	[5.826,29.919]	-26.986	0.0	[1. 0.]	[5.826,29.919]	0.005	[5.83 ,29.919]
206	[5.826,29.919]	-26.986	1.0	[0. 1.]	[5.83 ,29.919]	0.023	[5.83 ,29.942]
207	[5.83 ,29.942]	-26.992	0.0	[1. 0.]	[5.83 ,29.942]	0.005	[5.835,29.942]
207	[5.83 ,29.942]	-26.992	1.0	[0. 1.]	[5.835,29.942]	0.023	[5.835,29.965]
208	[5.835,29.965]	-26.998	0.0	[1. 0.]	[5.835,29.965]	0.005	[5.84 ,29.965]
208	[5.835,29.965]	-26.998	1.0	[0. 1.]	[5.84 ,29.965]	0.023	[5.84 ,29.988]
209	[5.84 ,29.988]	-27.004	0.0	[1. 0.]	[5.84 ,29.988]	0.005	[5.844,29.988]
209	[5.84 ,29.988]	-27.004	1.0	[0. 1.]	[5.844,29.988]	0.023	[5.844,30.011]
210	[5.844,30.011]	-27.010	0.0	[1. 0.]	[5.844,30.011]	0.005	[5.849,30.011]
210	[5.844,30.011]	-27.010	1.0	[0. 1.]	[5.849,30.011]	0.023	[5.849,30.034]
211	[5.849,30.034]	-27.016	0.0	[1. 0.]	[5.849,30.034]	0.005	[5.853,30.034]
211	[5.849,30.034]	-27.016	1.0	[0. 1.]	[5.853,30.034]	0.023	[5.853,30.057]
212	[5.853,30.057]	-27.022	0.0	[1. 0.]	[5.853,30.057]	0.005	[5.858,30.057]
212	[5.853,30.057]	-27.022	1.0	[0. 1.]	[5.858,30.057]	0.023	[5.858,30.08]
213	[5.858,30.08]	-27.028	0.0	[1. 0.]	[5.858,30.08]	0.005	[5.862,30.08]
213	[5.858,30.08]	-27.028	1.0	[0. 1.]	[5.862,30.08]	0.023	[5.862,30.103]
214	[5.862,30.103]	-27.034	0.0	[1. 0.]	[5.862,30.103]	0.005	[5.867,30.103]

214	[5.862,30.103]	-27.034	1.0	[0. 1.]	[5.867,30.103]	0.023	[5.867,30.125]
215	[5.867,30.125]	-27.040	0.0	[1. 0.]	[5.867,30.125]	0.005	[5.872,30.125]
215	[5.867,30.125]	-27.040	1.0	[0. 1.]	[5.872,30.125]	0.023	[5.872,30.148]
216	[5.872,30.148]	-27.046	0.0	[1. 0.]	[5.872,30.148]	0.005	[5.876,30.148]
216	[5.872,30.148]	-27.046	1.0	[0. 1.]	[5.876,30.148]	0.023	[5.876,30.171]
217	[5.876,30.171]	-27.051	0.0	[1. 0.]	[5.876,30.171]	0.005	[5.881,30.171]
217	[5.876,30.171]	-27.051	1.0	[0. 1.]	[5.881,30.171]	0.023	[5.881,30.194]
218	[5.881,30.194]	-27.057	0.0	[1. 0.]	[5.881,30.194]	0.005	[5.885,30.194]
218	[5.881,30.194]	-27.057	1.0	[0. 1.]	[5.885,30.194]	0.023	[5.885,30.217]
219	[5.885,30.217]	-27.063	0.0	[1. 0.]	[5.885,30.217]	0.005	[5.89 ,30.217]
219	[5.885,30.217]	-27.063	1.0	[0. 1.]	[5.89 ,30.217]	0.023	[5.89,30.24]
220	[5.89,30.24]	-27.068	0.0	[1. 0.]	[5.89,30.24]	0.005	[5.894,30.24]
220	[5.89,30.24]	-27.068	1.0	[0. 1.]	[5.894,30.24]	0.023	[5.894,30.263]
221	[5.894,30.263]	-27.074	0.0	[1. 0.]	[5.894,30.263]	0.005	[5.899,30.263]
221	[5.894,30.263]	-27.074	1.0	[0. 1.]	[5.899,30.263]	0.023	[5.899,30.286]
222	[5.899,30.286]	-27.079	0.0	[1. 0.]	[5.899,30.286]	0.005	[5.904,30.286]
222	[5.899,30.286]	-27.079	1.0	[0. 1.]	[5.904,30.286]	0.023	[5.904,30.309]
223	[5.904,30.309]	-27.085	0.0	[1. 0.]	[5.904,30.309]	0.005	[5.908,30.309]
223	[5.904,30.309]	-27.085	1.0	[0. 1.]	[5.908,30.309]	0.023	[5.908,30.331]
224	[5.908,30.331]	-27.090	0.0	[1. 0.]	[5.908,30.331]	0.005	[5.913,30.331]
224	[5.908,30.331]	-27.090	1.0	[0. 1.]	[5.913,30.331]	0.023	[5.913,30.354]
225	[5.913,30.354]	-27.096	0.0	[1. 0.]	[5.913,30.354]	0.005	[5.917,30.354]
225	[5.913,30.354]	-27.096	1.0	[0. 1.]	[5.917,30.354]	0.023	[5.917,30.377]
226	[5.917,30.377]	-27.101	0.0	[1. 0.]	[5.917,30.377]	0.005	[5.922,30.377]
226	[5.917,30.377]	-27.101	1.0	[0. 1.]	[5.922,30.377]	0.023	[5.922,30.4]
227	[5.922,30.4]	-27.106	0.0	[1. 0.]	[5.922,30.4]	0.005	[5.927,30.4]
227	[5.922,30.4]	-27.106	1.0	[0. 1.]	[5.927,30.4]	0.023	[5.927,30.423]
228	[5.927,30.423]	-27.112	0.0	[1. 0.]	[5.927,30.423]	0.005	[5.931,30.423]
228	[5.927,30.423]	-27.112	1.0	[0. 1.]	[5.931,30.423]	0.023	[5.931,30.446]
229	[5.931,30.446]	-27.117	0.0	[1. 0.]	[5.931,30.446]	0.005	[5.936,30.446]
229	[5.931,30.446]	-27.117	1.0	[0. 1.]	[5.936,30.446]	0.023	[5.936,30.469]
230	[5.936,30.469]	-27.122	0.0	[1. 0.]	[5.936,30.469]	0.002	[5.937,30.469]
230	[5.936,30.469]	-27.122	1.0	[0. 1.]	[5.937,30.469]	0.008	[5.937,30.476]
231	[5.937,30.476]	-27.124	NaN	None	None	NaN	None

- The complete output table of the cyclic coordinate search set 2:

k	$x^{(k)}$	$f(x^{(k)})$	j	$d_j^{(k)}$	y_j	$\alpha_j^{(k)}$	y_{j+1}
0	[10,35]	50641.000	0.0	[1. 0.]	[10,35]	-3.166	[6.834,35.]
0	[10,35]	50641.000	1.0	[0. 1.]	[6.834,35.]	-0.041	[6.834,34.959]
1	[6.834,34.959]	-27.329	0.0	[1. 0.]	[6.834,34.959]	-0.008	[6.826,34.959]
1	[6.834,34.959]	-27.329	1.0	[0. 1.]	[6.826,34.959]	-0.038	[6.826,34.921]
2	[6.826,34.921]	-27.334	0.0	[1. 0.]	[6.826,34.921]	-0.008	[6.819,34.921]
2	[6.826,34.921]	-27.334	1.0	[0. 1.]	[6.819,34.921]	-0.038	[6.819,34.883]
3	[6.819,34.883]	-27.339	0.0	[1. 0.]	[6.819,34.883]	-0.008	[6.811,34.883]
3	[6.819,34.883]	-27.339	1.0	[0. 1.]	[6.811,34.883]	-0.038	[6.811,34.844]
4	[6.811,34.844]	-27.344	0.0	[1. 0.]	[6.811,34.844]	-0.008	[6.803,34.844]
4	[6.811,34.844]	-27.344	1.0	[0. 1.]	[6.803,34.844]	-0.038	[6.803,34.806]
5	[6.803,34.806]	-27.349	0.0	[1. 0.]	[6.803,34.806]	-0.008	[6.796,34.806]
5	[6.803,34.806]	-27.349	1.0	[0. 1.]	[6.796,34.806]	-0.038	[6.796,34.768]
6	[6.796,34.768]	-27.353	0.0	[1. 0.]	[6.796,34.768]	-0.008	[6.788,34.768]
6	[6.796,34.768]	-27.353	1.0	[0. 1.]	[6.788,34.768]	-0.038	[6.788,34.73]
7	[6.788,34.73]	-27.358	0.0	[1. 0.]	[6.788,34.73]	-0.008	[6.78,34.73]
7	[6.788,34.73]	-27.358	1.0	[0. 1.]	[6.78,34.73]	-0.038	[6.78 ,34.692]
8	[6.78 ,34.692]	-27.362	0.0	[1. 0.]	[6.78 ,34.692]	-0.008	[6.773,34.692]
8	[6.78 ,34.692]	-27.362	1.0	[0. 1.]	[6.773,34.692]	-0.038	[6.773,34.654]
9	[6.773,34.654]	-27.366	0.0	[1. 0.]	[6.773,34.654]	-0.005	[6.768,34.654]
9	[6.773,34.654]	-27.366	1.0	[0. 1.]	[6.768,34.654]	-0.023	[6.768,34.631]
10	[6.768,34.631]	-27.369	0.0	[1. 0.]	[6.768,34.631]	-0.005	[6.764,34.631]
10	[6.768,34.631]	-27.369	1.0	[0. 1.]	[6.764,34.631]	-0.023	[6.764,34.608]
11	[6.764,34.608]	-27.371	0.0	[1. 0.]	[6.764,34.608]	-0.005	[6.759,34.608]
11	[6.764,34.608]	-27.371	1.0	[0. 1.]	[6.759,34.608]	-0.023	[6.759,34.585]
12	[6.759,34.585]	-27.373	0.0	[1. 0.]	[6.759,34.585]	-0.005	[6.754,34.585]
12	[6.759,34.585]	-27.373	1.0	[0. 1.]	[6.754,34.585]	-0.023	[6.754,34.562]
13	[6.754,34.562]	-27.376	0.0	[1. 0.]	[6.754,34.562]	-0.005	[6.75 ,34.562]
13	[6.754,34.562]	-27.376	1.0	[0. 1.]	[6.75 ,34.562]	-0.023	[6.75 ,34.539]
14	[6.75 ,34.539]	-27.378	0.0	[1. 0.]	[6.75 ,34.539]	-0.005	[6.745,34.539]
14	[6.75 ,34.539]	-27.378	1.0	[0. 1.]	[6.745,34.539]	-0.023	[6.745,34.516]
15	[6.745,34.516]	-27.380	0.0	[1. 0.]	[6.745,34.516]	-0.005	[6.741,34.516]
15	[6.745,34.516]	-27.380	1.0	[0. 1.]	[6.741,34.516]	-0.023	[6.741,34.493]
16	[6.741,34.493]	-27.383	0.0	[1. 0.]	[6.741,34.493]	-0.005	[6.736,34.493]

16	[6.741,34.493]	-27.383	1.0	[0. 1.]	[6.736,34.493]	-0.023	[6.736,34.471]
17	[6.736,34.471]	-27.385	0.0	[1. 0.]	[6.736,34.471]	-0.005	[6.732,34.471]
17	[6.736,34.471]	-27.385	1.0	[0. 1.]	[6.732,34.471]	-0.023	[6.732,34.448]
18	[6.732,34.448]	-27.387	0.0	[1. 0.]	[6.732,34.448]	-0.005	[6.727,34.448]
18	[6.732,34.448]	-27.387	1.0	[0. 1.]	[6.727,34.448]	-0.023	[6.727,34.425]
19	[6.727,34.425]	-27.389	0.0	[1. 0.]	[6.727,34.425]	-0.005	[6.722,34.425]
19	[6.727,34.425]	-27.389	1.0	[0. 1.]	[6.722,34.425]	-0.023	[6.722,34.402]
20	[6.722,34.402]	-27.391	0.0	[1. 0.]	[6.722,34.402]	-0.005	[6.718,34.402]
20	[6.722,34.402]	-27.391	1.0	[0. 1.]	[6.718,34.402]	-0.023	[6.718,34.379]
21	[6.718,34.379]	-27.393	0.0	[1. 0.]	[6.718,34.379]	-0.005	[6.713,34.379]
21	[6.718,34.379]	-27.393	1.0	[0. 1.]	[6.713,34.379]	-0.023	[6.713,34.356]
22	[6.713,34.356]	-27.395	0.0	[1. 0.]	[6.713,34.356]	-0.005	[6.709,34.356]
22	[6.713,34.356]	-27.395	1.0	[0. 1.]	[6.709,34.356]	-0.023	[6.709,34.333]
23	[6.709,34.333]	-27.397	0.0	[1. 0.]	[6.709,34.333]	-0.005	[6.704,34.333]
23	[6.709,34.333]	-27.397	1.0	[0. 1.]	[6.704,34.333]	-0.023	[6.704,34.31]
24	[6.704,34.31]	-27.399	0.0	[1. 0.]	[6.704,34.31]	-0.005	[6.7 ,34.31]
24	[6.704,34.31]	-27.399	1.0	[0. 1.]	[6.7 ,34.31]	-0.023	[6.7 ,34.287]
25	[6.7 ,34.287]	-27.401	0.0	[1. 0.]	[6.7 ,34.287]	-0.005	[6.695,34.287]
25	[6.7 ,34.287]	-27.401	1.0	[0. 1.]	[6.695,34.287]	-0.023	[6.695,34.265]
26	[6.695,34.265]	-27.402	0.0	[1. 0.]	[6.695,34.265]	-0.005	[6.69 ,34.265]
26	[6.695,34.265]	-27.402	1.0	[0. 1.]	[6.69 ,34.265]	-0.023	[6.69 ,34.242]
27	[6.69 ,34.242]	-27.404	0.0	[1. 0.]	[6.69 ,34.242]	-0.005	[6.686,34.242]
27	[6.69 ,34.242]	-27.404	1.0	[0. 1.]	[6.686,34.242]	-0.023	[6.686,34.219]
28	[6.686,34.219]	-27.406	0.0	[1. 0.]	[6.686,34.219]	-0.005	[6.681,34.219]
28	[6.686,34.219]	-27.406	1.0	[0. 1.]	[6.681,34.219]	-0.023	[6.681,34.196]
29	[6.681,34.196]	-27.408	0.0	[1. 0.]	[6.681,34.196]	-0.005	[6.677,34.196]
29	[6.681,34.196]	-27.408	1.0	[0. 1.]	[6.677,34.196]	-0.023	[6.677,34.173]
30	[6.677,34.173]	-27.409	0.0	[1. 0.]	[6.677,34.173]	-0.005	[6.672,34.173]
30	[6.677,34.173]	-27.409	1.0	[0. 1.]	[6.672,34.173]	-0.023	[6.672,34.15]
31	[6.672,34.15]	-27.411	0.0	[1. 0.]	[6.672,34.15]	-0.005	[6.667,34.15]
31	[6.672,34.15]	-27.411	1.0	[0. 1.]	[6.667,34.15]	-0.023	[6.667,34.127]
32	[6.667,34.127]	-27.412	0.0	[1. 0.]	[6.667,34.127]	-0.005	[6.663,34.127]
32	[6.667,34.127]	-27.412	1.0	[0. 1.]	[6.663,34.127]	-0.023	[6.663,34.104]
33	[6.663,34.104]	-27.414	0.0	[1. 0.]	[6.663,34.104]	-0.005	[6.658,34.104]
33	[6.663,34.104]	-27.414	1.0	[0. 1.]	[6.658,34.104]	-0.023	[6.658,34.081]
34	[6.658,34.081]	-27.415	0.0	[1. 0.]	[6.658,34.081]	-0.005	[6.654,34.081]

34	[6.658,34.081]	-27.415	1.0	[0. 1.]	[6.654,34.081]	-0.023	[6.654,34.059]
35	[6.654,34.059]	-27.417	0.0	[1. 0.]	[6.654,34.059]	-0.005	[6.649,34.059]
35	[6.654,34.059]	-27.417	1.0	[0. 1.]	[6.649,34.059]	-0.023	[6.649,34.036]
36	[6.649,34.036]	-27.418	0.0	[1. 0.]	[6.649,34.036]	-0.005	[6.645,34.036]
36	[6.649,34.036]	-27.418	1.0	[0. 1.]	[6.645,34.036]	-0.023	[6.645,34.013]
37	[6.645,34.013]	-27.420	0.0	[1. 0.]	[6.645,34.013]	-0.005	[6.64 ,34.013]
37	[6.645,34.013]	-27.420	1.0	[0. 1.]	[6.64 ,34.013]	-0.023	[6.64,33.99]
38	[6.64,33.99]	-27.421	0.0	[1. 0.]	[6.64,33.99]	-0.005	[6.635,33.99]
38	[6.64,33.99]	-27.421	1.0	[0. 1.]	[6.635,33.99]	-0.023	[6.635,33.967]
39	[6.635,33.967]	-27.422	0.0	[1. 0.]	[6.635,33.967]	-0.005	[6.631,33.967]
39	[6.635,33.967]	-27.422	1.0	[0. 1.]	[6.631,33.967]	-0.023	[6.631,33.944]
40	[6.631,33.944]	-27.423	0.0	[1. 0.]	[6.631,33.944]	-0.005	[6.626,33.944]
40	[6.631,33.944]	-27.423	1.0	[0. 1.]	[6.626,33.944]	-0.023	[6.626,33.921]
41	[6.626,33.921]	-27.425	0.0	[1. 0.]	[6.626,33.921]	-0.005	[6.622,33.921]
41	[6.626,33.921]	-27.425	1.0	[0. 1.]	[6.622,33.921]	-0.023	[6.622,33.898]
42	[6.622,33.898]	-27.426	0.0	[1. 0.]	[6.622,33.898]	-0.005	[6.617,33.898]
42	[6.622,33.898]	-27.426	1.0	[0. 1.]	[6.617,33.898]	-0.023	[6.617,33.875]
43	[6.617,33.875]	-27.427	0.0	[1. 0.]	[6.617,33.875]	-0.005	[6.613,33.875]
43	[6.617,33.875]	-27.427	1.0	[0. 1.]	[6.613,33.875]	-0.023	[6.613,33.853]
44	[6.613,33.853]	-27.428	0.0	[1. 0.]	[6.613,33.853]	-0.005	[6.608,33.853]
44	[6.613,33.853]	-27.428	1.0	[0. 1.]	[6.608,33.853]	-0.023	[6.608,33.83]
45	[6.608,33.83]	-27.429	0.0	[1. 0.]	[6.608,33.83]	-0.005	[6.603,33.83]
45	[6.608,33.83]	-27.429	1.0	[0. 1.]	[6.603,33.83]	-0.023	[6.603,33.807]
46	[6.603,33.807]	-27.430	0.0	[1. 0.]	[6.603,33.807]	-0.005	[6.599,33.807]
46	[6.603,33.807]	-27.430	1.0	[0. 1.]	[6.599,33.807]	-0.023	[6.599,33.784]
47	[6.599,33.784]	-27.431	0.0	[1. 0.]	[6.599,33.784]	-0.005	[6.594,33.784]
47	[6.599,33.784]	-27.431	1.0	[0. 1.]	[6.594,33.784]	-0.023	[6.594,33.761]
48	[6.594,33.761]	-27.432	0.0	[1. 0.]	[6.594,33.761]	-0.005	[6.59 ,33.761]
48	[6.594,33.761]	-27.432	1.0	[0. 1.]	[6.59 ,33.761]	-0.023	[6.59 ,33.738]
49	[6.59 ,33.738]	-27.432	0.0	[1. 0.]	[6.59 ,33.738]	-0.005	[6.585,33.738]
49	[6.59 ,33.738]	-27.432	1.0	[0. 1.]	[6.585,33.738]	-0.023	[6.585,33.715]
50	[6.585,33.715]	-27.433	0.0	[1. 0.]	[6.585,33.715]	-0.005	[6.581,33.715]
50	[6.585,33.715]	-27.433	1.0	[0. 1.]	[6.581,33.715]	-0.023	[6.581,33.692]
51	[6.581,33.692]	-27.434	0.0	[1. 0.]	[6.581,33.692]	-0.005	[6.576,33.692]
51	[6.581,33.692]	-27.434	1.0	[0. 1.]	[6.576,33.692]	-0.023	[6.576,33.669]
52	[6.576,33.669]	-27.435	0.0	[1. 0.]	[6.576,33.669]	-0.005	[6.571,33.669]

52	[6.576,33.669]	-27.435	1.0	[0. 1.]	[6.571,33.669]	-0.023	[6.571,33.647]
53	[6.571,33.647]	-27.435	0.0	[1. 0.]	[6.571,33.647]	-0.005	[6.567,33.647]
53	[6.571,33.647]	-27.435	1.0	[0. 1.]	[6.567,33.647]	-0.023	[6.567,33.624]
54	[6.567,33.624]	-27.436	0.0	[1. 0.]	[6.567,33.624]	-0.005	[6.562,33.624]
54	[6.567,33.624]	-27.436	1.0	[0. 1.]	[6.562,33.624]	-0.023	[6.562,33.601]
55	[6.562,33.601]	-27.437	0.0	[1. 0.]	[6.562,33.601]	-0.005	[6.558,33.601]
55	[6.562,33.601]	-27.437	1.0	[0. 1.]	[6.558,33.601]	-0.023	[6.558,33.578]
56	[6.558,33.578]	-27.437	0.0	[1. 0.]	[6.558,33.578]	-0.005	[6.553,33.578]
56	[6.558,33.578]	-27.437	1.0	[0. 1.]	[6.553,33.578]	-0.023	[6.553,33.555]
57	[6.553,33.555]	-27.438	0.0	[1. 0.]	[6.553,33.555]	-0.005	[6.548,33.555]
57	[6.553,33.555]	-27.438	1.0	[0. 1.]	[6.548,33.555]	-0.023	[6.548,33.532]
58	[6.548,33.532]	-27.438	0.0	[1. 0.]	[6.548,33.532]	-0.005	[6.544,33.532]
58	[6.548,33.532]	-27.438	1.0	[0. 1.]	[6.544,33.532]	-0.023	[6.544,33.509]
59	[6.544,33.509]	-27.439	0.0	[1. 0.]	[6.544,33.509]	-0.005	[6.539,33.509]
59	[6.544,33.509]	-27.439	1.0	[0. 1.]	[6.539,33.509]	-0.023	[6.539,33.486]
60	[6.539,33.486]	-27.439	0.0	[1. 0.]	[6.539,33.486]	-0.005	[6.535,33.486]
60	[6.539,33.486]	-27.439	1.0	[0. 1.]	[6.535,33.486]	-0.023	[6.535,33.463]
61	[6.535,33.463]	-27.439	0.0	[1. 0.]	[6.535,33.463]	-0.005	[6.53 ,33.463]
61	[6.535,33.463]	-27.439	1.0	[0. 1.]	[6.53 ,33.463]	-0.023	[6.53 ,33.441]
62	[6.53 ,33.441]	-27.440	0.0	[1. 0.]	[6.53 ,33.441]	-0.005	[6.526,33.441]
62	[6.53 ,33.441]	-27.440	1.0	[0. 1.]	[6.526,33.441]	-0.023	[6.526,33.418]
63	[6.526,33.418]	-27.440	0.0	[1. 0.]	[6.526,33.418]	-0.005	[6.521,33.418]
63	[6.526,33.418]	-27.440	1.0	[0. 1.]	[6.521,33.418]	-0.023	[6.521,33.395]
64	[6.521,33.395]	-27.440	0.0	[1. 0.]	[6.521,33.395]	-0.005	[6.516,33.395]
64	[6.521,33.395]	-27.440	1.0	[0. 1.]	[6.516,33.395]	-0.023	[6.516,33.372]
65	[6.516,33.372]	-27.440	0.0	[1. 0.]	[6.516,33.372]	-0.005	[6.512,33.372]
65	[6.516,33.372]	-27.440	1.0	[0. 1.]	[6.512,33.372]	-0.023	[6.512,33.349]
66	[6.512,33.349]	-27.440	0.0	[1. 0.]	[6.512,33.349]	-0.005	[6.507,33.349]
66	[6.512,33.349]	-27.440	1.0	[0. 1.]	[6.507,33.349]	-0.023	[6.507,33.326]
67	[6.507,33.326]	-27.440	0.0	[1. 0.]	[6.507,33.326]	-0.005	[6.503,33.326]
67	[6.507,33.326]	-27.440	1.0	[0. 1.]	[6.503,33.326]	-0.023	[6.503,33.303]
68	[6.503,33.303]	-27.440	0.0	[1. 0.]	[6.503,33.303]	-0.005	[6.498,33.303]
68	[6.503,33.303]	-27.440	1.0	[0. 1.]	[6.498,33.303]	-0.023	[6.498,33.28]
69	[6.498,33.28]	-27.440	0.0	[1. 0.]	[6.498,33.28]	-0.005	[6.494,33.28]
69	[6.498,33.28]	-27.440	1.0	[0. 1.]	[6.494,33.28]	-0.023	[6.494,33.257]
70	[6.494,33.257]	-27.440	0.0	[1. 0.]	[6.494,33.257]	-0.005	[6.489,33.257]

70	[6.494,33.257]	-27.440	1.0	[0. 1.]	[6.489,33.257]	-0.023	[6.489,33.235]
71	[6.489,33.235]	-27.440	0.0	[1. 0.]	[6.489,33.235]	-0.005	[6.484,33.235]
71	[6.489,33.235]	-27.440	1.0	[0. 1.]	[6.484,33.235]	-0.023	[6.484,33.212]
72	[6.484,33.212]	-27.440	0.0	[1. 0.]	[6.484,33.212]	-0.005	[6.48 ,33.212]
72	[6.484,33.212]	-27.440	1.0	[0. 1.]	[6.48 ,33.212]	-0.023	[6.48 ,33.189]
73	[6.48 ,33.189]	-27.440	0.0	[1. 0.]	[6.48 ,33.189]	-0.002	[6.478,33.189]
73	[6.48 ,33.189]	-27.440	1.0	[0. 1.]	[6.478,33.189]	-0.008	[6.478,33.181]
74	[6.478,33.181]	-27.440	NaN	None	None	NaN	None