BOĞAZIÇI UNIVERSITY

NONLINEAR MODELS IN OPERATIONS RESEARCH
IE 440

Homework 4

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

28 November 2019

Department of Industrial Engineering
Boğaziçi University

1 Introduction

The project is implemented using Python as the programming language. First the given functions and their gradient functions are converted to lambda functions using ”sympy” package.

The source code used to import required dependencies, converting functions to lambda expressions:

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

x1 = Symbol("x1")
x2 = Symbol("x2")

func1 = (5*x1 - x2)**4 + (x1 - 2)**2 + x1 - 2*x2 + 12
func2 = 100*(x2 - x1**2)**2 + (1 - x1)**2


f1 = lambdify([[x1,x2]], func1, "numpy")
f2 = lambdify([[x1,x2]], func2, "numpy")

gf1 = lambdify([[x1,x2]], func1.diff([[x1, x2]]), "numpy")
gf2 = lambdify([[x1,x2]], func2.diff([[x1, x2]]), "numpy")

grad_f1 = lambda x_arr : np.array(gf1(x_arr)).reshape(1,2)
grad_f2 = lambda x_arr : np.array(gf2(x_arr)).reshape(1,2)

hf1 = lambdify([[x1,x2]], (func1.diff([[x1, x2]])).diff([[x1, x2]]), "numpy")
hf2 = lambdify([[x1,x2]], (func2.diff([[x1, x2]])).diff([[x1, x2]]), "numpy")

hess_f1= lambda x_arr : np.array(hf1(np.array(x_arr).reshape(2,)))
hess_f2= lambda x_arr : np.array(hf2(np.array(x_arr).reshape(2,)))
```

Some useful functions for output table construction:

```
np_str = lambda x_k : np.array2string(x_k.reshape(len(x_k)), precision=3, separator=',')

f_str = lambda x : "{0:.4f}".format(x)

class OutputTable:
    def __init__(self):
        self.table = pd.DataFrame([],columns=['k', 'x^k', 'f(x^k)', 'd^k', 'a^k', 'x^k+1'])
    def add_row(self, k, xk, fxk, dk, ak, xkp):
        self.table.loc[len(self.table)] = [k, np_str(xk), f_str(np.asscalar(fxk)), np_str(dk), ak, np_str(xkp)]
    def print_latex(self):
        print(self.table.to_latex(index=False))
```

Exact line search algorithm is implemented using Bisection Method. The source used to implement it:

```
def BisectionMethod(f,epsilon, a=-100,b=100) :
    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, eps=0.0000000001):
    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, epsilon=eps)
    return alp_star
```

2 Steepest Descent Method

The steepest descent method is an iterative method that minimizes the given function by using its first derivative. It determines a direction which is the negative of the derivative of the function at the given point. Moving on this direction decreases the function value since it is a descent direction. After finding the direction, the step length, how long to move, is determined by the exact line search which is described above. This process is repeated until the magnitude of the derivative becomes smaller than a given epsilon.

The algorithm is given above, and it is used to find the given 2 functions minimum with 2 different sets of epsilon and initial point.

```
def steepestDescentMethod(f, grad_f, x_0, epsilon):
    xk = np.array(x_0).reshape(2,1)
    k = 0
    stop = False
    output = OutputTable()
    while(stop == False):
        d = - np.transpose(grad_f(xk))
        if(np.linalg.norm(d) < epsilon):
            stop = True
        else:
            a = ExactLineSearch(f,xk,d)
            xkp = xk + a*d
            output.add_row(k, xk, f(xk), d, a, xkp)
            k += 1
            xk = xkp
    output.add_row(k,xk,f(xk),d,None,np.array([]))
    return xk, np.asscalar(f(xk)), output
```

Solution set 1 for first function:

- $x^{(0)}$: [10 10]
- ε_1 : 0.001

Output of the solution set 1:

k	$x^{(k)}$	$f(x^{(k)})$	$d^{(k)}$	$\alpha^{(k)}$	$x^{(k+1)}$
0	[10,10]	2560066	[-1280017, 256002]	6.07647e-06	[2.222,11.556]
1	[2.222,11.556]	-8.80048	[0.325,1.646]	13.2118	[6.51 ,33.306]
2	[6.51 ,33.306]	-27.4356	[-1.333, 0.263]	0.00527333	[6.503,33.307]
3	[6.503,33.307]	-27.4405	[-0. ,-0.001]	-14.4792	[6.506,33.323]
4	[6.506,33.323]	-27.4405	[-0.001,-0.002]	0.271505	[6.505,33.322]
5	[6.505,33.322]	-27.4405	[0.029,-0.008]	0.00494709	[6.506,33.322]
6	[6.506,33.322]	-27.4405	[6.406e-05,-2.249e-03]	0.571153	[6.506,33.321]
7	[6.506,33.321]	-27.4405	[-0.055, 0.009]	0.00505372	[6.505,33.321]
8	[6.505,33.321]	-27.4405	[-0. ,-0.002]	10.0149	[6.502,33.3]
9	[6.502,33.3]	-27.4404	[-0.206, 0.04]	0.00507377	[6.501,33.3]
10	[6.501,33.3]	-27.4405	[-2.454e-05,-5.019e-04]	None	[]

$$x^* = [\ 6.501,33.3\ ]$$
$$f(x^*) = -27.4405$$

Solution set 2 for first function:

- $x^{(0)}$: [-25,-15]
- ε_1 : 0.001

Output of the solution set 2:

k	$x^{(k)}$	$f(x^{(k)})$	$d^{(k)}$	$\alpha^{(k)}$	$x^{(k+1)}$
0	[-25,-15]	146410746	[26620053,-5323998]	8.0031e-07	[-3.696,-19.261]
1	[-3.696,-19.261]	79.6416	[0.816,3.915]	7.81757	[2.684,11.345]
2	[2.684,11.345]	10.9678	[-180.825, 37.691]	0.00267321	[2.201,11.446]
3	[2.201,11.446]	-8.61301	[0.344,1.651]	11.8403	[6.275,30.994]
4	[6.275,30.994]	-25.4153	[-10.661, 2.222]	0.0207447	[6.054,31.04]
5	[6.054,31.04]	-27.2396	[0.036,0.171]	10.6039	[6.433,32.856]
6	[6.433,32.856]	-27.4012	[-3.191, 0.665]	0.00575807	[6.414,32.86]
7	[6.414,32.86]	-27.4331	[0.007,0.033]	10.7422	[6.488,33.214]
8	[6.488,33.214]	-27.439	[-0.7 , 0.145]	0.00516958	[6.484,33.215]
9	[6.484,33.215]	-27.4403	[0.001,0.006]	18.2062	[6.507,33.324]
10	[6.507,33.324]	-27.4404	[-0.243, 0.046]	0.00507676	[6.506,33.324]
11	[6.506,33.324]	-27.4405	[-0. ,-0.002]	12.481	[6.501,33.295]
12	[6.501,33.295]	-27.4404	[-0.21 , 0.041]	0.00507319	[6.5 ,33.295]
13	[6.5 ,33.295]	-27.4406	[-2.536e-05,-1.136e-04]	None	[]

$$x^* = [\ 6.5\ ,33.295]$$
$$f(x^*) = -27.4406$$

Solution set 1 for second function:

- $x^{(0)}$: [2,-4]
- ε_1 : 0.01

Output of the solution set 1:

k	$x^{(k)}$	$f(x^{(k)})$	$d^{(k)}$	$\alpha^{(k)}$	$x^{(k+1)}$
0	[2,-4]	6401	[-6402, 1600]	0.00033167	[-0.123,-3.469]
1	[-0.123,-3.469]	1215.47	[174.172,696.909]	0.00637716	[0.987,0.975]
2	[0.987,0.975]	0.00015976	[0.05 ,-0.012]	0.00107117	[0.987,0.975]
3	[0.987,0.975]	0.000158347	[0.003,0.011]	0.0206554	[0.987,0.975]
4	[0.987,0.975]	0.000156945	[0.049,-0.012]	0.00107097	[0.988,0.975]
5	[0.988,0.975]	0.000155556	[0.003,0.011]	0.0206606	[0.988,0.975]
6	[0.988,0.975]	0.00015418	[0.049,-0.012]	0.00107075	[0.988,0.975]
7	[0.988,0.975]	0.000152815	[0.003,0.011]	0.0206664	[0.988,0.976]
8	[0.988,0.976]	0.000151463	[0.049,-0.012]	0.00107053	[0.988,0.976]
9	[0.988,0.976]	0.000150122	[0.003,0.011]	0.020673	[0.988,0.976]
10	[0.988,0.976]	0.000148793	[0.048,-0.012]	0.00107032	[0.988,0.976]
11	[0.988,0.976]	0.000147476	[0.003,0.011]	0.0206791	[0.988,0.976]
12	[0.988,0.976]	0.000146171	[0.048,-0.012]	0.00107011	[0.988,0.976]
13	[0.988,0.976]	0.000144877	[0.003,0.011]	0.0206858	[0.988,0.976]
14	[0.988,0.976]	0.000143594	[0.047,-0.012]	0.0010699	[0.988,0.976]
15	[0.988,0.976]	0.000142323	[0.003,0.011]	0.020689	[0.988,0.976]
16	[0.988,0.976]	0.000141063	[0.047,-0.012]	0.0010697	[0.988,0.976]
17	[0.988,0.976]	0.000139814	[0.003,0.011]	0.0206966	[0.988,0.977]
18	[0.988,0.977]	0.000138576	[0.046,-0.012]	0.00106949	[0.988,0.977]
19	[0.988,0.977]	0.000137349	[0.003,0.011]	0.0207016	[0.988,0.977]
20	[0.988,0.977]	0.000136133	[0.046,-0.012]	0.00106929	[0.988,0.977]
21	[0.988,0.977]	0.000134927	[0.003,0.01]	0.0207037	[0.988,0.977]
22	[0.988,0.977]	0.000133732	[0.046,-0.011]	0.0010691	[0.988,0.977]
23	[0.988,0.977]	0.000132548	[0.003,0.01]	0.0207032	[0.989,0.977]
24	[0.989,0.977]	0.000131375	[0.045,-0.011]	0.00106891	[0.989,0.977]
25	[0.989,0.977]	0.000130212	[0.003,0.01]	0.0207091	[0.989,0.977]
26	[0.989,0.977]	0.000129059	[0.045,-0.011]	0.00106872	[0.989,0.977]

27	[0.989,0.977]	0.000127916	[0.003,0.01]	0.0207095	[0.989,0.978]
28	[0.989,0.978]	0.000126784	[0.044,-0.011]	0.00106854	[0.989,0.978]
29	[0.989,0.978]	0.000125662	[0.003,0.01]	0.0207139	[0.989,0.978]
30	[0.989,0.978]	0.000124549	[0.044,-0.011]	0.00106835	[0.989,0.978]
31	[0.989,0.978]	0.000123447	[0.002,0.01]	0.0207186	[0.989,0.978]
32	[0.989,0.978]	0.000122354	[0.044,-0.011]	0.00106814	[0.989,0.978]
33	[0.989,0.978]	0.000121271	[0.002,0.01]	0.020731	[0.989,0.978]
34	[0.989,0.978]	0.000120197	[0.043,-0.011]	0.00106794	[0.989,0.978]
35	[0.989,0.978]	0.000119132	[0.002,0.01]	0.0207364	[0.989,0.978]
36	[0.989,0.978]	0.000118077	[0.043,-0.011]	0.00106776	[0.989,0.978]
37	[0.989,0.978]	0.000117031	[0.002,0.01]	None	[]

$$x^* = [0.989,0.978]$$
$$f(x^*) = 0.000117031$$

Solution set 2 for second function:

- $x^{(0)}$: [-2. ,-3.5]
- ε_1 : 0.002

Output of the solution set 2:

k	$x^{(k)}$	$f(x^{(k)})$	$d^{(k)}$	$\alpha^{(k)}$	$x^{(k+1)}$
0	[-2. ,-3.5]	5634.000000	[6006.,1500.]	0.000354036	[0.126,-2.969]
1	[0.126,-2.969]	891.730769	[-149.096, 596.982]	0.00591938	[-0.756, 0.565]
2	[-0.756, 0.565]	3.089270	[5.645,1.41]	0.311481	[1.002,1.004]
3	[1.002,1.004]	0.000004	[0.001,-0.002]	0.00235886	[1.002,1.004]
4	[1.002,1.004]	0.000004	[-0.003,-0.001]	0.00172039	[1.002,1.004]
5	[1.002,1.004]	0.000004	[0.001,-0.002]	0.00235916	[1.002,1.004]
6	[1.002,1.004]	0.000004	[-0.003,-0.001]	0.00171994	[1.002,1.004]
7	[1.002,1.004]	0.000004	[0.001,-0.002]	0.0023609	[1.002,1.004]
8	[1.002,1.004]	0.000004	[-0.003,-0.001]	0.00172183	[1.002,1.004]
9	[1.002,1.004]	0.000004	[0.001,-0.002]	0.00235275	[1.002,1.004]
...	...	...	...	...	...
164	[1.002,1.003]	0.000003	[-0.002,-0.001]	0.001718	[1.002,1.003]
165	[1.002,1.003]	0.000003	[0. ,-0.002]	0.0023663	[1.002,1.003]
166	[1.002,1.003]	0.000003	[-0.002,-0.001]	0.00171834	[1.002,1.003]
167	[1.002,1.003]	0.000003	[0. ,-0.002]	0.00236276	[1.002,1.003]
168	[1.002,1.003]	0.000003	[-0.002,-0.001]	0.00171951	[1.002,1.003]
169	[1.002,1.003]	0.000003	[0. ,-0.002]	0.00236421	[1.002,1.003]
170	[1.002,1.003]	0.000003	[-0.002,-0.001]	0.00171868	[1.002,1.003]
171	[1.002,1.003]	0.000003	[0. ,-0.002]	0.00236575	[1.002,1.003]
172	[1.002,1.003]	0.000003	[-0.002,-0.001]	0.00171961	[1.002,1.003]
173	[1.002,1.003]	0.000003	[0. ,-0.002]	None	[]

$$x^* = [1.002, 1.003]$$
$$f(x^*) = 0.000003$$

Conclusion:
The steepest descent direction method works much faster in the first function since the algorithm becomes very slower in the second function as approaching to the minimum due to the banana shaped valley structure of the second function. The algorithm stacks in the region and starts to zigzagging as seen the solution sets. It gets so close to the minimum point within a few iteration and starts to zigzagging in order to reach the minimum.

3 Newton's Method

In Newton's Method, the direction is calculated by multiplying the inverse of the Hessian matrix with the gradient vector.The inversion of a matrix is a costly operation and direction is not necessarily a descent direction. However, if we select the initial point in the convex area, Hessian will be positive definite and direction will be a descent direction. Therefore, Newton's Method converges to local min.

```
def NewtonsMethod(x_0,epsilon,f,grad_f,Hessian_f):
    xk = np.array(x_0).reshape(2,1)
    k=0
    output = OutputTable()
    while(True):
        d_k=-np.linalg.inv(Hessian_f(xk))@np.transpose(grad_f(xk))
        alpha_k=ExactLineSearch(f,xk,d_k)
        xkp=xk+alpha_k*d_k
        if(np.linalg.norm(grad_f(xk)) < epsilon):
            break
        output.add_row(k, xk, f(xk), d_k, alpha_k, xkp)
        xk = xkp
        k += 1
    output.add_row(k,xk,f(xk),d_k,None,np.array([]))
    return xk, np.asscalar(f(xk)), output
```

Solution set 1 for first function:

- $x^{(0)} : (-5, 1)$
- $\varepsilon_1 : 0.01$

Output of the solution set 1:

k	$x^{(k)}$	$f(x^{(k)})$	$d^{(k)}$	$\alpha^{(k)}$	$x^{(k+1)}$
0	[-5, 1]	457030.0000	[11.5 ,48.834]	2.724279	[26.329,134.036]
1	[26.329,134.036]	394.8089	[-19.829,-99.914]	1.014391	[6.215,32.685]
2	[6.215,32.685]	-22.6456	[0.285,0.954]	1.667151	[6.69 ,34.275]
3	[6.69 ,34.275]	-27.4010	[-0.19,-0.98]	1.002499	[6.5 ,33.292]
4	[6.5 ,33.292]	-27.4405	[0. ,0.001]	3.037450	[6.501,33.297]
5	[6.501,33.297]	-27.4405	[-0.001,-0.003]	1.336735	[6.5 ,33.293]
6	[6.5 ,33.293]	-27.4405	[0. ,0.001]	-1.757936	[6.499,33.291]
7	[6.499,33.291]	-27.4405	[0.001,0.003]	1.214216	[6.5 ,33.294]
8	[6.5 ,33.294]	-27.4406	[-0. ,-0.001]	6.237393	[6.499,33.291]
9	[6.499,33.291]	-27.4405	[0.001,0.003]	-0.012258	[6.499,33.291]
10	[6.499,33.291]	-27.4405	[0.001,0.003]	1.763892	[6.501,33.296]
11	[6.501,33.296]	-27.4405	[-0.001,-0.002]	2.418095	[6.499,33.29]
12	[6.499,33.29]	-27.4405	[0.001,0.003]	0.942628	[6.5 ,33.294]
13	[6.5 ,33.294]	-27.4406	[6.355e-05,1.848e-04]	NaN	[]

$$x^* = (6.5, 33.294)$$
$$f(x^*) = -27.4406$$

Solution set 2 for first function:

- $x^{(0)}$: $(-25, 75)$
- ε_1 : 0.001

Output of the solution set 2:

k	$x^{(k)}$	$f(x^{(k)})$	$d^{(k)}$	$\alpha^{(k)}$	$x^{(k+1)}$
0	[-25, 75]	1600000566.0000	[31.5 ,90.833]	2.962919	[68.332,344.132]
1	[68.332,344.132]	3829.3422	[-61.832,-309.956]	1.001747	[6.392,33.634]
2	[6.392,33.634]	-21.7344	[0.108,0.042]	1.756464	[6.582,33.707]
3	[6.582,33.707]	-27.4338	[-0.082,-0.413]	1.002057	[6.5 ,33.293]
4	[6.5 ,33.293]	-27.4406	[0. ,0.001]	49.983209	[6.508,33.334]
5	[6.508,33.334]	-27.4405	[-0.008,-0.04]	0.732422	[6.502,33.304]
6	[6.502,33.304]	-27.4405	[-0.002,-0.011]	2.272174	[6.497,33.28]
7	[6.497,33.28]	-27.4405	[0.003,0.014]	4.400717	[6.51,33.34]
8	[6.51,33.34]	-27.4405	[-0.01 ,-0.047]	0.975800	[6.5 ,33.295]
9	[6.5 ,33.295]	-27.4406	[-0. ,-0.001]	46.081543	[6.49 ,33.243]
10	[6.49 ,33.243]	-27.4404	[0.01 ,0.051]	0.975901	[6.5 ,33.292]
11	[6.5 ,33.292]	-27.4406	[0. ,0.001]	99.694440	[6.525,33.415]
12	[6.525,33.415]	-27.4399	[-0.025,-0.121]	1.003612	[6.5 ,33.293]
13	[6.5 ,33.293]	-27.4406	[8.932e-05,4.290e-04]	NaN	[]

$$x^* = (6.5, 33.293)$$
$$f(x^*) = -27.4406$$

Solution set 1 for second function:

- $x^{(0)} : (-2, 4)$
- $\varepsilon_1 : 0.01$

Output of the solution set 1:

k	$x^{(k)}$	$f(x^{(k)})$	$d^{(k)}$	$\alpha^{(k)}$	$x^{(k+1)}$
0	[-2, 4]	38434.0000	[8.5 ,37.834]	2.507532	[19.314,98.87]
1	[19.314,98.87]	161.3480	[-12.814,-64.805]	1.028494	[6.135,32.219]
2	[6.135,32.219]	-23.5204	[0.365,1.381]	1.587915	[6.715,34.411]
3	[6.715,34.411]	-27.3868	[-0.215,-1.115]	1.006787	[6.499,33.288]
4	[6.499,33.288]	-27.4405	[0.001,0.005]	1.616125	[6.501,33.297]
5	[6.501,33.297]	-27.4405	[-0.001,-0.003]	1.814999	[6.499,33.291]
6	[6.499,33.291]	-27.4405	[0.001,0.003]	1.558940	[6.5 ,33.295]
7	[6.5 ,33.295]	-27.4405	[-0. ,-0.001]	10.962735	[6.496,33.279]
8	[6.496,33.279]	-27.4404	[0.004,0.015]	1.177985	[6.501,33.296]
9	[6.501,33.296]	-27.4405	[-0.001,-0.003]	6.449573	[6.496,33.279]
10	[6.496,33.279]	-27.4404	[0.004,0.015]	1.055527	[6.5 ,33.295]
11	[6.5 ,33.295]	-27.4406	[-0. ,-0.001]	83.594513	[6.482,33.224]
12	[6.482,33.224]	-27.4386	[0.018,0.07]	1.023983	[6.5 ,33.296]
13	[6.5 ,33.296]	-27.4406	[-0. ,-0.002]	NaN	[]

$$x^* = (6.5, 33.296)$$
$$f(x^*) = -27.4406$$

Solution set 2 for second function:

- $x^{(0)} : (-10, 1)$
- $\varepsilon_1 : 0.001$

Output of the solution set 2:

k	$x^{(k)}$	$f(x^{(k)})$	$d^{(k)}$	$\alpha^{(k)}$	$x^{(k+1)}$
0	[-10, 1]	6765345.0000	[16.5,65.5]	2.853838	[37.088,187.927]
1	[37.088,187.927]	942.5554	[-30.588,-153.743]	1.007234	[6.279,33.071]
2	[6.279,33.071]	-21.6320	[0.221,0.606]	1.731007	[6.662,34.121]
3	[6.662,34.121]	-27.4131	[-0.162,-0.827]	1.000977	[6.5 ,33.293]
4	[6.5 ,33.293]	-27.4406	[0.,0.]	-3.326661	[6.499,33.292]
5	[6.499,33.292]	-27.4405	[0.001,0.002]	0.976539	[6.5 ,33.294]
6	[6.5 ,33.294]	-27.4406	[1.604e-05,3.586e-05]	71.725821	[6.501,33.296]
7	[6.501,33.296]	-27.4405	[-0.001,-0.003]	1.049374	[6.5 ,33.294]
8	[6.5 ,33.294]	-27.4406	[5.600e-05,1.122e-04]	43.741345	[6.502,33.298]
9	[6.502,33.298]	-27.4404	[-0.002,-0.005]	1.173416	[6.5 ,33.293]
...	...	...	...	...	...
250	[6.5 ,33.293]	-27.4405	[-8.076e-11, 9.509e-04]	7.812506	[6.5,33.3]
251	[6.5,33.3]	-27.4404	[5.501e-10,-6.427e-03]	1.120387	[6.5 ,33.293]
252	[6.5 ,33.293]	-27.4405	[-6.623e-11, 7.220e-04]	9.390618	[6.5,33.3]
253	[6.5,33.3]	-27.4404	[5.557e-10,-6.013e-03]	1.089574	[6.5 ,33.293]
254	[6.5 ,33.293]	-27.4405	[-4.978e-11, 4.931e-04]	11.594878	[6.5 ,33.299]
255	[6.5 ,33.299]	-27.4404	[5.274e-10,-5.191e-03]	1.107518	[6.5 ,33.293]
256	[6.5 ,33.293]	-27.4405	[-5.670e-11, 5.244e-04]	12.616582	[6.5,33.3]
257	[6.5,33.3]	-27.4404	[6.587e-10,-6.046e-03]	0.976179	[6.5 ,33.294]
258	[6.5 ,33.294]	-27.4406	[1.569e-11,-1.902e-04]	-3.186058	[6.5 ,33.294]
259	[6.5 ,33.294]	-27.4405	[6.568e-11,-7.956e-04]	4.044482	[6.5 ,33.291]
260	[6.5 ,33.291]	-27.4405	[-2.000e-10, 2.429e-03]	1.292392	[6.5 ,33.294]
261	[6.5 ,33.294]	-27.4405	[5.847e-11,-7.170e-04]	8.548274	[6.5 ,33.288]
262	[6.5 ,33.288]	-27.4404	[-4.413e-10, 5.448e-03]	0.988770	[6.5 ,33.294]
263	[6.5 ,33.294]	-27.4406	[-4.956e-12, 2.396e-05]	NaN	[]

$$x^* = (6.5, 33.294)$$
$$f(x^*) = -27.4406$$

Conclusion:
Newton's Method works well on these two functions and it was able to find the global minimums with the appropriate starting points.However, finding the minimum of the second function could be harder if we increase the precision or select inappropriate initial point.

4 DFP Method

Finding the Newton's direction requires inversion of the Hessian matrix and inversion of a matrix is a costly operation. Therefore, DFP offers an recurrent estimation for the inverse of the Hessian using the change information in x and the gradient of f(x). It guarantees that the estimated Hessian is symmetric and positive definite at every step.

```
def DFP(f, grad_f, x_0, epsilon):
    xk = np.array(x_0).reshape(2,1)
    k = 0
    H = np.identity(len(x_0))
    stop = False
    output = OutputTable()
    while(stop == False):
        d = -H @ np.transpose(grad_f(xk))
        if(np.linalg.norm(d) < epsilon):
            stop = True
        else:
            a = ExactLineSearch(f,xk,d)
            xkp = xk + a*d
            p = xkp - xk
            q = np.transpose(grad_f(xkp)) - np.transpose(grad_f(xk))
            A = (p @ np.transpose(p)) / (p.transpose() @ q)
            B = - (H @ q @ np.transpose( H @ q)) / (q.transpose() @ H @ q)
            Hkp = H + A + B
            output.add_row(k, xk, f(xk), d, a, xkp)
            k += 1
            xk = xkp
            H = Hkp
    output.add_row(k,xk,f(xk),d,None,np.array([]))
    return xk, np.asscalar(f(xk)), output
```

Solution set 1 for first function:

- $x^{(0)}$: $(0,0)$
- ε_1 : 0.001

Output of the solution set 1:

k	$x^{(k)}$	$f(x^{(k)})$	$d^{(k)}$	$\alpha^{(k)}$	$x^{(k+1)}$
0	[0,0]	16.0000	[3.,2.]	0.047375	[0.142,0.095]
1	[0.142,0.095]	15.5482	[0.467,2.478]	12.343352	[5.908,30.676]
2	[5.908,30.676]	-26.4979	[0.514,1.701]	0.465735	[6.147,31.468]
3	[6.147,31.468]	-27.3030	[0.364,1.861]	0.991801	[6.508,33.314]
4	[6.508,33.314]	-27.4391	[-0.005,-0.012]	1.431509	[6.501,33.297]
5	[6.501,33.297]	-27.4406	[-0.001,-0.003]	29.368085	[6.484,33.211]
6	[6.484,33.211]	-27.4403	[0.016,0.082]	1.074982	[6.501,33.3]
7	[6.501,33.3]	-27.4405	[-0.001,-0.006]	9.378767	[6.49 ,33.242]
8	[6.49 ,33.242]	-27.4404	[0.01 ,0.052]	1.080873	[6.501,33.298]
9	[6.501,33.298]	-27.4406	[-0.001,-0.004]	34.796338	[6.472,33.152]
10	[6.472,33.152]	-27.4397	[0.028,0.141]	1.005456	[6.5 ,33.294]
11	[6.5 ,33.294]	-27.4406	[-0. ,-0.001]	NaN	[]

$$x^* = (6.5, 33.294)$$
$$f(x^*) = -27.4406$$

Solution set 2 for first function:

- $x^{(0)}$:$(5,5)$
- ε_1 :0.0001

Output of the solution set 2:

k	$x^{(k)}$	$f(x^{(k)})$	$d^{(k)}$	$\alpha^{(k)}$	$x^{(k+1)}$
0	[5,5]	160016.0000	[-160007., 32002.]	0.000024	[1.195,5.761]
1	[1.195,5.761]	2.3231	[0.408,2.04]	13.001370	[6.5 ,32.287]
2	[6.5 ,32.287]	-25.8185	[-7.411e-05, 8.897e-04]	100.000000	[6.493,32.376]
3	[6.493,32.376]	-26.0726	[-0.391, 4.747]	0.131527	[6.441,33.]
4	[6.441,33.]	-27.4371	[0. ,0.002]	74.993849	[6.476,33.175]
5	[6.476,33.175]	-27.4400	[0.024,0.118]	0.995020	[6.5 ,33.293]
6	[6.5 ,33.293]	-27.4405	[7.192e-05,1.557e-03]	1.440430	[6.5 ,33.295]
7	[6.5 ,33.295]	-27.4405	[-0. ,-0.002]	-0.553897	[6.5 ,33.296]
8	[6.5 ,33.296]	-27.4405	[-0. ,-0.002]	6.250003	[6.499,33.281]
9	[6.499,33.281]	-27.4403	[0.001,0.012]	1.055908	[6.5 ,33.294]
10	[6.5 ,33.294]	-27.4406	[-5.357e-05,-6.734e-04]	12.619487	[6.499,33.286]
11	[6.499,33.286]	-27.4405	[0.001,0.008]	1.073317	[6.5 ,33.294]
12	[6.5 ,33.294]	-27.4406	[-4.564e-05,-5.734e-04]	-13.088995	[6.501,33.302]
13	[6.501,33.302]	-27.4405	[-0.001,-0.008]	1.018119	[6.5 ,33.294]
14	[6.5 ,33.294]	-27.4406	[1.165e-05,1.437e-04]	73.694942	[6.501,33.304]
15	[6.501,33.304]	-27.4404	[-0.001,-0.01]	0.973478	[6.5 ,33.294]
16	[6.5 ,33.294]	-27.4406	[-2.247e-05,-2.751e-04]	71.475835	[6.498,33.274]
17	[6.498,33.274]	-27.4401	[0.002,0.019]	1.013158	[6.5 ,33.294]
18	[6.5 ,33.294]	-27.4406	[-2.083e-05,-2.549e-04]	-12.121595	[6.5 ,33.297]
19	[6.5 ,33.297]	-27.4405	[-0. ,-0.003]	1.321167	[6.5 ,33.293]
20	[6.5 ,33.293]	-27.4405	[8.780e-05,1.071e-03]	26.367188	[6.502,33.321]
21	[6.502,33.321]	-27.4396	[-0.002,-0.027]	1.000023	[6.5 ,33.294]
22	[6.5 ,33.294]	-27.4406	[4.333e-08,1.297e-05]	NaN	[]

$$x^* = (6.5, 33.294)$$
$$f(x^*) = -27.4406$$

Solution set 1 for second function:

- $x^{(0)}$: $(1.2, 1.6)$
- ε_1 : 10^{-9}

Output of the solution set 1:

k	$x^{(k)}$	$f(x^{(k)})$	$d^{(k)}$	$\alpha^{(k)}$	$x^{(k+1)}$
0	[1.2,1.6]	2.6000	[76.4,-32.]	0.000725	[1.255,1.577]
1	[1.255,1.577]	0.0653	[-0.073,-0.175]	2.067016	[1.105,1.216]
2	[1.105,1.216]	0.0141	[-0.012,-0.024]	10.274499	[0.983,0.97]
3	[0.983,0.97]	0.0015	[-0.025,-0.056]	0.352653	[0.974,0.95]
4	[0.974,0.95]	0.0007	[0.008,0.015]	3.232717	[1. ,0.999]
5	[1. ,0.999]	0.0000	[0. ,0.001]	1.270673	[1.,1.]
6	[1.,1.]	0.0000	[0.,0.]	0.801349	[1.,1.]
7	[1.,1.]	0.0000	[1.576e-07,3.587e-07]	1.003928	[1.,1.]
8	[1.,1.]	0.0000	[4.827e-10,9.773e-10]	0.999220	[1.,1.]
9	[1.,1.]	0.0000	[-1.793e-15, 1.973e-14]	NaN	[]

$$x^* = (1,1)$$
$$f(x^*) = 0.0000$$

Solution set 2 for second function:

- $x^{(0)} : (-2,-3)$
- $\varepsilon_1 : 10^{-5}$

Output of the solution set 2:

k	$x^{(k)}$	$f(x^{(k)})$	$d^{(k)}$	$\alpha^{(k)}$	$x^{(k+1)}$
0	[-2,-3]	4909.0000	[5606.,1400.]	0.000379	[0.127,-2.469]
1	[0.127,-2.469]	618.2588	[-79.43 ,504.031]	0.078003	[-6.069,36.847]
2	[-6.069,36.847]	49.9878	[-0.006,-0.001]	0.090790	[-6.07 ,36.847]
3	[-6.07 ,36.847]	49.9836	[0.004,-0.043]	95.312548	[-5.736,32.793]
4	[-5.736,32.793]	46.4856	[-0.01 , 0.129]	2.708229	[-5.764,33.144]
5	[-5.764,33.144]	46.4092	[0.002,-0.027]	100.000000	[-5.523,30.463]
6	[-5.523,30.463]	42.6737	[0.398,-4.424]	0.611933	[-5.279,27.755]
7	[-5.279,27.755]	40.6931	[0.02,-0.21]	16.747784	[-4.937,24.235]
8	[-4.937,24.235]	37.2076	[0.005,-0.039]	16.180277	[-4.856,23.611]
9	[-4.856,23.611]	34.3796	[0.027,-0.263]	10.101984	[-4.586,20.954]
10	[-4.586,20.954]	31.8773	[0.03 ,-0.277]	5.146031	[-4.433,19.529]
11	[-4.433,19.529]	31.0832	[0.009,-0.077]	62.915015	[-3.844,14.689]
12	[-3.844,14.689]	24.2917	[-0.002, 0.031]	5.764711	[-3.856,14.866]
13	[-3.856,14.866]	23.5904	[0.022,-0.169]	13.189697	[-3.567,12.639]
14	[-3.567,12.639]	21.5425	[0.017,-0.12]	13.964080	[-3.328,10.96]

15	[-3.328,10.96]	20.0618	[0.005,-0.028]	31.935715	[-3.168,10.082]
16	[-3.168,10.082]	17.6041	[0.016,-0.107]	13.851727	[-2.941, 8.595]
17	[-2.941, 8.595]	15.8586	[0.024,-0.147]	4.756928	[-2.826, 7.895]
18	[-2.826, 7.895]	15.4805	[0.009,-0.045]	46.203853	[-2.429, 5.813]
19	[-2.429, 5.813]	12.5402	[-0. , 0.009]	11.941529	[-2.434, 5.921]
20	[-2.434, 5.921]	11.7897	[0.017,-0.08]	15.309067	[-2.18 , 4.689]
21	[-2.18 , 4.689]	10.5201	[0.014,-0.06]	13.922143	[-1.985, 3.85]
22	[-1.985, 3.85]	9.7249	[0.004,-0.011]	35.485838	[-1.847, 3.449]
23	[-1.847, 3.449]	8.2395	[0.013,-0.051]	15.194702	[-1.649, 2.675]
24	[-1.649, 2.675]	7.2102	[0.019,-0.067]	5.407696	[-1.545, 2.315]
25	[-1.545, 2.315]	6.9740	[0.007,-0.018]	53.242350	[-1.181, 1.332]
26	[-1.181, 1.332]	5.1607	[-0.001, 0.008]	10.456657	[-1.19 , 1.413]
27	[-1.19 , 1.413]	4.7975	[0.014,-0.033]	15.479920	[-0.97 , 0.895]
28	[-0.97 , 0.895]	4.1037	[0.012,-0.022]	15.515158	[-0.791, 0.56]
29	[-0.791, 0.56]	3.6231	[0.003,-0.001]	27.198888	[-0.709, 0.521]
30	[-0.709, 0.521]	2.9540	[0.013,-0.02]	12.990690	[-0.542, 0.261]
31	[-0.542, 0.261]	2.4836	[0.016,-0.019]	5.994603	[-0.443, 0.146]
32	[-0.443, 0.146]	2.3424	[0.006,-0.003]	65.283202	[-0.083,-0.028]
33	[-0.083,-0.028]	1.2906	[-0.003, 0.007]	4.849439	[-0.095, 0.005]
34	[-0.095, 0.005]	1.2010	[0.014,-0.002]	12.403178	[0.079,-0.023]
35	[0.079,-0.023]	0.9335	[0.01 ,0.003]	17.149069	[0.251,0.027]
36	[0.251,0.027]	0.6884	[0.002,0.004]	12.581271	[0.273,0.079]
37	[0.273,0.079]	0.5300	[0.015,0.008]	8.789063	[0.405,0.145]
38	[0.405,0.145]	0.3912	[0.014,0.011]	6.847523	[0.501,0.223]
39	[0.501,0.223]	0.3246	[0.004,0.006]	19.628906	[0.578,0.345]
40	[0.578,0.345]	0.1883	[0.013,0.014]	6.893705	[0.67 ,0.439]
41	[0.67 ,0.439]	0.1196	[0.016,0.021]	3.562737	[0.729,0.514]
42	[0.729,0.514]	0.1011	[0.005,0.009]	42.744348	[0.958,0.915]
43	[0.958,0.915]	0.0024	[-0.012,-0.015]	56.847522	[0.274,0.086]
44	[0.274,0.086]	0.5395	[-0.014,-0.019]	0.131214	[0.272,0.083]
45	[0.272,0.083]	0.5393	[0.002,0.001]	79.496779	[0.443,0.179]
46	[0.443,0.179]	0.3397	[0.003,0.003]	21.864283	[0.511,0.236]
47	[0.511,0.236]	0.3039	[0.001,0.001]	100.000000	[0.614,0.378]
48	[0.614,0.378]	0.1487	[0.042,0.056]	4.148575	[0.788,0.611]
49	[0.788,0.611]	0.0537	[0.003,0.005]	49.529805	[0.925,0.849]
50	[0.925,0.849]	0.0098	[-0.001,-0.]	3.609610	[0.921,0.847]

51	[0.921,0.847]	0.0063	[0.008,0.015]	6.681827	[0.975,0.949]
52	[0.975,0.949]	0.0013	[0.003,0.008]	3.379421	[0.987,0.974]
53	[0.987,0.974]	0.0002	[0.009,0.017]	1.273894	[0.998,0.996]
54	[0.998,0.996]	0.0000	[0.001,0.003]	1.472127	[1.,1.]
55	[1.,1.]	0.0000	[5.529e-05,1.062e-04]	0.850425	[1.,1.]
56	[1.,1.]	0.0000	[6.493e-07,1.375e-06]	NaN	[]

$$x^* = (1, 1)$$
$$f(x^*) = 0.0000$$

Conclusion:

DFB works well on these two functions and it was able to find the global minimums with the appropriate starting points. Finding the minimum of the second function was harder so a selection procedure for the inital point required.

5 BFGS Method

The only difference between DFP and BFGS method is how A and B matrices are formed to determine the H^{k+1}.

The Python code to implement BFGS algorithm:

```
def BFGS(f, grad_f, x_0, epsilon, line_search_tol = 0.0000001):
    xk = np.array(x_0).reshape(2,1)
    k = 0
    H = np.identity(len(x_0))
    stop = False
    output = OutputTable()
    while(stop == False):
        d = -H @ np.transpose(grad_f(xk))
        if(np.linalg.norm(d) < epsilon):
            stop = True
        if(k == -1):
            break
        else:
            a = ExactLineSearch(f,xk,d, line_search_tol)
            xkp = xk + a*d
            p = xkp - xk
            q = np.transpose(grad_f(xkp)) - np.transpose(grad_f(xk))
            A = ((1+ np.transpose(q) @ H @ q) / (np.transpose(q) @ p)) * (p @ np.
                transpose(p)) / (np.transpose(p) @ q)
            B = - (p @ np.transpose(q) @ H + H @ q @ np.transpose(p)) / (np.transpose(q)
                @ p)
            Hkp = H + A + B
            output.add_row(k, xk, f(xk), d, a, xkp)
            k += 1
```

```python
            xk = xkp
            H = Hkp
    output.add_row(k,xk,f(xk),d,None,np.array([]))
    return xk, np.asscalar(f(xk)), output
```

Solution set 1 for first function:

- $x^{(0)}$: (0, 0)
- ε_1 : 0.01

Output of the solution set 1:

k	$x^{(k)}$	$f(x^{(k)})$	$d^{(k)}$	$\alpha^{(k)}$	$x^{(k+1)}$
0	[0,0]	16.0000	[3.,2.]	0.047375	[0.142,0.095]
1	[0.142,0.095]	15.5482	[0.914,4.848]	6.308132	[5.908,30.676]
2	[5.908,30.676]	-26.4980	[2.239,7.413]	0.106881	[6.147,31.468]
3	[6.147,31.468]	-27.3030	[0.005,0.025]	72.460937	[6.508,33.314]
4	[6.508,33.314]	-27.4391	[-0.003,-0.007]	2.636336	[6.501,33.297]
5	[6.501,33.297]	-27.4406	[-0.003,-0.007]	NaN	[]

$$x^* = (6.50060599, 33.29678992)$$
$$f(x^*) = \text{-}27.44055$$

Algorithm successfully found a local minimum point.

Solution set 2 for first function:

- $x^{(0)}$: (10, 10)
- ε_1 : 0.01

Output of the solution set 2:

k	$x^{(k)}$	$f(x^{(k)})$	$d^{(k)}$	$\alpha^{(k)}$	$x^{(k+1)}$
0	[10,10]	2560066.0000	[-1280017., 256002.]	0.000006	[2.311,11.538]
1	[2.311,11.538]	-8.6681	[0.322,1.611]	13.000301	[6.5 ,32.484]
2	[6.5 ,32.484]	-26.2171	[-2.837e-09,-1.415e-08]	-17.036686	[6.5 ,32.484]
3	[6.5 ,32.484]	-26.2171	[-2.837e-09,-1.415e-08]	NaN	[]

$$x^* = (6.50006, 32.48387)$$
$$f(x^*) = \text{-}26.21713987$$

The solution is very close to the one obtained with first solution set.

Solution set 1 for second function:

- $x^{(0)}$: (0, 0)
- ε_1 : 0.01

Output of the solution set 1:

k	$x^{(k)}$	$f(x^{(k)})$	$d^{(k)}$	$\alpha^{(k)}$	$x^{(k+1)}$
0	[0,0]	1.0000	[2.,0.]	0.080631	[0.161,0.]
1	[0.161,0.]	0.7711	[13.526, 5.201]	0.009728	[0.293,0.051]
2	[0.293,0.051]	0.6237	[0.238,0.351]	3.563826	[1.141,1.303]
3	[1.141,1.303]	0.0201	[0.1 ,0.044]	0.005316	[1.141,1.303]
4	[1.141,1.303]	0.0200	[-0.001,-0.003]	40.222919	[1.081,1.164]
5	[1.081,1.164]	0.0092	[-0.001,-0.003]	NaN	[]

$$x^* = (1.08132106, 1.16412038)$$
$$f(x^*) = 0.00924978$$

Algorithm successfully converged to a local minimum point.

Solution set 2 for second function:

- $x^{(0)}$: (10, 10)
- ε_1 : 0.001

Output of the solution set 2:

k	$x^{(k)}$	$f(x^{(k)})$	$d^{(k)}$	$\alpha^{(k)}$	$x^{(k+1)}$
0	[10,10]	810081.0000	[-360018., 18000.]	0.000019	[3.215,10.339]
1	[3.215,10.339]	4.9073	[-0.024,-0.483]	0.010887	[3.215,10.334]
2	[3.215,10.334]	4.9061	[8.995e-08,1.800e-06]	81.190871	[3.215,10.334]
3	[3.215,10.334]	4.9061	[8.995e-08,1.800e-06]	NaN	[]

$$x^* = (3.21490915, 10.33410792)$$
$$f(x^*) = 4.90605754$$

Algorithm found a worse solution than the previous one. When the initial point is given as (10, 10), Bisection Search method's and BFGS method's epsilon values (tolerances) needed to be decreased. Otherwise algorithm finds way worse solutions.

Conclusion:

The performance of BFGS method is good at high precision. It can generally find the solution in few steps. However, DFP's performance seems to be better than BFGS.

6 Appendix

The complete source code:

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


# %%
x1 = Symbol("x1")
x2 = Symbol("x2")

func1 = (5*x1 - x2)**4 + (x1 - 2)**2 + x1 - 2*x2 + 12
func2 = 100*(x2 - x1**2)**2 + (1 - x1)**2


f1 = lambdify([[x1,x2]], func1, "numpy")
f2 = lambdify([[x1,x2]], func2, "numpy")

gf1 = lambdify([[x1,x2]], func1.diff([[x1, x2]]), "numpy")
gf2 = lambdify([[x1,x2]], func2.diff([[x1, x2]]), "numpy")

grad_f1 = lambda x_arr : np.array(gf1(x_arr)).reshape(1,2)
grad_f2 = lambda x_arr : np.array(gf2(x_arr)).reshape(1,2)

hf1 = lambdify([[x1,x2]], (func1.diff([[x1, x2]])).diff([[x1, x2]]), "numpy")
hf2 = lambdify([[x1,x2]], (func2.diff([[x1, x2]])).diff([[x1, x2]]), "numpy")

hess_f1= lambda x_arr : np.array(hf1(np.array(x_arr).reshape(2,)))
hess_f2= lambda x_arr : np.array(hf2(np.array(x_arr).reshape(2,)))


# %%
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(0,3,0.01)
y = np.arange(0,3,0.01)
X,Y = meshgrid(x, y) # grid of point
Z = f2([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.savefig("graph.png")
plt.show()

# %% [markdown]
# ### Useful Functions

# %%
np_str = lambda x_k : np.array2string(x_k.reshape(len(x_k)), precision=3, separator=','
    )

f_str = lambda x : "{0:.4f}".format(x)


# %%
class OutputTable:
    def __init__(self):
        self.table = pd.DataFrame([],columns=['k', 'x^k', 'f(x^k)', 'd^k', 'a^k', 'x^k+1
            '])
    def add_row(self, k, xk, fxk, dk, ak, xkp):
        self.table.loc[len(self.table)] = [k, np_str(xk), f_str(np.asscalar(fxk)),
            np_str(dk), ak, np_str(xkp)]
    def print_latex(self):
        print(self.table.to_latex(index=False))

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

# %%
def BisectionMethod(f,epsilon, a=-100,b=100) :
    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, eps=0.0000000001):
    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, epsilon=eps)
    return alp_star

# %% [markdown]
# ## Steepest Descent Method

# %%
def steepestDescentMethod(f, grad_f, x_0, epsilon):
    xk = np.array(x_0).reshape(2,1)
    k = 0
    stop = False
    output = OutputTable()
    while(stop == False):
        d = - np.transpose(grad_f(xk))
        if(np.linalg.norm(d) < epsilon):
            stop = True
        else:
            a = ExactLineSearch(f,xk,d)
            xkp = xk + a*d
            output.add_row(k, xk, f(xk), d, a, xkp)
            k += 1
            xk = xkp
    output.add_row(k,xk,f(xk),d,None,np.array([]))
    return xk, np.asscalar(f(xk)), output


# %%
xs1, fs1, outputs1 = steepestDescentMethod(f1, grad_f1, [10,10], 0.001)
xs1, fs1


# %%
print(outputs1.table.to_latex(index=False))


# %%
xs2, fs2, outputs2 = steepestDescentMethod(f1, grad_f1, [-25,-15], 0.001)
xs2, fs2


# %%
print(outputs2.table.to_latex(index=False))


# %%
xs3, fs3, outputs3 = steepestDescentMethod(f2, grad_f2, [2,-4], 0.01)
xs3, fs3
```

```python


# %%
print(outputs3.table.to_latex(index=False))


# %%
xs4, fs4, outputs4 = steepestDescentMethod(f2, grad_f2, [-2,-3.5], 0.002)
xs4, fs4


# %%
outputs4.table

# %% [markdown]
# ## Newton's Method

# %%
def NewtonsMethod(x_0,epsilon,f,grad_f,Hessian_f):
    xk = np.array(x_0).reshape(2,1)
    k=0
    output = OutputTable()
    while(True):
        d_k=-np.linalg.inv(Hessian_f(xk))@np.transpose(grad_f(xk))
        alpha_k=ExactLineSearch(f,xk,d_k)
        xkp=xk+alpha_k*d_k
        if(np.linalg.norm(grad_f(xk)) < epsilon):
            break
        output.add_row(k, xk, f(xk), d_k, alpha_k, xkp)
        xk = xkp
        k += 1
    output.add_row(k,xk,f(xk),d_k,None,np.array([]))
    return xk, np.asscalar(f(xk)), output


# %%
x_f1_s1,f1_s1, outputf1_1 = NewtonsMethod([-5,1], 0.01,f1,grad_f1,hess_f1)
x_f1_s1,f1_s1
print( outputf1_1.table.to_latex(index=False))


# %%
x_f1_s2,f1_s2, outputf1_2 = NewtonsMethod([-25,75], 0.001,f1,grad_f1,hess_f1)
x_f1_s2,f1_s2
print( outputf1_2.table.to_latex(index=False))


# %%
x_f2_s1,f2_s1, outputf2_1 = NewtonsMethod([-2,4], 0.01,f1,grad_f1,hess_f1)
print( outputf2_1.table.to_latex(index=False))
# %%

x_f2_s2,f2_s2, outputf2_2 = NewtonsMethod([-10,1], 0.001,f1,grad_f1,hess_f1)
print( outputf2_2.table.to_latex(index=False))
```

```python
# %% [markdown]
# ## DFP

# %%
def DFP(f, grad_f, x_0, epsilon):
    xk = np.array(x_0).reshape(2,1)
    k = 0
    H = np.identity(len(x_0))
    stop = False
    output = OutputTable()
    while(stop == False):
        d = -H @ np.transpose(grad_f(xk))
        if(np.linalg.norm(d) < epsilon):
            stop = True
        else:
            a = ExactLineSearch(f,xk,d)
            xkp = xk + a*d
            p = xkp - xk
            q = np.transpose(grad_f(xkp)) - np.transpose(grad_f(xk))
            A = (p @ np.transpose(p)) / (p.transpose() @ q)
            B = - (H @ q @ np.transpose( H @ q)) / (q.transpose() @ H @ q)
            Hkp = H + A + B
            output.add_row(k, xk, f(xk), d, a, xkp)
            k += 1
            xk = xkp
            H = Hkp
    output.add_row(k,xk,f(xk),d,None,np.array([]))
    return xk, np.asscalar(f(xk)), output


# %%
xs1, fs1, output1 = DFP(f1, grad_f1, [0,0], 0.001)
xs1, fs1


# %%
output1.print_latex()


# %%
xs2, fs2, output2 = DFP(f1, grad_f1, [5,5], 0.0001)
xs2, fs2


# %%
output2.print_latex()


# %%
xs3, fs3, output3 = DFP(f2, grad_f2, [1.2,1.6], 1e-9)
xs3, fs3


# %%
```

```
output3.print_latex()


# %%
xs4, fs4, output4 = DFP(f2, grad_f2, [-2,-3], 1e-5)
xs4, fs4


# %%
output4.print_latex()

# %% [markdown]
# ## BFGS

# %%
def BFGS(f, grad_f, x_0, epsilon, line_search_tol = 0.0000001):
    xk = np.array(x_0).reshape(2,1)
    k = 0
    H = np.identity(len(x_0))
    stop = False
    output = OutputTable()
    while(stop == False):
        d = -H @ np.transpose(grad_f(xk))
        if(np.linalg.norm(d) < epsilon):
            stop = True
        if(k == -1):
            break
        else:
            a = ExactLineSearch(f,xk,d, line_search_tol)
            xkp = xk + a*d
            p = xkp - xk
            q = np.transpose(grad_f(xkp)) - np.transpose(grad_f(xk))
            A = ((1+ np.transpose(q) @ H @ q) / (np.transpose(q) @ p)) * (p @ np.
                transpose(p)) / (np.transpose(p) @ q)
            B = - (p @ np.transpose(q) @ H + H @ q @ np.transpose(p)) / (np.transpose(q)
                @ p)
            Hkp = H + A + B
            output.add_row(k, xk, f(xk), d, a, xkp)
            k += 1
            xk = xkp
            H = Hkp
    output.add_row(k,xk,f(xk),d,None,np.array([]))
    return xk, np.asscalar(f(xk)), output


# %%
xs1, fs1, output1 = BFGS(f1, grad_f1, [0,0], 0.01)
xs1, fs1


# %%
output1.print_latex()
```

```
# %%
xs2, fs2, output2 = BFGS(f1, grad_f1, [10,10], 0.01)
xs2, fs2


# %%
output2.print_latex()


# %%
xs3, fs3, output3 = BFGS(f2, grad_f2, [0,0], 0.01)
xs3, fs3


# %%
output3.print_latex()


# %%
xs4, fs4, output4 = BFGS(f2, grad_f2, [10,10], 0.001, line_search_tol=10**(-9))
xs4, fs4


# %%
output4.print_latex()
```

The complete output of the solution set 2 for the function 2 for the steepest descent method:

k	$x^{(k)}$	$f(x^{(k)})$	$d^{(k)}$	$\alpha^{(k)}$	$x^{(k+1)}$
0	[-2. ,-3.5]	5634.000000	[6006.,1500.]	0.000354036	[0.126,-2.969]
1	[0.126,-2.969]	891.730769	[-149.096, 596.982]	0.00591938	[-0.756, 0.565]
2	[-0.756, 0.565]	3.089270	[5.645,1.41]	0.311481	[1.002,1.004]
3	[1.002,1.004]	0.000004	[0.001,-0.002]	0.00235886	[1.002,1.004]
4	[1.002,1.004]	0.000004	[-0.003,-0.001]	0.00172039	[1.002,1.004]
5	[1.002,1.004]	0.000004	[0.001,-0.002]	0.00235916	[1.002,1.004]
6	[1.002,1.004]	0.000004	[-0.003,-0.001]	0.00171994	[1.002,1.004]
7	[1.002,1.004]	0.000004	[0.001,-0.002]	0.0023609	[1.002,1.004]
8	[1.002,1.004]	0.000004	[-0.003,-0.001]	0.00172183	[1.002,1.004]
9	[1.002,1.004]	0.000004	[0.001,-0.002]	0.00235275	[1.002,1.004]
10	[1.002,1.004]	0.000004	[-0.003,-0.001]	0.00172555	[1.002,1.004]
11	[1.002,1.004]	0.000004	[0.001,-0.002]	0.00234651	[1.002,1.004]
12	[1.002,1.004]	0.000004	[-0.003,-0.001]	0.0017266	[1.002,1.004]
13	[1.002,1.004]	0.000004	[0.001,-0.002]	0.00234954	[1.002,1.004]
14	[1.002,1.004]	0.000004	[-0.003,-0.001]	0.00172562	[1.002,1.004]
15	[1.002,1.004]	0.000004	[0.001,-0.002]	0.00234915	[1.002,1.004]

16	[1.002,1.004]	0.000004	[-0.003,-0.001]	0.00172853	[1.002,1.004]
17	[1.002,1.004]	0.000004	[0.001,-0.002]	0.00233937	[1.002,1.004]
18	[1.002,1.004]	0.000004	[-0.003,-0.001]	0.00173203	[1.002,1.004]
19	[1.002,1.004]	0.000004	[0.001,-0.002]	0.00233649	[1.002,1.004]
20	[1.002,1.004]	0.000004	[-0.003,-0.001]	0.00173288	[1.002,1.004]
21	[1.002,1.004]	0.000004	[0.001,-0.002]	0.00233603	[1.002,1.004]
22	[1.002,1.004]	0.000004	[-0.003,-0.001]	0.00173319	[1.002,1.004]
23	[1.002,1.004]	0.000004	[0.001,-0.002]	0.0023369	[1.002,1.004]
24	[1.002,1.004]	0.000004	[-0.003,-0.001]	0.00173163	[1.002,1.004]
25	[1.002,1.004]	0.000004	[0.001,-0.002]	0.00234023	[1.002,1.004]
26	[1.002,1.004]	0.000004	[-0.003,-0.001]	0.00172998	[1.002,1.004]
27	[1.002,1.004]	0.000004	[0.001,-0.002]	0.00234279	[1.002,1.004]
28	[1.002,1.004]	0.000004	[-0.003,-0.001]	0.00173003	[1.002,1.004]
29	[1.002,1.004]	0.000004	[0.001,-0.002]	0.00234016	[1.002,1.004]
30	[1.002,1.004]	0.000004	[-0.003,-0.001]	0.00173088	[1.002,1.004]
31	[1.002,1.004]	0.000004	[0.001,-0.002]	0.00234112	[1.002,1.004]
32	[1.002,1.004]	0.000004	[-0.003,-0.001]	0.00172987	[1.002,1.004]
33	[1.002,1.004]	0.000004	[0.001,-0.002]	0.00234263	[1.002,1.004]
34	[1.002,1.004]	0.000004	[-0.003,-0.001]	0.00172948	[1.002,1.004]
35	[1.002,1.004]	0.000004	[0.001,-0.002]	0.00234321	[1.002,1.004]
36	[1.002,1.004]	0.000004	[-0.003,-0.001]	0.00172941	[1.002,1.004]
37	[1.002,1.004]	0.000004	[0.001,-0.002]	0.00234229	[1.002,1.004]
38	[1.002,1.004]	0.000004	[-0.003,-0.001]	0.00173003	[1.002,1.004]
39	[1.002,1.004]	0.000004	[0.001,-0.002]	0.00234191	[1.002,1.004]
40	[1.002,1.004]	0.000004	[-0.003,-0.001]	0.00172756	[1.002,1.004]
41	[1.002,1.004]	0.000004	[0.001,-0.002]	0.00235094	[1.002,1.004]
42	[1.002,1.004]	0.000004	[-0.003,-0.001]	0.00172527	[1.002,1.004]
43	[1.002,1.004]	0.000004	[0.001,-0.002]	0.0023514	[1.002,1.004]
44	[1.002,1.004]	0.000004	[-0.003,-0.001]	0.00172555	[1.002,1.004]
45	[1.002,1.004]	0.000004	[0.001,-0.002]	0.00234838	[1.002,1.004]
46	[1.002,1.004]	0.000004	[-0.003,-0.001]	0.00172573	[1.002,1.004]
47	[1.002,1.004]	0.000004	[0.001,-0.002]	0.00234991	[1.002,1.004]
48	[1.002,1.004]	0.000004	[-0.003,-0.001]	0.00172593	[1.002,1.004]
49	[1.002,1.004]	0.000004	[0.001,-0.002]	0.00234994	[1.002,1.004]
50	[1.002,1.004]	0.000004	[-0.003,-0.001]	0.00172742	[1.002,1.004]
51	[1.002,1.004]	0.000004	[0.001,-0.002]	0.00234377	[1.002,1.004]

52	[1.002,1.004]	0.000004	[-0.002,-0.001]	0.00172928	[1.002,1.004]
53	[1.002,1.004]	0.000004	[0.001,-0.002]	0.00234282	[1.002,1.004]
54	[1.002,1.004]	0.000004	[-0.002,-0.001]	0.00173096	[1.002,1.004]
55	[1.002,1.004]	0.000004	[0.001,-0.002]	0.00233761	[1.002,1.004]
56	[1.002,1.004]	0.000004	[-0.002,-0.001]	0.00173217	[1.002,1.004]
57	[1.002,1.004]	0.000004	[0.001,-0.002]	0.00233961	[1.002,1.004]
58	[1.002,1.004]	0.000003	[-0.002,-0.001]	0.00173189	[1.002,1.004]
59	[1.002,1.004]	0.000003	[0.001,-0.002]	0.00233801	[1.002,1.004]
60	[1.002,1.004]	0.000003	[-0.002,-0.001]	0.00173148	[1.002,1.004]
61	[1.002,1.004]	0.000003	[0.001,-0.002]	0.00233948	[1.002,1.004]
62	[1.002,1.004]	0.000003	[-0.002,-0.001]	0.00173302	[1.002,1.004]
63	[1.002,1.004]	0.000003	[0.001,-0.002]	0.00233546	[1.002,1.004]
64	[1.002,1.004]	0.000003	[-0.002,-0.001]	0.00173319	[1.002,1.004]
65	[1.002,1.004]	0.000003	[0.001,-0.002]	0.00233667	[1.002,1.004]
66	[1.002,1.004]	0.000003	[-0.002,-0.001]	0.00173219	[1.002,1.004]
67	[1.002,1.004]	0.000003	[0.001,-0.002]	0.00234038	[1.002,1.004]
68	[1.002,1.004]	0.000003	[-0.002,-0.001]	0.0017311	[1.002,1.004]
69	[1.002,1.004]	0.000003	[0.001,-0.002]	0.00234117	[1.002,1.004]
70	[1.002,1.004]	0.000003	[-0.002,-0.001]	0.00172778	[1.002,1.004]
71	[1.002,1.004]	0.000003	[0.001,-0.002]	0.00234954	[1.002,1.004]
72	[1.002,1.004]	0.000003	[-0.002,-0.001]	0.00172662	[1.002,1.004]
73	[1.002,1.004]	0.000003	[0.001,-0.002]	0.00234881	[1.002,1.004]
74	[1.002,1.004]	0.000003	[-0.002,-0.001]	0.00172667	[1.002,1.004]
75	[1.002,1.004]	0.000003	[0.001,-0.002]	0.00234817	[1.002,1.004]
76	[1.002,1.004]	0.000003	[-0.002,-0.001]	0.00172702	[1.002,1.004]
77	[1.002,1.004]	0.000003	[0.001,-0.002]	0.0023471	[1.002,1.004]
78	[1.002,1.004]	0.000003	[-0.002,-0.001]	0.00172857	[1.002,1.004]
79	[1.002,1.004]	0.000003	[0.001,-0.002]	0.00234369	[1.002,1.004]
80	[1.002,1.004]	0.000003	[-0.002,-0.001]	0.00172825	[1.002,1.004]
81	[1.002,1.004]	0.000003	[0.001,-0.002]	0.00234634	[1.002,1.004]
82	[1.002,1.004]	0.000003	[-0.002,-0.001]	0.00172905	[1.002,1.004]
83	[1.002,1.004]	0.000003	[0.001,-0.002]	0.00234283	[1.002,1.004]
84	[1.002,1.004]	0.000003	[-0.002,-0.001]	0.00173012	[1.002,1.004]
85	[1.002,1.004]	0.000003	[0.001,-0.002]	0.00234226	[1.002,1.004]
86	[1.002,1.004]	0.000003	[-0.002,-0.001]	0.00173004	[1.002,1.004]
87	[1.002,1.004]	0.000003	[0.001,-0.002]	0.00234223	[1.002,1.004]

88	[1.002,1.004]	0.000003	[-0.002,-0.001]	0.00173002	[1.002,1.004]
89	[1.002,1.004]	0.000003	[0.001,-0.002]	0.00234246	[1.002,1.004]
90	[1.002,1.004]	0.000003	[-0.002,-0.001]	0.0017311	[1.002,1.004]
91	[1.002,1.004]	0.000003	[0.001,-0.002]	0.00233948	[1.002,1.004]
92	[1.002,1.004]	0.000003	[-0.002,-0.001]	0.00173141	[1.002,1.004]
93	[1.002,1.004]	0.000003	[0.001,-0.002]	0.0023408	[1.002,1.004]
94	[1.002,1.004]	0.000003	[-0.002,-0.001]	0.00172954	[1.002,1.004]
95	[1.002,1.004]	0.000003	[0.001,-0.002]	0.00234544	[1.002,1.004]
96	[1.002,1.004]	0.000003	[-0.002,-0.001]	0.00172814	[1.002,1.004]
97	[1.002,1.004]	0.000003	[0.001,-0.002]	0.00234614	[1.002,1.004]
98	[1.002,1.004]	0.000003	[-0.002,-0.001]	0.00172821	[1.002,1.004]
99	[1.002,1.004]	0.000003	[0.001,-0.002]	0.00234651	[1.002,1.004]
100	[1.002,1.004]	0.000003	[-0.002,-0.001]	0.00172872	[1.002,1.004]
101	[1.002,1.004]	0.000003	[0.001,-0.002]	0.00234309	[1.002,1.004]
102	[1.002,1.004]	0.000003	[-0.002,-0.001]	0.0017294	[1.002,1.004]
103	[1.002,1.004]	0.000003	[0.001,-0.002]	0.00234537	[1.002,1.004]
104	[1.002,1.004]	0.000003	[-0.002,-0.001]	0.00172785	[1.002,1.004]
105	[1.002,1.004]	0.000003	[0.001,-0.002]	0.00234852	[1.002,1.004]
106	[1.002,1.004]	0.000003	[-0.002,-0.001]	0.0017263	[1.002,1.004]
107	[1.002,1.004]	0.000003	[0.001,-0.002]	0.00234973	[1.002,1.004]
108	[1.002,1.004]	0.000003	[-0.002,-0.001]	0.00172555	[1.002,1.004]
109	[1.002,1.004]	0.000003	[0.001,-0.002]	0.00235296	[1.002,1.004]
110	[1.002,1.004]	0.000003	[-0.002,-0.001]	0.00172369	[1.002,1.004]
111	[1.002,1.004]	0.000003	[0.001,-0.002]	0.0023555	[1.002,1.004]
112	[1.002,1.004]	0.000003	[-0.002,-0.001]	0.00172215	[1.002,1.004]
113	[1.002,1.004]	0.000003	[0.001,-0.002]	0.00235994	[1.002,1.004]
114	[1.002,1.004]	0.000003	[-0.002,-0.001]	0.00172111	[1.002,1.004]
115	[1.002,1.004]	0.000003	[0.001,-0.002]	0.00235898	[1.002,1.004]
116	[1.002,1.004]	0.000003	[-0.002,-0.001]	0.00172258	[1.002,1.004]
117	[1.002,1.004]	0.000003	[0.001,-0.002]	0.00235344	[1.002,1.004]
118	[1.002,1.004]	0.000003	[-0.002,-0.001]	0.00172411	[1.002,1.004]
119	[1.002,1.004]	0.000003	[0.001,-0.002]	0.00235522	[1.002,1.004]
120	[1.002,1.004]	0.000003	[-0.002,-0.001]	0.00172224	[1.002,1.004]
121	[1.002,1.004]	0.000003	[0.001,-0.002]	0.00235918	[1.002,1.004]
122	[1.002,1.004]	0.000003	[-0.002,-0.001]	0.00172313	[1.002,1.004]
123	[1.002,1.004]	0.000003	[0.001,-0.002]	0.00235213	[1.002,1.004]

124	[1.002,1.004]	0.000003	[-0.002,-0.001]	0.00172572	[1.002,1.004]
125	[1.002,1.004]	0.000003	[0.001,-0.002]	0.00234914	[1.002,1.004]
126	[1.002,1.004]	0.000003	[-0.002,-0.001]	0.001727	[1.002,1.004]
127	[1.002,1.004]	0.000003	[0.001,-0.002]	0.0023486	[1.002,1.004]
128	[1.002,1.004]	0.000003	[-0.002,-0.001]	0.00172705	[1.002,1.004]
129	[1.002,1.004]	0.000003	[0.001,-0.002]	0.00234858	[1.002,1.004]
130	[1.002,1.004]	0.000003	[-0.002,-0.001]	0.00172709	[1.002,1.004]
131	[1.002,1.004]	0.000003	[0.001,-0.002]	0.00234863	[1.002,1.004]
132	[1.002,1.004]	0.000003	[-0.002,-0.001]	0.00172677	[1.002,1.004]
133	[1.002,1.004]	0.000003	[0.001,-0.002]	0.00234992	[1.002,1.004]
134	[1.002,1.004]	0.000003	[-0.002,-0.001]	0.00172564	[1.002,1.004]
135	[1.002,1.004]	0.000003	[0.001,-0.002]	0.00235196	[1.002,1.004]
136	[1.002,1.004]	0.000003	[-0.002,-0.001]	0.0017256	[1.002,1.004]
137	[1.002,1.004]	0.000003	[0.001,-0.002]	0.00235023	[1.002,1.004]
138	[1.002,1.004]	0.000003	[-0.002,-0.001]	0.00172594	[1.002,1.004]
139	[1.002,1.004]	0.000003	[0.001,-0.002]	0.00235187	[1.002,1.004]
140	[1.002,1.004]	0.000003	[-0.002,-0.001]	0.0017244	[1.002,1.004]
141	[1.002,1.004]	0.000003	[0. ,-0.002]	0.00235506	[1.002,1.003]
142	[1.002,1.003]	0.000003	[-0.002,-0.001]	0.00172168	[1.002,1.003]
143	[1.002,1.003]	0.000003	[0. ,-0.002]	0.0023624	[1.002,1.003]
144	[1.002,1.003]	0.000003	[-0.002,-0.001]	0.00171997	[1.002,1.003]
145	[1.002,1.003]	0.000003	[0. ,-0.002]	0.00236128	[1.002,1.003]
146	[1.002,1.003]	0.000003	[-0.002,-0.001]	0.00171959	[1.002,1.003]
147	[1.002,1.003]	0.000003	[0. ,-0.002]	0.00236382	[1.002,1.003]
148	[1.002,1.003]	0.000003	[-0.002,-0.001]	0.00171853	[1.002,1.003]
149	[1.002,1.003]	0.000003	[0. ,-0.002]	0.00236641	[1.002,1.003]
150	[1.002,1.003]	0.000003	[-0.002,-0.001]	0.00171714	[1.002,1.003]
151	[1.002,1.003]	0.000003	[0. ,-0.002]	0.00236808	[1.002,1.003]
152	[1.002,1.003]	0.000003	[-0.002,-0.001]	0.00171597	[1.002,1.003]
153	[1.002,1.003]	0.000003	[0. ,-0.002]	0.00237088	[1.002,1.003]
154	[1.002,1.003]	0.000003	[-0.002,-0.001]	0.00171484	[1.002,1.003]
155	[1.002,1.003]	0.000003	[0. ,-0.002]	0.00237233	[1.002,1.003]
156	[1.002,1.003]	0.000003	[-0.002,-0.001]	0.00171343	[1.002,1.003]
157	[1.002,1.003]	0.000003	[0. ,-0.002]	0.00237753	[1.002,1.003]
158	[1.002,1.003]	0.000003	[-0.002,-0.001]	0.00171233	[1.002,1.003]
159	[1.002,1.003]	0.000003	[0. ,-0.002]	0.002376	[1.002,1.003]

160	[1.002,1.003]	0.000003	[-0.002,-0.001]	0.00171363	[1.002,1.003]
161	[1.002,1.003]	0.000003	[0. ,-0.002]	0.00237077	[1.002,1.003]
162	[1.002,1.003]	0.000003	[-0.002,-0.001]	0.00171703	[1.002,1.003]
163	[1.002,1.003]	0.000003	[0. ,-0.002]	0.00236602	[1.002,1.003]
164	[1.002,1.003]	0.000003	[-0.002,-0.001]	0.001718	[1.002,1.003]
165	[1.002,1.003]	0.000003	[0. ,-0.002]	0.0023663	[1.002,1.003]
166	[1.002,1.003]	0.000003	[-0.002,-0.001]	0.00171834	[1.002,1.003]
167	[1.002,1.003]	0.000003	[0. ,-0.002]	0.00236276	[1.002,1.003]
168	[1.002,1.003]	0.000003	[-0.002,-0.001]	0.00171951	[1.002,1.003]
169	[1.002,1.003]	0.000003	[0. ,-0.002]	0.00236421	[1.002,1.003]
170	[1.002,1.003]	0.000003	[-0.002,-0.001]	0.00171868	[1.002,1.003]
171	[1.002,1.003]	0.000003	[0. ,-0.002]	0.00236575	[1.002,1.003]
172	[1.002,1.003]	0.000003	[-0.002,-0.001]	0.00171961	[1.002,1.003]
173	[1.002,1.003]	0.000003	[0. ,-0.002]	None	[]

The complete output of the solution set 2 for the function 2 for the Newton's method:

k	$x^{(k)}$	$f(x^{(k)})$	$d^{(k)}$	$\alpha^{(k)}$	$x^{(k+1)}$
0	[-10, 1]	6765345.0000	[16.5,65.5]	2.853838	[37.088,187.927]
1	[37.088,187.927]	942.5554	[-30.588,-153.743]	1.007234	[6.279,33.071]
2	[6.279,33.071]	-21.6320	[0.221,0.606]	1.731007	[6.662,34.121]
3	[6.662,34.121]	-27.4131	[-0.162,-0.827]	1.000977	[6.5 ,33.293]
4	[6.5 ,33.293]	-27.4406	[0.,0.]	-3.326661	[6.499,33.292]
5	[6.499,33.292]	-27.4405	[0.001,0.002]	0.976539	[6.5 ,33.294]
6	[6.5 ,33.294]	-27.4406	[1.604e-05,3.586e-05]	71.725821	[6.501,33.296]
7	[6.501,33.296]	-27.4405	[-0.001,-0.003]	1.049374	[6.5 ,33.294]
8	[6.5 ,33.294]	-27.4406	[5.600e-05,1.122e-04]	43.741345	[6.502,33.298]
9	[6.502,33.298]	-27.4404	[-0.002,-0.005]	1.173416	[6.5 ,33.293]
10	[6.5 ,33.293]	-27.4405	[0. ,0.001]	2.448764	[6.501,33.295]
11	[6.501,33.295]	-27.4405	[-0.001,-0.001]	1.981931	[6.499,33.293]
12	[6.499,33.293]	-27.4405	[0.001,0.001]	2.288026	[6.501,33.295]
13	[6.501,33.295]	-27.4405	[-0.001,-0.001]	1.686125	[6.499,33.293]
14	[6.499,33.293]	-27.4405	[0.001,0.001]	-0.390744	[6.499,33.292]
15	[6.499,33.292]	-27.4405	[0.001,0.001]	2.375133	[6.501,33.296]
16	[6.501,33.296]	-27.4405	[-0.001,-0.002]	1.027546	[6.5 ,33.294]
17	[6.5 ,33.294]	-27.4406	[2.749e-05,3.692e-05]	-56.445361	[6.498,33.292]
18	[6.498,33.292]	-27.4404	[0.002,0.002]	1.196387	[6.5 ,33.294]

19	[6.5 ,33.294]	-27.4405	[-0.,-0.]	2.629081	[6.499,33.293]
20	[6.499,33.293]	-27.4405	[0.001,0.001]	0.942940	[6.5 ,33.294]
21	[6.5 ,33.294]	-27.4406	[2.883e-05,3.943e-05]	89.837504	[6.503,33.297]
22	[6.503,33.297]	-27.4402	[-0.003,-0.003]	0.967395	[6.5 ,33.294]
23	[6.5 ,33.294]	-27.4406	[-8.350e-05,-2.213e-04]	-44.517045	[6.504,33.304]
24	[6.504,33.304]	-27.4402	[-0.004,-0.01]	1.050586	[6.5 ,33.293]
25	[6.5 ,33.293]	-27.4405	[0.,0.]	-2.197862	[6.499,33.292]
26	[6.499,33.292]	-27.4405	[0.001,0.001]	1.562500	[6.5 ,33.294]
27	[6.5 ,33.294]	-27.4405	[-0. ,-0.001]	4.442324	[6.499,33.291]
28	[6.499,33.291]	-27.4405	[0.001,0.003]	1.122629	[6.5 ,33.294]
29	[6.5 ,33.294]	-27.4406	[-0.,-0.]	-12.556672	[6.502,33.298]
30	[6.502,33.298]	-27.4404	[-0.002,-0.004]	1.110147	[6.5 ,33.293]
31	[6.5 ,33.293]	-27.4405	[0.,0.]	-0.819779	[6.5 ,33.293]
32	[6.5 ,33.293]	-27.4405	[0. ,0.001]	0.323084	[6.5 ,33.293]
33	[6.5 ,33.293]	-27.4405	[0. ,0.001]	8.137244	[6.502,33.298]
34	[6.502,33.298]	-27.4404	[-0.002,-0.004]	1.196291	[6.5 ,33.293]
35	[6.5 ,33.293]	-27.4405	[0. ,0.001]	7.129866	[6.502,33.298]
36	[6.502,33.298]	-27.4404	[-0.002,-0.004]	0.927716	[6.5 ,33.294]
37	[6.5 ,33.294]	-27.4405	[-0.,-0.]	20.915652	[6.497,33.286]
38	[6.497,33.286]	-27.4402	[0.003,0.008]	1.074416	[6.5 ,33.294]
39	[6.5 ,33.294]	-27.4405	[-0. ,-0.001]	5.074978	[6.499,33.291]
40	[6.499,33.291]	-27.4405	[0.001,0.003]	1.402740	[6.5 ,33.295]
41	[6.5 ,33.295]	-27.4405	[-0. ,-0.001]	3.074394	[6.499,33.291]
42	[6.499,33.291]	-27.4405	[0.001,0.002]	0.573618	[6.5 ,33.293]
43	[6.5 ,33.293]	-27.4405	[0. ,0.001]	2.826817	[6.501,33.295]
44	[6.501,33.295]	-27.4405	[-0.001,-0.002]	2.517589	[6.499,33.291]
45	[6.499,33.291]	-27.4405	[0.001,0.003]	1.169698	[6.5 ,33.294]
46	[6.5 ,33.294]	-27.4406	[-0.,-0.]	13.850820	[6.498,33.288]
47	[6.498,33.288]	-27.4405	[0.002,0.006]	0.756836	[6.499,33.292]
48	[6.499,33.292]	-27.4405	[0.001,0.001]	3.887978	[6.502,33.298]
49	[6.502,33.298]	-27.4405	[-0.002,-0.004]	1.590443	[6.499,33.291]
50	[6.499,33.291]	-27.4405	[0.001,0.002]	3.126624	[6.502,33.299]
51	[6.502,33.299]	-27.4405	[-0.002,-0.005]	1.069370	[6.5 ,33.293]
52	[6.5 ,33.293]	-27.4406	[0.,0.]	53.375246	[6.507,33.311]
53	[6.507,33.311]	-27.4394	[-0.007,-0.017]	0.980864	[6.5 ,33.294]
54	[6.5 ,33.294]	-27.4406	[-0. ,-0.001]	99.554060	[6.487,33.223]

55	[6.487,33.223]	-27.4403	[0.013,0.071]	1.045254	[6.501,33.297]
56	[6.501,33.297]	-27.4406	[-0.001,-0.003]	-3.400046	[6.503,33.308]
57	[6.503,33.308]	-27.4405	[-0.003,-0.014]	3.161932	[6.494,33.263]
58	[6.494,33.263]	-27.4405	[0.006,0.031]	1.080835	[6.5 ,33.296]
59	[6.5 ,33.296]	-27.4406	[-0. ,-0.002]	38.419914	[6.483,33.201]
60	[6.483,33.201]	-27.4400	[0.017,0.093]	1.033218	[6.501,33.297]
61	[6.501,33.297]	-27.4406	[-0.001,-0.003]	31.263841	[6.483,33.198]
62	[6.483,33.198]	-27.4399	[0.017,0.096]	1.008612	[6.5 ,33.295]
63	[6.5 ,33.295]	-27.4406	[-0. ,-0.001]	95.675001	[6.486,33.203]
64	[6.486,33.203]	-27.4387	[0.014,0.091]	0.982727	[6.5 ,33.293]
65	[6.5 ,33.293]	-27.4406	[0. ,0.001]	46.264118	[6.511,33.339]
66	[6.511,33.339]	-27.4401	[-0.011,-0.046]	1.074266	[6.499,33.29]
67	[6.499,33.29]	-27.4405	[0.001,0.003]	6.904407	[6.505,33.313]
68	[6.505,33.313]	-27.4404	[-0.005,-0.019]	1.031443	[6.5 ,33.293]
69	[6.5 ,33.293]	-27.4406	[0. ,0.001]	-0.000771	[6.5 ,33.293]
70	[6.5 ,33.293]	-27.4406	[0. ,0.001]	75.985428	[6.511,33.337]
71	[6.511,33.337]	-27.4398	[-0.011,-0.044]	1.002556	[6.5 ,33.294]
72	[6.5 ,33.294]	-27.4406	[2.891e-05,-9.908e-05]	91.980602	[6.503,33.285]
73	[6.503,33.285]	-27.4387	[-0.003, 0.01]	0.970071	[6.5 ,33.294]
74	[6.5 ,33.294]	-27.4406	[-7.873e-05,-3.539e-04]	99.191856	[6.492,33.259]
75	[6.492,33.259]	-27.4404	[0.008,0.035]	1.096487	[6.501,33.297]
76	[6.501,33.297]	-27.4405	[-0.001,-0.003]	18.002322	[6.487,33.236]
77	[6.487,33.236]	-27.4403	[0.013,0.057]	1.032321	[6.5 ,33.296]
78	[6.5 ,33.296]	-27.4406	[-0. ,-0.002]	11.645217	[6.496,33.273]
79	[6.496,33.273]	-27.4405	[0.004,0.02]	1.462975	[6.502,33.303]
80	[6.502,33.303]	-27.4405	[-0.002,-0.009]	-0.119033	[6.502,33.304]
81	[6.502,33.304]	-27.4405	[-0.002,-0.01]	6.250192	[6.488,33.239]
82	[6.488,33.239]	-27.4403	[0.012,0.055]	0.974929	[6.5 ,33.292]
83	[6.5 ,33.292]	-27.4406	[0. ,0.001]	-14.219488	[6.495,33.273]
84	[6.495,33.273]	-27.4405	[0.005,0.021]	1.175888	[6.501,33.297]
85	[6.501,33.297]	-27.4405	[-0.001,-0.004]	3.046671	[6.498,33.286]
86	[6.498,33.286]	-27.4405	[0.002,0.007]	-0.449587	[6.498,33.283]
87	[6.498,33.283]	-27.4405	[0.002,0.011]	3.040034	[6.505,33.316]
88	[6.505,33.316]	-27.4405	[-0.005,-0.022]	1.655030	[6.497,33.279]
89	[6.497,33.279]	-27.4405	[0.003,0.014]	1.989842	[6.503,33.308]
90	[6.503,33.308]	-27.4405	[-0.003,-0.014]	1.893259	[6.497,33.281]

91	[6.497,33.281]	-27.4405	[0.003,0.013]	1.460227	[6.501,33.3]
92	[6.501,33.3]	-27.4405	[-0.001,-0.006]	5.397794	[6.494,33.268]
93	[6.494,33.268]	-27.4405	[0.006,0.026]	0.780743	[6.499,33.288]
94	[6.499,33.288]	-27.4405	[0.001,0.006]	12.707758	[6.515,33.36]
95	[6.515,33.36]	-27.4402	[-0.015,-0.066]	1.050997	[6.499,33.29]
96	[6.499,33.29]	-27.4405	[0.001,0.003]	21.126466	[6.515,33.361]
97	[6.515,33.361]	-27.4401	[-0.015,-0.067]	0.975029	[6.5 ,33.295]
98	[6.5 ,33.295]	-27.4406	[-0. ,-0.002]	47.838250	[6.483,33.212]
99	[6.483,33.212]	-27.4401	[0.017,0.082]	1.000046	[6.5 ,33.294]
100	[6.5 ,33.294]	-27.4406	[-8.024e-07,-4.145e-05]	49.969385	[6.5 ,33.292]
101	[6.5 ,33.292]	-27.4405	[3.930e-05,2.034e-03]	3.161621	[6.5 ,33.298]
102	[6.5 ,33.298]	-27.4405	[-8.494e-05,-4.381e-03]	1.574708	[6.5 ,33.291]
103	[6.5 ,33.291]	-27.4405	[4.882e-05,2.504e-03]	3.125000	[6.5 ,33.299]
104	[6.5 ,33.299]	-27.4405	[-0. ,-0.005]	1.265893	[6.5 ,33.292]
105	[6.5 ,33.292]	-27.4405	[2.758e-05,1.382e-03]	2.047395	[6.5 ,33.295]
106	[6.5 ,33.295]	-27.4405	[-2.889e-05,-1.447e-03]	6.250061	[6.5 ,33.286]
107	[6.5 ,33.286]	-27.4404	[0. ,0.008]	1.123145	[6.5 ,33.295]
108	[6.5 ,33.295]	-27.4405	[-1.868e-05,-1.001e-03]	-0.391017	[6.5 ,33.295]
109	[6.5 ,33.295]	-27.4405	[-2.598e-05,-1.392e-03]	3.540847	[6.5 ,33.29]
110	[6.5 ,33.29]	-27.4405	[6.601e-05,3.548e-03]	1.255649	[6.5 ,33.295]
111	[6.5 ,33.295]	-27.4405	[-1.688e-05,-9.191e-04]	7.815591	[6.5 ,33.287]
112	[6.5 ,33.287]	-27.4404	[0. ,0.006]	1.000212	[6.5 ,33.294]
113	[6.5 ,33.294]	-27.4406	[-2.44e-08,-4.25e-05]	87.490075	[6.5 ,33.29]
114	[6.5 ,33.29]	-27.4405	[2.110e-06,3.693e-03]	1.256149	[6.5 ,33.295]
115	[6.5 ,33.295]	-27.4405	[-5.405e-07,-9.618e-04]	1.561725	[6.5 ,33.293]
116	[6.5 ,33.293]	-27.4405	[3.036e-07,5.394e-04]	12.507725	[6.5,33.3]
117	[6.5,33.3]	-27.4404	[-3.494e-06,-6.160e-03]	0.999310	[6.5 ,33.294]
118	[6.5 ,33.294]	-27.4406	[-2.411e-09,-5.204e-05]	99.999571	[6.5 ,33.289]
119	[6.5 ,33.289]	-27.4405	[2.387e-07,5.185e-03]	1.025295	[6.5 ,33.294]
120	[6.5 ,33.294]	-27.4406	[-6.037e-09,-1.648e-04]	84.316189	[6.5 ,33.28]
121	[6.5 ,33.28]	-27.4398	[5.030e-07,1.398e-02]	0.989720	[6.5 ,33.294]
122	[6.5 ,33.294]	-27.4406	[5.170e-09,-9.946e-05]	49.594565	[6.5 ,33.289]
123	[6.5 ,33.289]	-27.4405	[-2.513e-07, 4.863e-03]	1.172119	[6.5 ,33.295]
124	[6.5 ,33.295]	-27.4405	[4.325e-08,-8.657e-04]	6.689878	[6.5 ,33.289]
125	[6.5 ,33.289]	-27.4405	[-2.461e-07, 4.956e-03]	1.124382	[6.5 ,33.294]
126	[6.5 ,33.294]	-27.4405	[3.061e-08,-6.467e-04]	7.582814	[6.5 ,33.289]

127	[6.5 ,33.289]	-27.4405	[-2.015e-07, 4.280e-03]	1.178009	[6.5 ,33.294]
128	[6.5 ,33.294]	-27.4405	[3.586e-08,-7.841e-04]	6.861116	[6.5 ,33.289]
129	[6.5 ,33.289]	-27.4405	[-2.102e-07, 4.622e-03]	1.052254	[6.5 ,33.294]
130	[6.5 ,33.294]	-27.4406	[1.098e-08,-2.682e-04]	12.499905	[6.5 ,33.291]
131	[6.5 ,33.291]	-27.4405	[-1.263e-07, 3.096e-03]	1.152752	[6.5 ,33.294]
132	[6.5 ,33.294]	-27.4405	[1.929e-08,-4.847e-04]	10.434911	[6.5 ,33.289]
133	[6.5 ,33.289]	-27.4405	[-1.82e-07, 4.60e-03]	1.105105	[6.5 ,33.294]
134	[6.5 ,33.294]	-27.4405	[1.913e-08,-5.097e-04]	8.036728	[6.5 ,33.29]
135	[6.5 ,33.29]	-27.4405	[-1.346e-07, 3.603e-03]	1.185384	[6.5 ,33.294]
136	[6.5 ,33.294]	-27.4405	[2.496e-08,-6.836e-04]	2.883359	[6.5 ,33.292]
137	[6.5 ,33.292]	-27.4405	[-4.701e-08, 1.289e-03]	3.333771	[6.5 ,33.297]
138	[6.5 ,33.297]	-27.4405	[1.097e-07,-2.999e-03]	0.952148	[6.5 ,33.294]
139	[6.5 ,33.294]	-27.4406	[5.250e-09,-1.548e-04]	-0.439596	[6.5 ,33.294]
140	[6.5 ,33.294]	-27.4406	[7.557e-09,-2.229e-04]	46.387498	[6.5 ,33.284]
141	[6.5 ,33.284]	-27.4402	[-3.430e-07, 1.025e-02]	1.008021	[6.5 ,33.294]
142	[6.5 ,33.294]	-27.4406	[2.751e-09,-2.133e-04]	33.175718	[6.5 ,33.287]
143	[6.5 ,33.287]	-27.4404	[-8.853e-08, 6.923e-03]	1.038741	[6.5 ,33.294]
144	[6.5 ,33.294]	-27.4406	[3.430e-09,-3.281e-04]	38.820103	[6.5 ,33.281]
145	[6.5 ,33.281]	-27.4400	[-1.297e-07, 1.261e-02]	1.013184	[6.5 ,33.294]
146	[6.5 ,33.294]	-27.4406	[1.710e-09,-3.642e-04]	-0.457860	[6.5 ,33.294]
147	[6.5 ,33.294]	-27.4405	[2.493e-09,-5.308e-04]	6.322405	[6.5 ,33.291]
148	[6.5 ,33.291]	-27.4405	[-1.327e-08, 2.835e-03]	1.165390	[6.5 ,33.294]
149	[6.5 ,33.294]	-27.4405	[2.195e-09,-4.787e-04]	9.538821	[6.5 ,33.29]
150	[6.5 ,33.29]	-27.4405	[-1.874e-08, 4.108e-03]	1.125201	[6.5 ,33.294]
151	[6.5 ,33.294]	-27.4405	[2.346e-09,-5.352e-04]	12.758446	[6.5 ,33.287]
152	[6.5 ,33.287]	-27.4404	[-2.759e-08, 6.343e-03]	1.078069	[6.5 ,33.294]
153	[6.5 ,33.294]	-27.4405	[2.154e-09,-5.453e-04]	-0.781254	[6.5 ,33.295]
154	[6.5 ,33.295]	-27.4405	[3.836e-09,-9.705e-04]	2.410503	[6.5 ,33.292]
155	[6.5 ,33.292]	-27.4405	[-5.411e-09, 1.370e-03]	1.537909	[6.5 ,33.294]
156	[6.5 ,33.294]	-27.4405	[2.911e-09,-7.386e-04]	3.024339	[6.5 ,33.292]
157	[6.5 ,33.292]	-27.4405	[-5.892e-09, 1.497e-03]	1.543873	[6.5 ,33.295]
158	[6.5 ,33.295]	-27.4405	[3.205e-09,-8.163e-04]	3.511486	[6.5 ,33.292]
159	[6.5 ,33.292]	-27.4405	[-8.048e-09, 2.055e-03]	1.512402	[6.5 ,33.295]
160	[6.5 ,33.295]	-27.4405	[4.124e-09,-1.057e-03]	2.464667	[6.5 ,33.292]
161	[6.5 ,33.292]	-27.4405	[-6.040e-09, 1.549e-03]	2.393457	[6.5 ,33.296]
162	[6.5 ,33.296]	-27.4405	[8.417e-09,-2.156e-03]	0.780865	[6.5 ,33.294]

163	[6.5 ,33.294]	-27.4405	[1.844e-09,-4.781e-04]	25.000000	[6.5 ,33.282]
164	[6.5 ,33.282]	-27.4401	[-4.427e-08, 1.164e-02]	0.998201	[6.5 ,33.294]
165	[6.5 ,33.294]	-27.4406	[-7.964e-11,-1.481e-04]	85.148233	[6.5 ,33.281]
166	[6.5 ,33.281]	-27.4400	[6.702e-09,1.267e-02]	0.997863	[6.5 ,33.294]
167	[6.5 ,33.294]	-27.4406	[1.432e-11,-1.729e-04]	76.766391	[6.5 ,33.281]
168	[6.5 ,33.281]	-27.4399	[-1.085e-09, 1.332e-02]	1.003763	[6.5 ,33.294]
169	[6.5 ,33.294]	-27.4406	[4.084e-12,-2.710e-04]	24.120638	[6.5 ,33.287]
170	[6.5 ,33.287]	-27.4404	[-9.442e-11, 6.315e-03]	1.078808	[6.5 ,33.294]
171	[6.5 ,33.294]	-27.4405	[7.441e-12,-5.473e-04]	-0.198555	[6.5 ,33.294]
172	[6.5 ,33.294]	-27.4405	[8.919e-12,-6.558e-04]	6.647228	[6.5 ,33.29]
173	[6.5 ,33.29]	-27.4405	[-5.037e-11, 3.720e-03]	1.025390	[6.5 ,33.294]
174	[6.5 ,33.294]	-27.4406	[1.279e-12,-1.118e-04]	99.975393	[6.5 ,33.283]
175	[6.5 ,33.283]	-27.4401	[-1.266e-10, 1.122e-02]	0.994632	[6.5 ,33.294]
176	[6.5 ,33.294]	-27.4406	[-6.792e-13,-9.698e-05]	63.414662	[6.5 ,33.288]
177	[6.5 ,33.288]	-27.4404	[4.239e-11,6.100e-03]	1.072458	[6.5 ,33.294]
178	[6.5 ,33.294]	-27.4405	[-3.072e-12,-4.883e-04]	14.881905	[6.5 ,33.287]
179	[6.5 ,33.287]	-27.4404	[4.264e-11,6.837e-03]	1.049354	[6.5 ,33.294]
180	[6.5 ,33.294]	-27.4406	[-2.104e-12,-3.958e-04]	18.141418	[6.5 ,33.287]
181	[6.5 ,33.287]	-27.4404	[3.607e-11,6.843e-03]	1.024884	[6.5 ,33.294]
182	[6.5 ,33.294]	-27.4406	[-8.977e-13,-2.289e-04]	12.394494	[6.5 ,33.291]
183	[6.5 ,33.291]	-27.4405	[1.023e-11,2.616e-03]	1.374407	[6.5 ,33.295]
184	[6.5 ,33.295]	-27.4405	[-3.83e-12,-9.87e-04]	2.477628	[6.5 ,33.292]
185	[6.5 ,33.292]	-27.4405	[5.66e-12,1.46e-03]	2.735925	[6.5 ,33.296]
186	[6.5 ,33.296]	-27.4405	[-9.826e-12,-2.529e-03]	1.624704	[6.5 ,33.292]
187	[6.5 ,33.292]	-27.4405	[6.138e-12,1.575e-03]	1.533889	[6.5 ,33.295]
188	[6.5 ,33.295]	-27.4405	[-3.277e-12,-8.430e-04]	1.440429	[6.5 ,33.293]
189	[6.5 ,33.293]	-27.4406	[1.443e-12,3.705e-04]	27.353241	[6.5 ,33.303]
190	[6.5 ,33.303]	-27.4402	[-3.803e-11,-9.647e-03]	1.048328	[6.5 ,33.293]
191	[6.5 ,33.293]	-27.4406	[1.839e-12,3.482e-04]	12.059593	[6.5 ,33.298]
192	[6.5 ,33.298]	-27.4405	[-2.034e-11,-3.832e-03]	1.169468	[6.5 ,33.293]
193	[6.5 ,33.293]	-27.4405	[3.446e-12,6.314e-04]	5.677640	[6.5 ,33.297]
194	[6.5 ,33.297]	-27.4405	[-1.612e-11,-2.943e-03]	1.148005	[6.5 ,33.293]
195	[6.5 ,33.293]	-27.4406	[2.385e-12,4.249e-04]	13.903552	[6.5 ,33.299]
196	[6.5 ,33.299]	-27.4404	[-3.078e-11,-5.445e-03]	1.125146	[6.5 ,33.293]
197	[6.5 ,33.293]	-27.4405	[3.851e-12,6.444e-04]	6.983636	[6.5 ,33.298]
198	[6.5 ,33.298]	-27.4405	[-2.304e-11,-3.838e-03]	1.246644	[6.5 ,33.293]

199	[6.5 ,33.293]	-27.4405	[5.683e-12,9.290e-04]	4.516799	[6.5 ,33.297]
200	[6.5 ,33.297]	-27.4405	[-1.999e-11,-3.255e-03]	0.779724	[6.5 ,33.294]
201	[6.5 ,33.294]	-27.4405	[-4.402e-12,-7.297e-04]	7.868494	[6.5 ,33.289]
202	[6.5 ,33.289]	-27.4405	[3.023e-11,5.043e-03]	1.039112	[6.5 ,33.294]
203	[6.5 ,33.294]	-27.4406	[-1.182e-12,-2.291e-04]	44.642837	[6.5 ,33.284]
204	[6.5 ,33.284]	-27.4402	[5.159e-11,1.013e-02]	1.008817	[6.5 ,33.294]
205	[6.5 ,33.294]	-27.4406	[-4.543e-13,-2.173e-04]	65.735150	[6.5 ,33.28]
206	[6.5 ,33.28]	-27.4398	[2.941e-11,1.432e-02]	0.991005	[6.5 ,33.294]
207	[6.5 ,33.294]	-27.4406	[2.651e-13,-1.265e-04]	62.493895	[6.5 ,33.286]
208	[6.5 ,33.286]	-27.4403	[-1.630e-11, 7.856e-03]	1.033641	[6.5 ,33.294]
209	[6.5 ,33.294]	-27.4406	[5.485e-13,-3.414e-04]	18.538515	[6.5 ,33.288]
210	[6.5 ,33.288]	-27.4404	[-9.619e-12, 6.033e-03]	1.107303	[6.5 ,33.294]
211	[6.5 ,33.294]	-27.4405	[1.032e-12,-6.924e-04]	5.338122	[6.5 ,33.291]
212	[6.5 ,33.291]	-27.4405	[-4.475e-12, 3.015e-03]	1.574779	[6.5 ,33.295]
213	[6.5 ,33.295]	-27.4405	[2.573e-12,-1.740e-03]	1.836916	[6.5 ,33.292]
214	[6.5 ,33.292]	-27.4405	[-2.154e-12, 1.455e-03]	2.126688	[6.5 ,33.295]
215	[6.5 ,33.295]	-27.4405	[2.427e-12,-1.639e-03]	1.123016	[6.5 ,33.294]
216	[6.5 ,33.294]	-27.4406	[-2.984e-13, 1.983e-04]	-1.172161	[6.5 ,33.293]
217	[6.5 ,33.293]	-27.4406	[-6.480e-13, 4.309e-04]	17.505673	[6.5 ,33.301]
218	[6.5 ,33.301]	-27.4404	[1.070e-11,-7.049e-03]	1.042427	[6.5 ,33.293]
219	[6.5 ,33.293]	-27.4406	[-4.540e-13, 2.362e-04]	53.224672	[6.5 ,33.306]
220	[6.5 ,33.306]	-27.4400	[2.371e-11,-1.215e-02]	1.017731	[6.5 ,33.294]
221	[6.5 ,33.294]	-27.4406	[-4.21e-13, 2.76e-05]	62.255668	[6.5 ,33.295]
222	[6.5 ,33.295]	-27.4405	[2.579e-11,-1.687e-03]	1.823365	[6.5 ,33.292]
223	[6.5 ,33.292]	-27.4405	[-2.123e-11, 1.388e-03]	1.953668	[6.5 ,33.295]
224	[6.5 ,33.295]	-27.4405	[2.025e-11,-1.324e-03]	2.897908	[6.5 ,33.291]
225	[6.5 ,33.291]	-27.4405	[-3.843e-11, 2.518e-03]	0.781250	[6.5 ,33.293]
226	[6.5 ,33.293]	-27.4405	[-8.407e-12, 5.432e-04]	9.674388	[6.5 ,33.298]
227	[6.5 ,33.298]	-27.4405	[7.292e-11,-4.685e-03]	1.074740	[6.5 ,33.293]
228	[6.5 ,33.293]	-27.4406	[-5.450e-12, 3.225e-04]	10.740089	[6.5 ,33.297]
229	[6.5 ,33.297]	-27.4405	[5.308e-11,-3.129e-03]	1.139253	[6.5 ,33.293]
230	[6.5 ,33.293]	-27.4406	[-7.391e-12, 4.236e-04]	22.070313	[6.5 ,33.303]
231	[6.5 ,33.303]	-27.4402	[1.557e-10,-8.827e-03]	1.049805	[6.5 ,33.293]
232	[6.5 ,33.293]	-27.4406	[-7.757e-12, 3.409e-04]	25.061508	[6.5 ,33.302]
233	[6.5 ,33.302]	-27.4403	[1.866e-10,-8.118e-03]	1.022718	[6.5 ,33.294]
234	[6.5 ,33.294]	-27.4406	[-4.240e-12, 1.008e-04]	49.877040	[6.5 ,33.299]

235	[6.5 ,33.299]	-27.4405	[2.072e-10,-4.899e-03]	1.141548	[6.5 ,33.293]
236	[6.5 ,33.293]	-27.4405	[-2.933e-11, 6.636e-04]	3.714282	[6.5 ,33.296]
237	[6.5 ,33.296]	-27.4405	[7.962e-11,-1.798e-03]	1.838951	[6.5 ,33.292]
238	[6.5 ,33.292]	-27.4405	[-6.680e-11, 1.507e-03]	2.838945	[6.5 ,33.296]
239	[6.5 ,33.296]	-27.4405	[1.228e-10,-2.764e-03]	1.375202	[6.5 ,33.293]
240	[6.5 ,33.293]	-27.4405	[-4.609e-11, 1.029e-03]	3.541574	[6.5 ,33.296]
241	[6.5 ,33.296]	-27.4405	[1.171e-10,-2.608e-03]	1.279591	[6.5 ,33.293]
242	[6.5 ,33.293]	-27.4405	[-3.275e-11, 7.211e-04]	9.440616	[6.5,33.3]
243	[6.5,33.3]	-27.4404	[2.764e-10,-6.041e-03]	1.086474	[6.5 ,33.293]
244	[6.5 ,33.293]	-27.4405	[-2.390e-11, 4.765e-04]	10.540674	[6.5 ,33.298]
245	[6.5 ,33.298]	-27.4405	[2.281e-10,-4.520e-03]	1.016187	[6.5 ,33.294]
246	[6.5 ,33.294]	-27.4406	[-3.691e-12, 4.733e-05]	80.468510	[6.5 ,33.297]
247	[6.5 ,33.297]	-27.4405	[2.933e-10,-3.744e-03]	1.140472	[6.5 ,33.293]
248	[6.5 ,33.293]	-27.4405	[-4.121e-11, 5.085e-04]	12.106454	[6.5 ,33.299]
249	[6.5 ,33.299]	-27.4404	[4.577e-10,-5.608e-03]	1.176454	[6.5 ,33.293]
250	[6.5 ,33.293]	-27.4405	[-8.076e-11, 9.509e-04]	7.812506	[6.5,33.3]
251	[6.5,33.3]	-27.4404	[5.501e-10,-6.427e-03]	1.120387	[6.5 ,33.293]
252	[6.5 ,33.293]	-27.4405	[-6.623e-11, 7.220e-04]	9.390618	[6.5,33.3]
253	[6.5,33.3]	-27.4404	[5.557e-10,-6.013e-03]	1.089574	[6.5 ,33.293]
254	[6.5 ,33.293]	-27.4405	[-4.978e-11, 4.931e-04]	11.594878	[6.5 ,33.299]
255	[6.5 ,33.299]	-27.4404	[5.274e-10,-5.191e-03]	1.107518	[6.5 ,33.293]
256	[6.5 ,33.293]	-27.4405	[-5.670e-11, 5.244e-04]	12.616582	[6.5,33.3]
257	[6.5,33.3]	-27.4404	[6.587e-10,-6.046e-03]	0.976179	[6.5 ,33.294]
258	[6.5 ,33.294]	-27.4406	[1.569e-11,-1.902e-04]	-3.186058	[6.5 ,33.294]
259	[6.5 ,33.294]	-27.4405	[6.568e-11,-7.956e-04]	4.044482	[6.5 ,33.291]
260	[6.5 ,33.291]	-27.4405	[-2.000e-10, 2.429e-03]	1.292392	[6.5 ,33.294]
261	[6.5 ,33.294]	-27.4405	[5.847e-11,-7.170e-04]	8.548274	[6.5 ,33.288]
262	[6.5 ,33.288]	-27.4404	[-4.413e-10, 5.448e-03]	0.988770	[6.5 ,33.294]
263	[6.5 ,33.294]	-27.4406	[-4.956e-12, 2.396e-05]	NaN	[]