

Optimization Practical Exercise

Sommersemester 2023

Donnermair Maximilian k12005908@students.jku.at
Fromherz Jakob k12011689@students.jku.at
Haslhofer Eva-Maria k12007773@students.jku.at
Scharnreitner Franz k12011695@students.jku.at
Weiss Hannah Maria k12021111@students.jku.at

12 May 2023

1 Exercise 1

1.1 Linesearch Algorithm with Wolfe-Powell Condition

```
1 function [x_new,f_new,g_new, exit_flag, alpha, eval] = LineSearch (f, x_old, f_old, g_old, p
   , phi_min, alpha_st)
2   %Linesearch algorithm with Wolfe-Powell Condition (Algorithm 4.2)
3
4   %Setting of parameters as described in the scriptum
5   tau = 0.1;
6   tau1 = 0.1;
7   tau2 = 0.6;
8   xi1 = 1;
9   xi2 = 10;
10  mu1 = 0.01;
11  mu2 = 0.9;
12  sigma = 0.9; %sigma greater than mu1
13
14  %if only 6 arguments are given
15  if nargin == 6
16      alpha_st = 1;
17  end
18
19  alpha_l = 0;
20  phi_l = f_old; %phi_l=phi(0)
21  x_old=x_old(:); % x_old always considered as column vector
22  dphi_l = dot(g_old,p);
23  exit_flag = 0;
24  flag = true; % means that alpha_r is infinity
25  alpha_r = 10^30; %alpha_r should be very large
26
27  alpha_tilde = 0; %initialization of alpha_tilde
28  eval = 0; %number of evaluations needed
29
30  while abs(alpha_r - alpha_l) > 10^(-15)
31      %evaluation of function, gradient and exit_flag
32      x_temp = x_old + alpha_st * p;
33      [f_temp, g_temp, exit_flag] = f(x_temp);
34      eval = eval + 1;
35
36      %function could not be evaluated (alpha_hat not in omega-prime)
37      if exit_flag ~= 0
38          alpha_r = alpha_st;
39          alpha_st = alpha_l + tau1 * (alpha_r - alpha_l);
40
41      %function could be evaluated (alpha_hat in omega-prime)
42      else
43          phi_hat = f_temp;
44
45          %if function smaller than phi_min, function is declared as
46          %unbounded
47          if phi_hat < phi_min
48              exit_flag = 2; % output is not minimum
49              fprintf("Error, unbounded function")
50              alpha = alpha_st;
51              x_new = x_temp;
52              f_new = f_temp;
53              return
54          end
55          %setting alpha_r and calculating alpha_st
56          if phi_hat > (f_old + mu1 * alpha_st * dphi_l)
57              flag = false; %alpha_r is not infinity
58              alpha_r = alpha_st;
59              length = alpha_r - alpha_l;
60              c = (phi_hat - phi_l - dphi_l*length)/ (length^2);
61              alpha_tilde = alpha_l - dphi_l/(2*c);
```

```

62         alpha_st = min(max(alpha_l + tau * length , alpha_tilde), alpha_r - tau *
63         length);
64     else
65         dphi_hat = dot(g_temp,p); %derivative of phi_hat
66         %calculating alpha_tilde
67         if dphi_hat < sigma * dphi_l
68             if flag
69                 if dphi_l/dphi_hat > (1 + xi2)/xi2
70                     alpha_tilde = alpha_st + (alpha_st - alpha_l) * max(dphi_hat/(
71                     dphi_l - dphi_hat),xi1);
72                 else
73                     alpha_tilde = alpha_st + xi2 * (alpha_st - alpha_l);
74                 end
75             else
76                 if dphi_l/dphi_hat > 1+(alpha_st-alpha_l)/(tau2*(alpha_r-alpha_st))
77                     alpha_tilde = alpha_st+max((alpha_st-alpha_l)*dphi_hat/(dphi_l-
78                     dphi_hat),tau1*(alpha_r-alpha_st));
79                 else
80                     alpha_tilde=alpha_st+tau2*(alpha_r-alpha_st);
81                 end
82             end
83             % setting alpha_l, phi_l, derivative of phi_l and alpha_st
84             alpha_l = alpha_st;
85             phi_l=phi_hat;
86             dphi_l = dphi_hat;
87             alpha_st=alpha_tilde;
88             %returning from algorithm
89         else
90             alpha = alpha_st;
91             x_new = x_old+alpha*p;
92             [f_new,g_new,exit_flag] = f(x_new);
93             eval = eval+1;
94             return
95         end
96     end
97 end
98 % final return (if there was no return before)
99 alpha = alpha_st;
100 x_new = x_old+alpha*p;
101 [f_new,g_new,exit_flag] = f(x_new);
102 eval = eval+1;
103 end

```

src/LineSearch.m

1.2 Method of steepest descent

```

1 function [x,f_val,g,exit_flag , iter , evals] = SteepestDescent (f, x0, phi_min,eps,itmax,
2     typ_f,typ_x)
3     %Steepest Descent Algorithm for testing LineSearch
4     %for variable input arguments
5     if nargin < 7
6         typ_x(1:length(x0)) = 10^-4;
7     end
8
9     if nargin < 6
10         typ_f = 10^-4;
11     end
12
13     if nargin < 5
14         itmax = 1000;
15     end

```

```

16
17 if nargin < 4
18     eps = 10^-6;
19 end
20
21 if nargin < 3
22     phi_min = -10^30;
23 end
24
25 %starting value for iteration, function value, gradient, exit_flag
26 xk = x0;
27 [fk,gk,exit_flag] = f(x0);
28 evals = 1;
29
30 %iterations of steepest descent
31 for iter = 1:itmax
32
33     %termination condition fullfilled (relative gradient less than tolerance)
34     if max(abs(gk) .* typ_x / typ_f) <= eps
35         x = xk;
36         f_val = fk;
37         g = gk;
38         return
39     end
40
41     %calling LineSearch
42     [xk,fk,gk,exit_flag,~,eval_temp] = LineSearch(f,xk,fk,gk,-gk,phi_min);
43     typ_f = max(typ_f,abs(fk));
44     typ_x = max(typ_x,abs(xk));
45     evals = evals + eval_temp; %updating number of evaluations
46 end
47
48 %Setting output, if termination condition is not fullfilled and maximal
49 %number of iterations reached
50 x = xk;
51 f_val = fk;
52 g = gk;
53 exit_flag = 1;
54
55 end

```

src/SteepestDescent.m

1.3 Testfunctions a) to d)

```

1 function [f_val, g, exit_flag] = f_a(x)
2     exit_flag=0;
3     x1=x(1);
4     x2=x(2);
5     f_val = 0.5*(x1+x2)^2 + 0.05*(x1-x2)^2;
6     g = [1.1*x1 + 0.9*x2; 0.9*x1 + 1.1*x2];
7 end

```

src/f.a.m

```

1 function [f_val, g, exit_flag] = f_b(x)
2     exit_flag=0;
3     x1=x(1);
4     x2=x(2);
5     f_val = 100*(x2-x1^2)^2 + (1-x1)^2;
6     g = [400*x1*(x1^2-x2) + 2*(x1-1); 200*(x2-x1^2)];
7 end

```

src/f.b.m

```

1 function [f_val, g, exit_flag] = f_c(x)
2     exit_flag=0;
3     n = length(x);
4     if min(x)<=0
5         f_val = 10^30; %dummy values
6         g = ones(n,1); %dummy values
7         exit_flag = 1;
8         return
9     end
10
11     x=x(:); %transform to column vector
12     %(1:n)', x, log(x) column vectors
13     f_val = sum((1:n)'.*x.*log(x)) + 1/sum(x);
14     g = zeros(n,1); %column vector
15     for i=1:n
16         g(i)=i*(log(x(i))+1) - 1/sum(x)^2;
17     end
18 end

```

src/f.c.m

```

1 function [f_val, g, exit_flag] = f_d(x)
2     exit_flag=0;
3     n = length(x);
4
5     x=x(:);
6     v=(1:n)';
7     A=(1/(100*n))*eye(n)+hilb(n);
8     xAx = x'*A*x;
9     f_val = (xAx)^2-(x'*A*v)^2;
10    g = 4*xAx*(A*x)-2*(x'*A*v)*A*v;
11 end

```

src/f.d.m

1.4 Testscript

```

1 diary test1.txt
2 diary on
3 disp("Funktion f_a test")
4
5 [x_val, f_val, g_val, ~, iter, evals] = SteepestDescent(@f_a,[10,-3]);
6 displayVals(x_val, f_val, g_val, iter, evals);
7
8 disp("_____")
9
10 disp("Funktion f_b test")
11
12 [x_val, f_val, g_val, ~, iter, evals]= SteepestDescent(@f_b,[-1,2],-10^30,10^-6,5000);
13 displayVals(x_val, f_val, g_val, iter, evals);
14
15 disp("_____")
16
17 disp("Funktion f_c test");
18 for j = [10 100 1000]
19     fprintf("For n = %d \n",j);
20     x0 = 5 * ones(1,j);
21     [x_val, f_val, g_val, ~, iter, evals] = SteepestDescent(@f_c,x0,-10^30,10^-9,10000);
22     displayVals(x_val, f_val, g_val, iter, evals);
23
24 end
25
26
27 disp("_____")
28

```

```

29 disp("Funktion f_d test");
30
31 for j = [10 100 1000]
32     fprintf("For n = %d \n",j);
33     x0 = zeros(j,1);
34     x0(j) = 10*j;
35     [x_val,f_val,g_val,~,iter,evals] = SteepestDescent(@f_d,x0,-10^30,10^-6,10000);
36     displayVals(x_val,f_val,g_val,iter,evals);
37
38 end
39
40 disp("_____")
41
42
43 diary off

```

src/testex1.m

With a quick printing function.

```

1 function retval = displayVals (x_val , f_val ,g_val ,iter ,evals)
2     if length(x_val) < 11
3         fprintf("Minimum computed at: ");
4         x_val
5
6
7     end
8
9     disp("Minimum at function f value: ");
10    f_val
11
12    disp("Norm of gradient: ");
13    no = norm(g_val);
14    no
15
16    fprintf("with %d iterations and %d function evals \n",iter,evals);
17
18
19 end

```

src/displayVals.m

We get the output

```

1 Funktion f_a test
2 Minimum computed at:
3 x_val =
4
5     1.0e-05 *
6
7     0.1724
8     -0.1690
9
10 Minimum at function f value:
11
12 f_val =
13
14     5.8328e-13
15
16 norm of gradient:
17
18 no =
19
20     4.8506e-07
21
22 with 71 iterations and 145 function evals
23
24 Funktion f_b test
25 Minimum computed at:

```

```

26 x_val =
27
28     1.0000
29     1.0000
30
31 Minimum at function f value:
32
33 f_val =
34
35     4.2321e-11
36
37 norm of gradient:
38
39 no =
40
41     6.1267e-06
42
43 with 4614 iterations and 20811 function evals
44
45 Funktion f_c test
46 For n = 10
47 Minimum computed at:
48 x_val =
49
50     0.3949
51     0.3811
52     0.3767
53     0.3745
54     0.3731
55     0.3722
56     0.3716
57     0.3712
58     0.3708
59     0.3705
60
61 Minimum at function f value:
62
63 f_val =
64
65    -19.9644
66
67 norm of gradient:
68
69 no =
70
71     7.6904e-08
72
73 with 89 iterations and 345 function evals
74 For n = 100
75 Minimum at function f value:
76
77 f_val =
78
79    -1.8578e+03
80
81 norm of gradient:
82
83 no =
84
85     2.2556e-06
86
87 with 461 iterations and 2055 function evals
88 For n = 1000
89 Minimum at function f value:
90
91 f_val =
92
93    -1.8412e+05

```

```

94
95 norm of gradient:
96
97 no =
98
99     4.8703e-04
100
101 with 3413 iterations and 18742 function evals
102
103 Funktion f_d test
104 For n = 10
105 Minimum computed at:
106 x_val =
107
108     0.7074
109     1.4125
110     2.1185
111     2.8354
112     3.5448
113     4.2462
114     4.9424
115     5.6360
116     6.3286
117     7.1201
118
119 Minimum at function f value:
120
121 f_val =
122
123     -1.6770e+04
124
125 norm of gradient:
126
127 no =
128
129     0.0436
130
131 with 10000 iterations and 50003 function evals
132 For n = 100
133 Minimum at function f value:
134
135 f_val =
136
137     -1.0983e+10
138
139 norm of gradient:
140
141 no =
142
143     3.6507e+04
144
145 with 10000 iterations and 80002 function evals
146
147 Funktion f_d test
148 For n = 1000
149 Minimum at function f value:
150
151 f_val =
152
153     -1.0513e+16
154
155 norm of gradient:
156
157 no =
158
159     7.1347e+07
160
161 with 10000 iterations and 110001 function evals

```


1.5 Interpretation

We can see, that the SteepestDescent works for f_a relatively well, while to get a relatively good result for f_b we already need > 4000 iterations. f_c works quite well, although the number of iterations needed grows fast with increase in dimension size. The algorithm fails to work for f_d however for big dimensions.

We want to analyze this behavior with Theorem 4.5 of our script. Therefore, we calculate with the help of the functions f_aH to f_dH from Exercise 2 the condition numbers in our calculated minimum. If we are not too far away from the minimum we get a good approximation of the hessian (because the hessian is continuous, furthermore we can see, that the condition number oh the Hessians at the calculated minima in exercise two are roughly the same, especially in order of magnitude) and therefore the condition number is near the calculated value. Then we calculate the upper bound of the convergence rate $Q1(f^k) < 1 - \eta * \frac{4*\kappa}{(1+\kappa)^2}$. For the chosen values $\mu_1 = 0.01$ and $\sigma = 0.9$, we get $\eta = 0.0396$.

We get the output:

```

1 >> testex1_cond
2
3 Funktion f_a test
4 Condition number of H in calculated minimum  10.0000
5 Upper bound for convergence rate 0.9869
6
7 Funktion f_b test
8 Condition number of H in calculated minimum  2507.9466
9 Upper bound for convergence rate 0.9999
10
11 Funktion f_c test
12 For n = 10
13 Condition number of H in calculated minimum  10.5232
14 Upper bound for convergence rate 0.9874
15
16 For n = 100
17 Condition number of H in calculated minimum  100.0715
18 Upper bound for convergence rate 0.9984
19
20 For n = 1000
21 Condition number of H in calculated minimum  1000.3861
22 Upper bound for convergence rate 0.9998
23
24
25 Funktion f_d test
26 For n = 10
27 Condition number of H in calculated minimum  1256.3686
28 Upper bound for convergence rate 0.9999
29
30 For n = 100
31 Condition number of H in calculated minimum  15456.3922
32 Upper bound for convergence rate 1.0000
33
34 For n = 1000
35 Condition number of H in calculated minimum  203338.7870
36 Upper bound for convergence rate 1.0000

```

The condition number of the hessian of our calculated minimum of f_a is 10 and the theorem tells us that the convergence rate is smaller than 0.9869. It converges very fast (our theorem only gives us an upper bound). The condition number of the hessian of f_b in the calculated minimum is 2508, the upper bound for the convergence rate is 0.9999. The algorithm converges quite slowly. The condition number of the hessian of f_c for n=10 in the calculated minimum is 10.5233, the upper bound 0.9874. This is better than for f_b. When we increase n the condition matrix and the upper bound increases. The condition number of the hessian of

f_d for n=10 in the calculated minimum is 1256, the upper bound 0.9999, quite near to 1. As we increase the dimension the upper bound is rounded to 1.

2 Exercise 2

2.1 Testfunctions a) to d) with Hessian

```

1 function [f_val, g, H] = f_aH(x)
2     x1=x(1);
3     x2=x(2);
4     f_val = 0.5*(x1+x2)^2 + 0.05*(x1-x2)^2;
5     g = [1.1*x1+0.9*x2;0.9*x1+1.1*x2];
6     if nargin > 2
7         H = [1.1,0.9;0.9,1.1];
8     end
9 end

```

src/f_aH.m

```

1 function [f_val, g, H] = f_bH(x)
2     x1=x(1);
3     x2=x(2);
4     f_val = 100*(x2-x1^2)^2 + (1-x1)^2;
5     g = [400*x1*(x1^2-x2) + 2*(x1-1); 200*(x2-x1^2)];
6     if nargin > 2
7         H = [1200*x1^2 - 400*x2 + 2, -400*x1; -400*x1, 200];
8     end
9 end

```

src/f_bH.m

```

1 function [f_val, g, H] = f_cH(x)
2     n = length(x);
3     if min(x)<=0
4         f_val = 10^30; %dummy values
5         g = ones(n,1); %dummy values
6         H = zeros(n);
7         return
8     end
9
10    x=x(:); %transform to column vector
11    %(1:n)', x, log(x) column vectors
12    f_val = sum((1:n)'.*x.*log(x)) + 1/sum(x);
13    g = zeros(n,1); %column vector
14    for i=1:n
15        g(i) = i*(reallog(x(i))+1) - 1/sum(x)^2;
16    end
17    if nargin > 2
18        h = 2/sum(x)^3;
19        H = h*ones(n,n) + diag((1:n)'./x,0);
20    end
21 end

```

src/f_cH.m

```

1 function [f_val, g, H] = f_dH(x)
2     n = length(x);
3
4     x=x(:);
5     v=(1:n)';
6     A=(1/(100*n))*eye(n)+hilb(n);
7     xAx = x'*A*x;
8     f_val = (xAx)^2-(x'*A*v)^2;
9

```

```

10     if nargin > 1
11         g = 4*xAx*(A*x) - 2*(x'*A*v)*A*v;
12         if nargin > 2
13             H = 4*(2*A*x + (A*x)' + x'*A*x*A) - 2*A*v*(A*v)';
14         end
15     end
16
17 end

```

src/f_dH.m

2.2 Tests in Exercise 2

```

1 dfile1 = 'Test1.txt';
2 if exist(dfile1, 'file') ; delete(dfile1); end
3 diary(dfile1)
4 diary on
5
6 options1= optimset('LargeScale','off','GradObj','on','Display','off');
7 options2= optimset('LargeScale','on','GradObj','on','Hessian','off','Display','off');
8 options3= optimset('Algorithm','trust-region','LargeScale','on','GradObj','on','Hessian','on',
9     '','Display','off');
10 options = [options1, options2, options3];
11
12 %(a)
13 disp('=====')
14 disp(' (A) ')
15 disp('=====')
16 for i=1:length(options)
17     disp('Testing objective function f_a with options: ')
18     disp(options(i));
19     if i==3
20         [x,f,~,out,g,H] = fminunc(@f_aH,[10,-3],options(i));
21     else
22         [x,f,~,out,g] = fminunc(@f_aH,[10,-3],options(i));
23     end
24     disp('Calculated minimizer: ');
25     x
26     disp('Calculated Minimum value: ' + f);
27     disp('Norm of gradient at calculated Minimum: ' + norm(g));
28     if i==3
29         disp('Condition of Hessian at calculated Minimum: ' + cond(H))
30     end
31     disp(out);
32     disp('=====')
33 end
34
35 disp('=====')
36 disp(' (B) ')
37 disp('=====')
38 for i=1:length(options)
39     disp('Testing objective function f_b with options: ')
40     disp(options(i));
41
42     if i==3
43         [x,f,~,out,g,H] = fminunc(@f_bH,[-1,2],options(i));
44     else
45         [x,f,~,out,g] = fminunc(@f_bH,[-1,2],options(i));
46     end
47     disp('Calculated minimizer: ');
48     x
49     disp('Calculated Minimum value: ' + f);
50     disp('Norm of gradient at calculated Minimum: ' + norm(g));
51     if i==3
52         disp('Condition of Hessian at calculated Minimum: ' + cond(H))

```

```

53     end
54     disp(out)
55     disp("=====")
56 end
57
58
59
60 disp("=====")
61 disp("C")
62 disp("=====")
63
64 for i=1:length(options)
65     disp("Testing objective function f_c with options: ")
66     disp(options(i));
67
68     disp("=====")
69     for j=1:3
70         n=10^j;
71         disp("Results for n= "+n);
72         if i==3
73             [x,f,~,out,g,H] = fminunc(@f_cH,5*ones(1,n),options(i));
74         else
75             [x,f,~,out,g] = fminunc(@f_cH,5*ones(1,n),options(i));
76         end
77         if j==1
78             disp("Calculated minimizer: ");
79             x
80         end
81         disp("Calculated Minimum value: " + f);
82         disp("Norm of gradient at calculated Minimum: " + norm(g));
83         if i==3
84             disp("Condition of Hessian at calculated Minimum: " + cond(H))
85         end
86         disp(out)
87         disp("=====")
88     end
89 end
90
91
92 disp("=====")
93 disp(" (D) ")
94 disp("=====")
95 for i=1:length(options)
96     disp("Testing objective function f_d with options: ")
97     disp(options(i));
98
99     disp("=====")
100    for j=1:3
101        n=10^j;
102        disp("Results for n= "+n);
103        if i==3
104            [x,f,~,out,g,H] = fminunc(@f_dH,[zeros(1,n-1),10*n],options(i));
105        else
106            [x,f,~,out,g] = fminunc(@f_dH,[zeros(1,n-1),10*n],options(i));
107        end
108        if j==1
109            disp("Calculated minimizer: ");
110            x
111        end
112        disp("Calculated Minimum value: " + f);
113        disp("Norm of gradient at calculated Minimum: " + norm(g));
114        if i==3
115            disp("Condition of Hessian at calculated Minimum: " + cond(H))
116        end
117        disp(out)
118        disp("=====")
119    end
120 end

```

```

121 end
122
123 diary off

```

src/Prog1Ex2.m

2.3 Test output

As can be observed, for every testfunction and every given starting value `fminunc` was called with the three required options. Running `Prog1Ex2.m` produces the output below (console output logged using `Matlab-diary` functionality)

```

1 =====
2 (A)
3 =====
4 Testing objective function f_a with options:
5     Display: 'off'
6     MaxFunEvals: []
7     MaxIter: []
8     TolFun: []
9     TolX: []
10    FunValCheck: []
11    OutputFcn: []
12    PlotFcns: []
13    ActiveConstrTol: []
14    Algorithm: []
15    AlwaysHonorConstraints: []
16    DerivativeCheck: []
17    Diagnostics: []
18    DiffMaxChange: []
19    DiffMinChange: []
20    FinDiffRelStep: []
21    FinDiffType: []
22    GoalsExactAchieve: []
23    GradConstr: []
24    GradObj: 'on'
25    HessFcn: []
26    Hessian: []
27    HessMult: []
28    HessPattern: []
29    HessUpdate: []
30    InitBarrierParam: []
31    InitTrustRegionRadius: []
32    Jacobian: []
33    JacobMult: []
34    JacobPattern: []
35    LargeScale: 'off'
36    MaxNodes: []
37    MaxPCGIter: []
38    MaxProjCGIter: []
39    MaxSQPIter: []
40    MaxTime: []
41    MeritFunction: []
42    MinAbsMax: []
43    NoStopIfFlatInfeas: []
44    ObjectiveLimit: []
45    PhaseOneTotalScaling: []
46    Preconditioner: []
47    PrecondBandWidth: []
48    RelLineSrchBnd: []
49    RelLineSrchBndDuration: []
50    ScaleProblem: []
51    SubproblemAlgorithm: []
52    TolCon: []
53    TolConSQP: []
54    TolGradCon: []

```

```

55         TolPCG: []
56         TolProjCG: []
57         TolProjCGAbs: []
58         TypicalX: []
59         UseParallel: []
60
61 Calculated minimizer:
62
63 x =
64
65     1.0e-06 *
66
67     -0.2002     0.6078
68
69 Calculated Minimum value: 1.1571e-13
70 Norm of gradient at calculated Minimum: 5.8763e-07
71     iterations: 8
72     funcCount: 9
73     stepsize: 1.8956e-04
74     lssteplength: 1
75     firstorderopt: 4.8838e-07
76     algorithm: 'quasi-newton'
77     message: 'Local minimum found. Optimization completed because the size of the
gradient is less than the default value of the optimality tolerance. Stopping
criteria details: Optimization completed: The first-order optimality measure,
5.251446e-08, is less than options.OptimalityTolerance = 1.000000e-06.
Optimization Metric Options relative
norm(gradient) = 5.25e-08 OptimalityTolerance = 1e-06 (default)'
78
79
80 Testing objective function f_a with options:
81     Display: 'off'
82     MaxFunEvals: []
83     MaxIter: []
84     TolFun: []
85     TolX: []
86     FunValCheck: []
87     OutputFcn: []
88     PlotFcns: []
89     ActiveConstrTol: []
90     Algorithm: []
91     AlwaysHonorConstraints: []
92     DerivativeCheck: []
93     Diagnostics: []
94     DiffMaxChange: []
95     DiffMinChange: []
96     FinDiffRelStep: []
97     FinDiffType: []
98     GoalsExactAchieve: []
99     GradConstr: []
100     GradObj: 'on'
101     HessFcn: []
102     Hessian: 'off'
103     HessMult: []
104     HessPattern: []
105     HessUpdate: []
106     InitBarrierParam: []
107     InitTrustRegionRadius: []
108     Jacobian: []
109     JacobMult: []
110     JacobPattern: []
111     LargeScale: 'on'
112     MaxNodes: []
113     MaxPCGIter: []
114     MaxProjCGIter: []
115     MaxSQPIter: []
116     MaxTime: []
117     MeritFunction: []

```

```

118         MinAbsMax: []
119         NoStopIfFlatInfeas: []
120         ObjectiveLimit: []
121         PhaseOneTotalScaling: []
122         Preconditioner: []
123         PrecondBandWidth: []
124         RelLineSrchBnd: []
125         RelLineSrchBndDuration: []
126         ScaleProblem: []
127         SubproblemAlgorithm: []
128         TolCon: []
129         TolConSQP: []
130         TolGradCon: []
131         TolPCG: []
132         TolProjCG: []
133         TolProjCGAbs: []
134         TypicalX: []
135         UseParallel: []
136
137 Calculated minimizer:
138
139 x =
140
141     1.0e-06 *
142
143     -0.2002     0.6078
144
145 Calculated Minimum value: 1.1571e-13
146 Norm of gradient at calculated Minimum: 5.8763e-07
147     iterations: 8
148     funcCount: 9
149     stepsize: 1.8956e-04
150     lssteplength: 1
151     firstorderopt: 4.8838e-07
152     algorithm: 'quasi-newton'
153     message: 'Local minimum found. Optimization completed because the size of the
154             gradient is less than the default value of the optimality tolerance. Stopping
155             criteria details: Optimization completed: The first-order optimality measure,
156             5.251446e-08, is less than options.OptimalityTolerance = 1.000000e-06.
157             Optimization Metric Options relative
158             norm(gradient) = 5.25e-08 OptimalityTolerance = 1e-06 (default)'
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180

```

```

156 Testing objective function f_a with options:
157     Display: 'off'
158     MaxFunEvals: []
159     MaxIter: []
160     TolFun: []
161     TolX: []
162     FunValCheck: []
163     OutputFcn: []
164     PlotFcns: []
165     ActiveConstrTol: []
166     Algorithm: 'trust-region'
167     AlwaysHonorConstraints: []
168     DerivativeCheck: []
169     Diagnostics: []
170     DiffMaxChange: []
171     DiffMinChange: []
172     FinDiffRelStep: []
173     FinDiffType: []
174     GoalsExactAchieve: []
175     GradConstr: []
176     GradObj: 'on'
177     HessFcn: []
178     Hessian: 'on'
179     HessMult: []
180     HessPattern: []

```

```

181         HessUpdate: []
182         InitBarrierParam: []
183         InitTrustRegionRadius: []
184         Jacobian: []
185         JacobMult: []
186         JacobPattern: []
187         LargeScale: 'on'
188         MaxNodes: []
189         MaxPCGIter: []
190         MaxProjCGIter: []
191         MaxSQPIter: []
192         MaxTime: []
193         MeritFunction: []
194         MinAbsMax: []
195         NoStopIfFlatInfeas: []
196         ObjectiveLimit: []
197         PhaseOneTotalScaling: []
198         Preconditioner: []
199         PrecondBandWidth: []
200         RelLineSrchBnd: []
201         RelLineSrchBndDuration: []
202         ScaleProblem: []
203         SubproblemAlgorithm: []
204         TolCon: []
205         TolConSQP: []
206         TolGradCon: []
207         TolPCG: []
208         TolProjCG: []
209         TolProjCGAbs: []
210         TypicalX: []
211         UseParallel: []
212
213 Calculated minimizer:
214
215 x =
216
217     1.0e-03 *
218
219     0.7869    -0.2361
220
221 Calculated Minimum value: 2.0406e-07
222 Norm of gradient at calculated Minimum: 0.00079236
223 Condition of Hessian at calculated Minimum: 10
224     iterations: 12
225     funcCount: 13
226     stepsize: 0.0025
227     cgiterations: 12
228     firstorderopt: 6.5317e-04
229     algorithm: 'trust-region'
230     message: 'Local minimum possible. fminunc stopped because the final change
in function value relative to its initial value is less than the default value of the
function tolerance. Stopping criteria details: Optimization stopped because
the relative objective function value is changing by less than options.
FunctionTolerance = 1.000000e-06. Optimization Metric
Options relative change objective = 7.82e-07 FunctionTolerance =
1e-06 (default)'
231     constrviolation: []
232
233 =====
234 =====
235 (B)
236 =====
237 Testing objective function f_b with options:
238     Display: 'off'
239     MaxFunEvals: []
240     MaxIter: []
241     TolFun: []
242     TolX: []

```



```

243         FunValCheck: []
244         OutputFcn: []
245         PlotFcns: []
246         ActiveConstrTol: []
247         Algorithm: []
248     AlwaysHonorConstraints: []
249         DerivativeCheck: []
250         Diagnostics: []
251         DiffMaxChange: []
252         DiffMinChange: []
253         FinDiffRelStep: []
254         FinDiffType: []
255     GoalsExactAchieve: []
256         GradConstr: []
257             GradObj: 'on'
258             HessFcn: []
259             Hessian: []
260             HessMult: []
261             HessPattern: []
262             HessUpdate: []
263         InitBarrierParam: []
264     InitTrustRegionRadius: []
265         Jacobian: []
266         JacobMult: []
267         JacobPattern: []
268         LargeScale: 'off'
269         MaxNodes: []
270         MaxPCGIter: []
271         MaxProjCGIter: []
272         MaxSQPIter: []
273         MaxTime: []
274         MeritFunction: []
275         MinAbsMax: []
276     NoStopIfFlatInfeas: []
277         ObjectiveLimit: []
278     PhaseOneTotalScaling: []
279         Preconditioner: []
280         PrecondBandWidth: []
281         RelLineSrchBnd: []
282     RelLineSrchBndDuration: []
283         ScaleProblem: []
284     SubproblemAlgorithm: []
285         TolCon: []
286         TolConSQP: []
287         TolGradCon: []
288         TolPCG: []
289         TolProjCG: []
290     TolProjCGAbs: []
291         TypicalX: []
292     UseParallel: []
293
294 Calculated minimizer:
295
296 x =
297
298     1.0000     1.0000
299
300 Calculated Minimum value: 6.6115e-11
301 Norm of gradient at calculated Minimum: 0.00029178
302     iterations: 39
303     funcCount: 50
304     stepsize: 4.0089e-04
305     lssteplength: 1
306     firstorderopt: 2.6299e-04
307     algorithm: 'quasi-newton'
308     message: 'Local minimum found. Optimization completed because the size of the
gradient is less than the default value of the optimality tolerance. Stopping
criteria details: Optimization completed: The first-order optimality measure,

```

```

309         6.624363e-07, is less than options.OptimalityTolerance = 1.000000e-06.
310         Optimization Metric Options relative
311         norm(gradient) = 6.62e-07 OptimalityTolerance = 1e-06 (default)'
312
313 =====
314
315 Testing objective function f_b with options:
316     Display: 'off'
317     MaxFunEvals: []
318     MaxIter: []
319     TolFun: []
320     TolX: []
321     FunValCheck: []
322     OutputFcn: []
323     PlotFcns: []
324     ActiveConstrTol: []
325     Algorithm: []
326     AlwaysHonorConstraints: []
327     DerivativeCheck: []
328     Diagnostics: []
329     DiffMaxChange: []
330     DiffMinChange: []
331     FinDiffRelStep: []
332     FinDiffType: []
333     GoalsExactAchieve: []
334     GradConstr: []
335     GradObj: 'on'
336     HessFcn: []
337     Hessian: 'off'
338     HessMult: []
339     HessPattern: []
340     HessUpdate: []
341     InitBarrierParam: []
342     InitTrustRegionRadius: []
343     Jacobian: []
344     JacobMult: []
345     JacobPattern: []
346     LargeScale: 'on'
347     MaxNodes: []
348     MaxPCGIter: []
349     MaxProjCGIter: []
350     MaxSQPIter: []
351     MaxTime: []
352     MeritFunction: []
353     MinAbsMax: []
354     NoStopIfFlatInfeas: []
355     ObjectiveLimit: []
356     PhaseOneTotalScaling: []
357     Preconditioner: []
358     PrecondBandWidth: []
359     RelLineSrchBnd: []
360     RelLineSrchBndDuration: []
361     ScaleProblem: []
362     SubproblemAlgorithm: []
363     TolCon: []
364     TolConSQP: []
365     TolGradCon: []
366     TolPCG: []
367     TolProjCG: []
368     TolProjCGAbs: []
369     TypicalX: []
370     UseParallel: []
371
372 Calculated minimizer:
373
374 x =
375
376     1.0000     1.0000
377
378

```

```

374 Calculated Minimum value: 6.6115e-11
375 Norm of gradient at calculated Minimum: 0.00029178
376     iterations: 39
377     funcCount: 50
378     stepsize: 4.0089e-04
379     lssteplength: 1
380     firstorderopt: 2.6299e-04
381     algorithm: 'quasi-newton'
382     message: 'Local minimum found. Optimization completed because the size of the
gradient is less than the default value of the optimality tolerance. Stopping
criteria details: Optimization completed: The first-order optimality measure,
6.624363e-07, is less than options.OptimalityTolerance = 1.000000e-06.
Optimization Metric Options relative
norm(gradient) = 6.62e-07 OptimalityTolerance = 1e-06 (default)'
```

```

384
385 Testing objective function f_b with options:
386     Display: 'off'
387     MaxFunEvals: []
388     MaxIter: []
389     TolFun: []
390     TolX: []
391     FunValCheck: []
392     OutputFcn: []
393     PlotFcns: []
394     ActiveConstrTol: []
395     Algorithm: 'trust-region'
396     AlwaysHonorConstraints: []
397     DerivativeCheck: []
398     Diagnostics: []
399     DiffMaxChange: []
400     DiffMinChange: []
401     FinDiffRelStep: []
402     FinDiffType: []
403     GoalsExactAchieve: []
404     GradConstr: []
405     GradObj: 'on'
406     HessFcn: []
407     Hessian: 'on'
408     HessMult: []
409     HessPattern: []
410     HessUpdate: []
411     InitBarrierParam: []
412     InitTrustRegionRadius: []
413     Jacobian: []
414     JacobMult: []
415     JacobPattern: []
416     LargeScale: 'on'
417     MaxNodes: []
418     MaxPCGIter: []
419     MaxProjCGIter: []
420     MaxSQPIter: []
421     MaxTime: []
422     MeritFunction: []
423     MinAbsMax: []
424     NoStopIfFlatInfeas: []
425     ObjectiveLimit: []
426     PhaseOneTotalScaling: []
427     Preconditioner: []
428     PrecondBandWidth: []
429     RelLineSrchBnd: []
430     RelLineSrchBndDuration: []
431     ScaleProblem: []
432     SubproblemAlgorithm: []
433     TolCon: []
434     TolConSQP: []
435     TolGradCon: []
436     TolPCG: []

```

```

437         TolProjCG: []
438         TolProjCGAbs: []
439         TypicalX: []
440         UseParallel: []
441
442 Calculated minimizer:
443
444 x =
445
446     1.0000     1.0000
447
448 Calculated Minimum value: 1.931e-17
449 Norm of gradient at calculated Minimum: 3.3208e-08
450 Condition of Hessian at calculated Minimum: 2508.0095
451     iterations: 30
452     funcCount: 31
453     stepsize: 2.3363e-05
454     cgiterations: 25
455     firstorderopt: 2.7783e-08
456     algorithm: 'trust-region'
457     message: 'Local minimum found. Optimization completed because the size of
the gradient is less than the default value of the optimality tolerance. Stopping
criteria details: Optimization completed: The first-order optimality measure,
2.778283e-08, is less than options.OptimalityTolerance = 1.000000e-06, and no
negative/zero curvature is detected in the trust-region model. Optimization
Metric Options relative first-order optimality
= 2.78e-08 OptimalityTolerance = 1e-06 (default)'
458     constrviolation: []
459
460 =====
461 =====
462 C
463 =====
464
465 Testing objective function f_c with options:
466     Display: 'off'
467     MaxFunEvals: []
468     MaxIter: []
469     TolFun: []
470     TolX: []
471     FunValCheck: []
472     OutputFcn: []
473     PlotFcns: []
474     ActiveConstrTol: []
475     Algorithm: []
476     AlwaysHonorConstraints: []
477     DerivativeCheck: []
478     Diagnostics: []
479     DiffMaxChange: []
480     DiffMinChange: []
481     FinDiffRelStep: []
482     FinDiffType: []
483     GoalsExactAchieve: []
484     GradConstr: []
485     GradObj: 'on'
486     HessFcn: []
487     Hessian: []
488     HessMult: []
489     HessPattern: []
490     HessUpdate: []
491     InitBarrierParam: []
492     InitTrustRegionRadius: []
493     Jacobian: []
494     JacobMult: []
495     JacobPattern: []
496     LargeScale: 'off'
497     MaxNodes: []
498     MaxPCGIter: []
499     MaxProjCGIter: []

```

```

499         MaxSQPIter: []
500         MaxTime: []
501         MeritFunction: []
502         MinAbsMax: []
503         NoStopIfFlatInfeas: []
504         ObjectiveLimit: []
505         PhaseOneTotalScaling: []
506         Preconditioner: []
507         PrecondBandWidth: []
508         RelLineSrchBnd: []
509         RelLineSrchBndDuration: []
510         ScaleProblem: []
511         SubproblemAlgorithm: []
512         TolCon: []
513         TolConSQP: []
514         TolGradCon: []
515         TolPCG: []
516         TolProjCG: []
517         TolProjCGAbs: []
518         TypicalX: []
519         UseParallel: []
520
521
522 Results for n= 10
523 Calculated minimizer:
524
525 x =
526
527     0.3949    0.3811    0.3767    0.3745    0.3731    0.3722    0.3716    0.3712    0.3708
528     0.3705
529 Calculated Minimum value: -19.9644
530 Norm of gradient at calculated Minimum: 1.0378e-05
531     iterations: 29
532     funcCount: 55
533     stepsize: 3.0515e-06
534     lssteplength: 1
535     firstorderopt: 8.0412e-06
536     algorithm: 'quasi-newton'
537     message: 'Local minimum found. Optimization completed because the size of the
538             gradient is less than the default value of the optimality tolerance. Stopping
539             criteria details: Optimization completed: The first-order optimality measure,
540             2.967900e-07, is less than options.OptimalityTolerance = 1.000000e-06.
541             Optimization Metric Options relative
542             norm(gradient) = 2.97e-07 OptimalityTolerance = 1e-06 (default)'
543
544
545 Results for n= 100
546 Calculated Minimum value: -1857.764
547 Norm of gradient at calculated Minimum: 0.00082223
548     iterations: 79
549     funcCount: 160
550     stepsize: 2.0476e-05
551     lssteplength: 0.1895
552     firstorderopt: 1.7558e-04
553     algorithm: 'quasi-newton'
554     message: 'Local minimum found. Optimization completed because the size of the
555             gradient is less than the default value of the optimality tolerance. Stopping
556             criteria details: Optimization completed: The first-order optimality measure,
557             6.702941e-07, is less than options.OptimalityTolerance = 1.000000e-06.
558             Optimization Metric Options relative
559             norm(gradient) = 6.70e-07 OptimalityTolerance = 1e-06 (default)'
560
561
562 Results for n= 1000
563 Calculated Minimum value: -184123.6576
564 Norm of gradient at calculated Minimum: 0.010364
565     iterations: 203

```

```

556         funcCount: 408
557         stepsize: 2.7483e-05
558         lssteplength: 0.2436
559         firstorderopt: 0.0019
560         algorithm: 'quasi-newton'
561         message: 'Local minimum found. Optimization completed because the size of the
gradient is less than the default value of the optimality tolerance. Stopping
criteria details: Optimization completed: The first-order optimality measure,
7.088470e-07, is less than options.OptimalityTolerance = 1.000000e-06.
Optimization Metric Options relative
norm(gradient) = 7.09e-07 OptimalityTolerance = 1e-06 (default)'
```

```

563
564 Testing objective function f.c with options:
565         Display: 'off'
566         MaxFunEvals: []
567         MaxIter: []
568         TolFun: []
569         TolX: []
570         FunValCheck: []
571         OutputFcn: []
572         PlotFcns: []
573         ActiveConstrTol: []
574         Algorithm: []
575         AlwaysHonorConstraints: []
576         DerivativeCheck: []
577         Diagnostics: []
578         DiffMaxChange: []
579         DiffMinChange: []
580         FinDiffRelStep: []
581         FinDiffType: []
582         GoalsExactAchieve: []
583         GradConstr: []
584         GradObj: 'on'
585         HessFcn: []
586         Hessian: 'off'
587         HessMult: []
588         HessPattern: []
589         HessUpdate: []
590         InitBarrierParam: []
591         InitTrustRegionRadius: []
592         Jacobian: []
593         JacobMult: []
594         JacobPattern: []
595         LargeScale: 'on'
596         MaxNodes: []
597         MaxPCGIter: []
598         MaxProjCGIter: []
599         MaxSQPIter: []
600         MaxTime: []
601         MeritFunction: []
602         MinAbsMax: []
603         NoStopIfFlatInfeas: []
604         ObjectiveLimit: []
605         PhaseOneTotalScaling: []
606         Preconditioner: []
607         PrecondBandWidth: []
608         RelLineSrchBnd: []
609         RelLineSrchBndDuration: []
610         ScaleProblem: []
611         SubproblemAlgorithm: []
612         TolCon: []
613         TolConSQP: []
614         TolGradCon: []
615         TolPCG: []
616         TolProjCG: []
617         TolProjCGAbs: []
618         TypicalX: []

```

```

619         UseParallel: []
620
621
622 Results for n= 10
623 Calculated minimizer:
624
625 x =
626
627     0.3949    0.3811    0.3767    0.3745    0.3731    0.3722    0.3716    0.3712    0.3708
628     0.3705
629
630 Calculated Minimum value: -19.9644
631 Norm of gradient at calculated Minimum: 1.0378e-05
632     iterations: 29
633     funcCount: 55
634     stepsize: 3.0515e-06
635     lssteplength: 1
636     firstorderopt: 8.0412e-06
637     algorithm: 'quasi-newton'
638     message: 'Local minimum found. Optimization completed because the size of the
639             gradient is less than the default value of the optimality tolerance. Stopping
640             criteria details: Optimization completed: The first-order optimality measure,
641             2.967900e-07, is less than options.OptimalityTolerance = 1.000000e-06.
642             Optimization Metric Options relative
643             norm(gradient) = 2.97e-07 OptimalityTolerance = 1e-06 (default)'
644
645
646 Results for n= 100
647 Calculated Minimum value: -1857.764
648 Norm of gradient at calculated Minimum: 0.00082223
649     iterations: 79
650     funcCount: 160
651     stepsize: 2.0476e-05
652     lssteplength: 0.1895
653     firstorderopt: 1.7558e-04
654     algorithm: 'quasi-newton'
655     message: 'Local minimum found. Optimization completed because the size of the
656             gradient is less than the default value of the optimality tolerance. Stopping
657             criteria details: Optimization completed: The first-order optimality measure,
658             6.702941e-07, is less than options.OptimalityTolerance = 1.000000e-06.
659             Optimization Metric Options relative
660             norm(gradient) = 6.70e-07 OptimalityTolerance = 1e-06 (default)'
661
662
663 Results for n= 1000
664 Calculated Minimum value: -184123.6576
665 Norm of gradient at calculated Minimum: 0.010364
666     iterations: 203
667     funcCount: 408
668     stepsize: 2.7483e-05
669     lssteplength: 0.2436
670     firstorderopt: 0.0019
671     algorithm: 'quasi-newton'
672     message: 'Local minimum found. Optimization completed because the size of the
673             gradient is less than the default value of the optimality tolerance. Stopping
674             criteria details: Optimization completed: The first-order optimality measure,
675             7.088470e-07, is less than options.OptimalityTolerance = 1.000000e-06.
676             Optimization Metric Options relative
677             norm(gradient) = 7.09e-07 OptimalityTolerance = 1e-06 (default)'
678
679
680 Testing objective function f_c with options:
681     Display: 'off'
682     MaxFunEvals: []
683     MaxIter: []
684     TolFun: []
685     TolX: []
686     FunValCheck: []

```

```

671         OutputFcn: []
672         PlotFcns: []
673         ActiveConstrTol: []
674         Algorithm: 'trust-region'
675     AlwaysHonorConstraints: []
676         DerivativeCheck: []
677         Diagnostics: []
678         DiffMaxChange: []
679         DiffMinChange: []
680         FinDiffRelStep: []
681         FinDiffType: []
682         GoalsExactAchieve: []
683         GradConstr: []
684             GradObj: 'on'
685             HessFcn: []
686             Hessian: 'on'
687             HessMult: []
688             HessPattern: []
689             HessUpdate: []
690         InitBarrierParam: []
691         InitTrustRegionRadius: []
692             Jacobian: []
693             JacobMult: []
694             JacobPattern: []
695             LargeScale: 'on'
696             MaxNodes: []
697             MaxPCGIter: []
698             MaxProjCGIter: []
699             MaxSQPIter: []
700             MaxTime: []
701         MeritFunction: []
702             MinAbsMax: []
703         NoStopIfFlatInfeas: []
704         ObjectiveLimit: []
705         PhaseOneTotalScaling: []
706         Preconditioner: []
707         PrecondBandWidth: []
708         RelLineSrchBnd: []
709         RelLineSrchBndDuration: []
710         ScaleProblem: []
711         SubproblemAlgorithm: []
712             TolCon: []
713             TolConSQP: []
714             TolGradCon: []
715             TolPCG: []
716             TolProjCG: []
717             TolProjCGAbs: []
718             TypicalX: []
719         UseParallel: []
720
721
722 Results for n= 10
723 Calculated minimizer:
724
725 x =
726
727     0.3949    0.3812    0.3767    0.3745    0.3731    0.3723    0.3716    0.3712    0.3708
728     0.3705
729
729 Calculated Minimum value: -19.9644
730 Norm of gradient at calculated Minimum: 0.00012999
731 Condition of Hessian at calculated Minimum: 10.523
732     iterations: 12
733     funcCount: 13
734     stepsize: 0.0015
735     cgiterations: 8
736     firstorderopt: 9.4470e-05
737     algorithm: 'trust-region'

```



```

738         message: 'Local minimum possible. fminunc stopped because the final change
in function value relative to its initial value is less than the default value of the
function tolerance. Stopping criteria details: Optimization stopped because
the relative objective function value is changing by less than options.
FunctionTolerance = 1.000000e-06. Optimization Metric
Options relative change objective = 2.20e-07 FunctionTolerance =
1e-06 (default)'
739 constrviolation: []
740
741
742 Results for n= 100
743 Calculated Minimum value: -1857.764
744 Norm of gradient at calculated Minimum: 1.8685e-07
745 Condition of Hessian at calculated Minimum: 100.0717
746 iterations: 11
747 funcCount: 12
748 stepsize: 2.2483e-05
749 cgiterations: 9
750 firstorderopt: 1.8658e-07
751 algorithm: 'trust-region'
752 message: 'Local minimum found. Optimization completed because the size of
the gradient is less than the default value of the optimality tolerance. Stopping
criteria details: Optimization completed: The first-order optimality measure,
1.865793e-07, is less than options.OptimalityTolerance = 1.000000e-06, and no
negative/ zero curvature is detected in the trust-region model. Optimization
Metric Options relative first-order optimality
= 1.87e-07 OptimalityTolerance = 1e-06 (default)'
753 constrviolation: []
754
755
756 Results for n= 1000
757 Calculated Minimum value: -184123.6576
758 Norm of gradient at calculated Minimum: 0.0010339
759 Condition of Hessian at calculated Minimum: 999.9952
760 iterations: 16
761 funcCount: 17
762 stepsize: 0.0087
763 cgiterations: 12
764 firstorderopt: 1.0831e-04
765 algorithm: 'trust-region'
766 message: 'Local minimum possible. fminunc stopped because the final change
in function value relative to its initial value is less than the default value of the
function tolerance. Stopping criteria details: Optimization stopped because
the relative objective function value is changing by less than options.
FunctionTolerance = 1.000000e-06. Optimization Metric
Options relative change objective = 2.83e-08 FunctionTolerance =
1e-06 (default)'
767 constrviolation: []
768
769
770
771 (D)
772
773 Testing objective function f_d with options:
774 Display: 'off'
775 MaxFunEvals: []
776 MaxIter: []
777 TolFun: []
778 TolX: []
779 FunValCheck: []
780 OutputFcn: []
781 PlotFcns: []
782 ActiveConstrTol: []
783 Algorithm: []
784 AlwaysHonorConstraints: []
785 DerivativeCheck: []
786 Diagnostics: []
787 DiffMaxChange: []

```

```

788         DiffMinChange: []
789         FinDiffRelStep: []
790         FinDiffType: []
791         GoalsExactAchieve: []
792         GradConstr: []
793         GradObj: 'on'
794         HessFcn: []
795         Hessian: []
796         HessMult: []
797         HessPattern: []
798         HessUpdate: []
799         InitBarrierParam: []
800         InitTrustRegionRadius: []
801         Jacobian: []
802         JacobMult: []
803         JacobPattern: []
804         LargeScale: 'off'
805         MaxNodes: []
806         MaxPCGIter: []
807         MaxProjCGIter: []
808         MaxSQPIter: []
809         MaxTime: []
810         MeritFunction: []
811         MinAbsMax: []
812         NoStopIfFlatInfeas: []
813         ObjectiveLimit: []
814         PhaseOneTotalScaling: []
815         Preconditioner: []
816         PrecondBandWidth: []
817         RelLineSrchBnd: []
818         RelLineSrchBndDuration: []
819         ScaleProblem: []
820         SubproblemAlgorithm: []
821         TolCon: []
822         TolConSQP: []
823         TolGradCon: []
824         TolPCG: []
825         TolProjCG: []
826         TolProjCGAbs: []
827         TypicalX: []
828         UseParallel: []
829
830
831 Results for n= 10
832 Calculated minimizer:
833
834 x =
835
836     0.7073    1.4114    2.1295    2.8233    3.5305    4.2432    4.9540    5.6603    6.3619
837     7.0693
838 Calculated Minimum value: -16769.7156
839 Norm of gradient at calculated Minimum: 0.029042
840     iterations: 54
841     funcCount: 57
842     stepsize: 0.0481
843     lssteplength: 1
844     firstorderopt: 0.0135
845     algorithm: 'quasi-newton'
846     message: 'Local minimum found. Optimization completed because the size of the
      gradient is less than the default value of the optimality tolerance. Stopping
      criteria details: Optimization completed: The first-order optimality measure,
      9.368012e-07, is less than options.OptimalityTolerance = 1.000000e-06.
      Optimization Metric Options relative
      norm(gradient) = 9.37e-07 OptimalityTolerance = 1e-06 (default)'
847
848
849 Results for n= 100

```

```

850 Calculated Minimum value: -10987678753.1027
851 Norm of gradient at calculated Minimum: 13.4558
852     iterations: 78
853     funcCount: 83
854     stepsize: 0.2896
855     lssteplength: 1
856     firstorderopt: 3.7624
857     algorithm: 'quasi-newton'
858     message: 'Local minimum found. Optimization completed because the size of the
      gradient is less than the default value of the optimality tolerance. Stopping
      criteria details: Optimization completed: The first-order optimality measure,
      6.246662e-07, is less than options.OptimalityTolerance = 1.000000e-06.
      Optimization Metric Options relative
      norm(gradient) = 6.25e-07 OptimalityTolerance = 1e-06 (default)'
```

```

859
860
861 Results for n= 1000
862 Calculated Minimum value: -1.051366255854342e+16
863 Norm of gradient at calculated Minimum: 25147.6496
864     iterations: 152
865     funcCount: 161
866     stepsize: 3.3739
867     lssteplength: 1
868     firstorderopt: 3.6475e+03
869     algorithm: 'quasi-newton'
870     message: 'Local minimum found. Optimization completed because the size of the
      gradient is less than the default value of the optimality tolerance. Stopping
      criteria details: Optimization completed: The first-order optimality measure,
      5.936574e-07, is less than options.OptimalityTolerance = 1.000000e-06.
      Optimization Metric Options relative
      norm(gradient) = 5.94e-07 OptimalityTolerance = 1e-06 (default)'
```

```

871
872
873 Testing objective function f_d with options:
874     Display: 'off'
875     MaxFunEvals: []
876     MaxIter: []
877     TolFun: []
878     TolX: []
879     FunValCheck: []
880     OutputFcn: []
881     PlotFcns: []
882     ActiveConstrTol: []
883     Algorithm: []
884     AlwaysHonorConstraints: []
885     DerivativeCheck: []
886     Diagnostics: []
887     DiffMaxChange: []
888     DiffMinChange: []
889     FinDiffRelStep: []
890     FinDiffType: []
891     GoalsExactAchieve: []
892     GradConstr: []
893     GradObj: 'on'
894     HessFcn: []
895     Hessian: 'off'
896     HessMult: []
897     HessPattern: []
898     HessUpdate: []
899     InitBarrierParam: []
900     InitTrustRegionRadius: []
901     Jacobian: []
902     JacobMult: []
903     JacobPattern: []
904     LargeScale: 'on'
905     MaxNodes: []
906     MaxPCGIter: []
907     MaxProjCGIter: []

```

```

908         MaxSQPIter: []
909         MaxTime: []
910         MeritFunction: []
911         MinAbsMax: []
912         NoStopIfFlatInfeas: []
913         ObjectiveLimit: []
914         PhaseOneTotalScaling: []
915         Preconditioner: []
916         PrecondBandWidth: []
917         RelLineSrchBnd: []
918         RelLineSrchBndDuration: []
919         ScaleProblem: []
920         SubproblemAlgorithm: []
921         TolCon: []
922         TolConSQP: []
923         TolGradCon: []
924         TolPCG: []
925         TolProjCG: []
926         TolProjCGAbs: []
927         TypicalX: []
928         UseParallel: []
929
930
931 Results for n= 10
932 Calculated minimizer:
933
934 x =
935
936     0.7073    1.4114    2.1295    2.8233    3.5305    4.2432    4.9540    5.6603    6.3619
937     7.0693
938
939 Calculated Minimum value: -16769.7156
940 Norm of gradient at calculated Minimum: 0.029042
941     iterations: 54
942     funcCount: 57
943     stepsize: 0.0481
944     lssteplength: 1
945     firstorderopt: 0.0135
946     algorithm: 'quasi-newton'
947     message: 'Local minimum found. Optimization completed because the size of the
948             gradient is less than the default value of the optimality tolerance. Stopping
949             criteria details: Optimization completed: The first-order optimality measure,
950             9.368012e-07, is less than options.OptimalityTolerance = 1.000000e-06.
951             Optimization Metric Options relative
952             norm(gradient) = 9.37e-07 OptimalityTolerance = 1e-06 (default)'
953
954
955 Results for n= 100
956 Calculated Minimum value: -10987678753.1027
957 Norm of gradient at calculated Minimum: 13.4558
958     iterations: 78
959     funcCount: 83
960     stepsize: 0.2896
961     lssteplength: 1
962     firstorderopt: 3.7624
963     algorithm: 'quasi-newton'
964     message: 'Local minimum found. Optimization completed because the size of the
965             gradient is less than the default value of the optimality tolerance. Stopping
966             criteria details: Optimization completed: The first-order optimality measure,
967             6.246662e-07, is less than options.OptimalityTolerance = 1.000000e-06.
968             Optimization Metric Options relative
969             norm(gradient) = 6.25e-07 OptimalityTolerance = 1e-06 (default)'
970
971
972 Results for n= 1000
973 Calculated Minimum value: -1.051366255854342e+16
974 Norm of gradient at calculated Minimum: 25147.6496
975     iterations: 152

```

```

965         funcCount: 161
966         stepsize: 3.3739
967         lssteplength: 1
968         firstorderopt: 3.6475e+03
969         algorithm: 'quasi-newton'
970         message: 'Local minimum found. Optimization completed because the size of the
          gradient is less than the default value of the optimality tolerance. Stopping
          criteria details: Optimization completed: The first-order optimality measure,
          5.936574e-07, is less than options.OptimalityTolerance = 1.000000e-06.
          Optimization Metric Options relative
          norm(gradient) = 5.94e-07 OptimalityTolerance = 1e-06 (default)'
```

```

971
972
973 Testing objective function f_d with options:
974         Display: 'off'
975         MaxFunEvals: []
976         MaxIter: []
977         TolFun: []
978         TolX: []
979         FunValCheck: []
980         OutputFcn: []
981         PlotFcns: []
982         ActiveConstrTol: []
983         Algorithm: 'trust-region'
984         AlwaysHonorConstraints: []
985         DerivativeCheck: []
986         Diagnostics: []
987         DiffMaxChange: []
988         DiffMinChange: []
989         FinDiffRelStep: []
990         FinDiffType: []
991         GoalsExactAchieve: []
992         GradConstr: []
993         GradObj: 'on'
994         HessFcn: []
995         Hessian: 'on'
996         HessMult: []
997         HessPattern: []
998         HessUpdate: []
999         InitBarrierParam: []
1000         InitTrustRegionRadius: []
1001         Jacobian: []
1002         JacobMult: []
1003         JacobPattern: []
1004         LargeScale: 'on'
1005         MaxNodes: []
1006         MaxPCGIter: []
1007         MaxProjCGIter: []
1008         MaxSQPIter: []
1009         MaxTime: []
1010         MeritFunction: []
1011         MinAbsMax: []
1012         NoStopIfFlatInfeas: []
1013         ObjectiveLimit: []
1014         PhaseOneTotalScaling: []
1015         Preconditioner: []
1016         PrecondBandWidth: []
1017         RelLineSrchBnd: []
1018         RelLineSrchBndDuration: []
1019         ScaleProblem: []
1020         SubproblemAlgorithm: []
1021         TolCon: []
1022         TolConSQP: []
1023         TolGradCon: []
1024         TolPCG: []
1025         TolProjCG: []
1026         TolProjCGAbs: []
1027         TypicalX: []
```

```

1028         UseParallel: []
1029
1030
1031 Results for n= 10
1032 Calculated minimizer:
1033
1034 x =
1035
1036     0.7976    0.6364    2.8053    3.8166    4.0723    3.9870    3.7920    3.6033    3.4749
1037     11.9926
1038 Calculated Minimum value: -16758.8021
1039 Norm of gradient at calculated Minimum: 11.9204
1040 Condition of Hessian at calculated Minimum: 1254.7567
1041     iterations: 401
1042     funcCount: 402
1043     stepsize: 0.0255
1044     cgitations: 1936
1045     firstorderopt: 6.9473
1046     algorithm: 'trust-region'
1047     message: 'Solver stopped prematurely. fminunc stopped because it exceeded
1048 the iteration limit, options .MaxIterations = 400 (the default value).'
1049     constrviolation: []
1050
1051 Results for n= 100
1052 Calculated Minimum value: -10968504409.5875
1053 Norm of gradient at calculated Minimum: 86246.1883
1054 Condition of Hessian at calculated Minimum: 15458.5502
1055     iterations: 401
1056     funcCount: 402
1057     stepsize: 0.1551
1058     cgitations: 8032
1059     firstorderopt: 4.6548e+04
1060     algorithm: 'trust-region'
1061     message: 'Solver stopped prematurely. fminunc stopped because it exceeded
1062 the iteration limit, options .MaxIterations = 400 (the default value).'
1063     constrviolation: []
1064
1065 Results for n= 1000
1066 Calculated Minimum value: -1.051215141638858e+16
1067 Norm of gradient at calculated Minimum: 3110412089.4636
1068 Condition of Hessian at calculated Minimum: 203328.5383
1069     iterations: 253
1070     funcCount: 254
1071     stepsize: 4.9497
1072     cgitations: 818
1073     firstorderopt: 4.9217e+08
1074     algorithm: 'trust-region'
1075     message: 'Local minimum possible. fminunc stopped because the final change
in function value relative to its initial value is less than the default value of the
function tolerance. Stopping criteria details: Optimization stopped because
the relative objective function value is changing by less than options.
FunctionTolerance = 1.000000e-06. Optimization Metric
Options relative change objective = 9.91e-07 FunctionTolerance =
1e-06 (default)'
1076     constrviolation: []
1077
1078

```

src/Test1.txt

2.4 Interpretation

We will proceed to extract the relevant information of the above output for every function tested. For the subsequent sections we will use the following notation

1. O1 : LargeScale = off, GradObj = on
2. O2 : LargeScale = on, GradObj = on, Hessian = off
3. O3 : LargeScale = on, GradObj = on, Hessian = on

Also notice that the default "quasi-newton" algorithm used in `fminunc` does not make use of a provided Hessian, therefore the algorithm has been changed to "trust-region" in O3 when an analytic Hessian is supplied. Additionally, where an analytic Hessian is supplied (i.e. O3) the condition of the Hessian at the approximated minimizer has been calculated.

2.4.1 f_a

Option	fminunc output description	Minimum value	Norm of gradient	iterations	cond(H)
O1	Local Min found	$1.1571 \cdot 10^{-13}$	$5.8763 \cdot 10^{-7}$	8	/
O2	Local Min found	$1.157 \cdot 10^{-13}$	$5.8763 \cdot 10^{-7}$	8	/
O3	Local Min possible	$2.0406 \cdot 10^{-7}$	0.00079236	12	10

All options yield a very small minimum value after few iterations, however comparing the order of magnitude of the different outputs (i.e. calculated minimal values and norms of gradients at calculated minimum) one notices, that O1 and O2 seem to be about twice as accurate (i.e. square of the error in O3) as O3 while also needing fewer iterations. An educated guess would attribute this rather strange disparity to the use of different algorithms in O1,O2 ("quasi-newton") and O3("trust-region").

2.4.2 f_b

Option	fminunc output description	Minimum value	Norm of gradient	iterations	cond(H)
O1	Local Min found	$6.6115e - 11$	0.00029178	39	/
O2	Local Min found	$6.6115e - 11$	0.00029178	39	/
O3	Local Min possible	$1.931e - 17$	$3.3208e - 08$	30	2508.0095

Again all options yield a minimum value very close to zero. This time, however, O3 performs better, in that it calculates a minimizer whose value is smaller by a factor of 10^{-6} than that produced by O1, O2 where additionally the norm of the gradient is significantly smaller (by a factor of about 10^{-4}) than the norm of the gradient of the outputs of O1 and O3. This is not very surprising, given the fact that in O3 an analytic Hessian is supplied, allowing for faster convergence. Interestingly enough, `fminunc` outputs "local Min possible" with O3, probably due to positive semi-definiteness of the Hessian in the approximated minimizer.

2.4.3 f_c

1. $n = 10$

Option	fminunc output description	Minimum value	Norm of gradient	iterations	cond(H)
O1	Local Min found	-19.9644	$1.0378e - 05$	29	/
O2	Local Min found	-19.9644	$1.0378e - 05$	29	/
O3	Local Min possible	-19.9644	0.00012999	12	10.523

2. $n = 100$

Option	fminunc output description	Minimum value	Norm of gradient	iterations	cond(H)
O1	Local Min found	-1857.764	0.00082223	79	/
O2	Local Min found	-1857.764	0.00082223	29	/
O3	Local Min found	-1857.764	$1.8685e - 07$	11	100.0717

3. $n = 1000$

Option	fminunc output description	Minimum value	Norm of gradient	iterations	cond(H)
O1	Local Min found	-184123.6576	0.010364	203	/
O2	Local Min found	-184123.6576	0.010364	203	/
O3	Local Min possible	-184123.6576	0.0010339	16	999.9952

For every input (i.e. $n = 10, 100, 1000$) **fminunc** produced the same result independent of the options used. Differences can be observed in the norms of the gradients at the approximated minimizers and the amount of iterations needed. Here, the most striking contrast can be found for $n = 1000$: While O1 and O2 require 2 about 200 iterations, O3 terminates after 16 iterations while also producing an approximated minimizers where the norm of the gradient is smaller than the output of O1,O2 by a factor of 10. Concerning iterations, O3 seems to perform better allround. For $n = 100$ we also obtain an approximate minimizer where the norm of the gradient is smaller by a factor of 10^3 compared to what O1 and O2 outputs. The fact that for $n = 10, 1000$ O3 outputs "local minimum possible" may again be attributed to positive semidefiniteness of the Hessian at the approximated minimizer.

2.4.4 f.d

1. $n = 10$

Option	fminunc output description	Minimum value	Norm of gradient	iterations	cond(H)
O1	Local Min found	-16769.7156	0.029042	54	/
O2	Local Min found	-16769.7156	0.029042	54	/
O3	Solver stopped prematurely; exceeded it limit	-16758.8021	11.9204	401	1254.7567

2. $n = 100$

Option	fminunc output description	Minimum value	Norm of gradient	iterations	cond(H)
O1	Local Min found	-10987678753.1027	13.4558	78	/
O2	Local Min found	-10987678753.1027	13.4558	78	/
O3	Solver stopped prematurely; exceeded it limit	-10968504409.5875	86246.1883	401	15458.5502

3. $n = 1000$

Option	fminunc output description	Minimum value	Norm of gradient	iterations	cond(H)
O1	Local Min found	$-1.051366255854342e + 16$	25147.6496	152	/
O2	Local Min found	$-1.051366255854342e + 16$	25147.6496	152	/
O3	Local Min possible	$-1.051215141638858e + 16$	3110412089.4636	253	203328.5383

What makes the test of function **f.d** above stand out in comparison to the tests beforehand is that **fminunc** terminates prematurely after having calculated 400 iterations without meeting the default convergence criteria. Observe, that the approximated minimum values of O1 and O2 seem to coincide for all inputs, as does the norm of the gradient and the number of required iterations. We remark, that for $n = 100$ and especially for $n = 1000$, the norm of the gradient at the calculated minimizer is no longer close to zero, meaning we are no longer "close" to a stationary point, yet **fminunc**'s output description states "local minimum found". What is even more interesting is the fact, that O3 stops prematurely, needing more than the default maximal number of iterations, when O1 and O2 (without using the Hessian) terminate after a reasonable amount of iterations. This effect might be caused by making use of a badly conditioned Hessian, slowing down convergence. The calculated minimum values of O3 seem to be coinciding to the outputs of O1 and O2 up to some significant digit, increasing the default maxiter bound would allow for more accurate approximation. Striking is the astronomical norm of the gradient in O3 with $n = 1000$, yet **fminunc** describes the output to be a possible local minimum.

3 Exercise 3

3.1 Source Code


```

1 fprintf('===== (a) =====\n');
2 %minimum should be at (0,0), f(min) should be 0
3 x0 = [10, -3];
4 tic
5 [x, fval, exitflag, output] = fminsearch(@f_aH, x0);
6 toc
7 if exitflag == 0
8     fprintf('Iterations or function evaluations exceeded options.')
9     return
10 end
11 fprintf('\nSolution found at: [ ');
12 fprintf(' %g ',x);
13 fprintf(' ]\nFunction value: %.8f\n\n', fval);
14 output
15
16 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
17
18 fprintf('===== (b) =====\n');
19 %minimum should be at (1,1), f(min) should be 0
20 x0 = [-1, 2];
21 tic
22 [x, fval, exitflag, output] = fminsearch(@f_bH, x0);
23 toc
24 if exitflag == 0
25     fprintf('Iterations or function evaluations exceeded options.')
26     return
27 end
28 fprintf('\nSolution found at: [ ');
29 fprintf(' %g ',x);
30 fprintf(' ]\nFunction value: %.8f\n\n', fval);
31 output
32
33 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
34
35 options = optimset('MaxFunEvals', 1000000, 'MaxIter', 1000000);
36 fprintf('===== (c) =====\n');
37 for i = 1:3
38     fprintf('SIZE: n = 10^%i\n',i);
39     x0 = 5*ones(10^i, 1);
40     tic
41     [x, fval, exitflag, output] = fminsearch(@f_cH, x0, options);
42     toc
43     if exitflag == 0
44         fprintf('Iterations or function evaluations exceeded options.')
45         return
46     end
47     if i < 3
48         fprintf('\nSolution found at: [\n');
49         fprintf(' %g\n',x);
50         fprintf(' ]');
51     end
52     fprintf('\nFunction value: %.8f\n', fval);
53     output
54 end
55
56 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
57
58 options = optimset('MaxFunEvals', 1000000, 'MaxIter', 1000000);
59 fprintf('===== (d) =====\n');
60 for i = 1:3
61     fprintf('SIZE: n = 10^%i\n',i);
62     x0 = [zeros(1,10^i-1),10^(i+1)];
63     tic
64     [x, fval, exitflag, output] = fminsearch(@f_dH, x0, options);
65     toc
66     if exitflag == 0
67         fprintf('Iterations or function evaluations exceeded options.')

```

```

68     return
69 end
70 if i == 1
71     fprintf('\nSolution found at: [\n');
72     fprintf('%g\n',x);
73     fprintf(']');
74 end
75 fprintf('\nFunction value: %.8f\n', fval);
76 output
77 end

```

src/Pract1Ex3.m

3.2 Output

```

1 ===== (a) =====
2 Elapsed time is 0.012591 seconds.
3
4 Solution found at: [ -3.79534e-05  1.22752e-05 ]
5 Function value: 0.00000000
6
7
8 output =
9
10 struct with fields:
11
12     iterations: 52
13     funcCount: 97
14     algorithm: 'Nelder-Mead simplex direct search'
15     message: 'Optimization terminated: the current x satisfies the termination
16             criteria using OPTIONS.TolX of 1.000000e-04 and F(X) satisfies the convergence
17             criteria using OPTIONS.TolFun of 1.000000e-04 '
18
19 ===== (b) =====
20 Elapsed time is 0.010781 seconds.
21
22 Solution found at: [ 0.999991  0.999983 ]
23 Function value: 0.00000000
24
25
26 output =
27
28 struct with fields:
29
30     iterations: 105
31     funcCount: 195
32     algorithm: 'Nelder-Mead simplex direct search'
33     message: 'Optimization terminated: the current x satisfies the termination
34             criteria using OPTIONS.TolX of 1.000000e-04 and F(X) satisfies the convergence
35             criteria using OPTIONS.TolFun of 1.000000e-04 '
36
37 ===== (c) =====
38 SIZE: n = 10^1
39
40 Exiting: Maximum number of function evaluations has been exceeded
41 - increase MaxFunEvals option.
42 Current function value: 53.708279
43
44 Elapsed time is 0.183353 seconds.
45 Iterations or function evaluations exceeded options.
46
47 ==> increase MaxFunEvals (default: 200*#variables) to 100000
48 Now:
49
50 Exiting: Maximum number of iterations has been exceeded
51 - increase MaxIter option.

```

```

48         Current function value: 53.625224
49
50 ==> increase MaxIter (same default) to 100000
51 Now:
52
53 SIZE: n = 10^1
54 Elapsed time is 0.140587 seconds.
55
56 Solution found at: [
57 0.394952
58 0.381119
59 0.376755
60 0.374441
61 0.373128
62 0.372251
63 0.37164
64 0.371131
65 0.370765
66 0.370491
67 ]
68 Function value: -19.96439719
69
70 output =
71
72     struct with fields:
73
74         iterations: 3883
75         funcCount: 5422
76         algorithm: 'Nelder-Mead simplex direct search'
77         message: 'Optimization terminated: the current x satisfies the termination
78                 criteria using OPTIONS.TolX of 1.000000e-04 and F(X) satisfies the convergence
79                 criteria using OPTIONS.TolFun of 1.000000e-04'
80
81 SIZE: n = 10^2
82
83 Exiting: Maximum number of function evaluations has been exceeded
84         - increase MaxFunEvals option.
85         Current function value: 28071.812796
86
87 Elapsed time is 279.552686 seconds.
88 Iterations or function evaluations exceeded options.
89
90 ==> increase both Max-values to 1000000
91 Now:
92
93 SIZE: n = 10^2
94 Elapsed time is 783.989220 seconds.
95
96 Function value: 27997.73143250
97
98 output =
99
100     struct with fields:
101
102         iterations: 358765
103         funcCount: 385746
104         algorithm: 'Nelder-Mead simplex direct search'
105         message: 'Optimization terminated: the current x satisfies the termination
106                 criteria using OPTIONS.TolX of 1.000000e-04 and F(X) satisfies the convergence
107                 criteria using OPTIONS.TolFun of 1.000000e-04'
108
109 SIZE: n = 10^3
110 Operation terminated by user during fminsearch (line 322)
111
112 ===== (d) =====
113
114 SIZE: n = 10^1
115 Elapsed time is 0.012700 seconds.

```

```

112 Solution found at: [
113 0.482444
114 0.863983
115 1.8512
116 11.5311
117 7.55222
118 -7.94144
119 -0.775348
120 -0.633636
121 -5.43603
122 31.9502
123 ]
124 Function value: -16468.38764339
125
126 output =
127
128     struct with fields:
129
130         iterations: 1098
131         funcCount: 1642
132         algorithm: 'Nelder-Mead simplex direct search'
133         message: 'Optimization terminated: the current x satisfies the termination
134                  criteria using OPTIONS.TolX of 1.000000e-04 and F(X) satisfies the convergence
135                  criteria using OPTIONS.TolFun of 1.000000e-04'
136
137 SIZE: n = 10^2
138 Elapsed time is 7.013846 seconds.
139
140 Function value: -10409439951.64850044
141
142 output =
143
144     struct with fields:
145
146         iterations: 142685
147         funcCount: 152272
148         algorithm: 'Nelder-Mead simplex direct search'
149         message: 'Optimization terminated: the current x satisfies the termination
150                  criteria using OPTIONS.TolX of 1.000000e-04 and F(X) satisfies the convergence
151                  criteria using OPTIONS.TolFun of 1.000000e-04'
152
153 SIZE: n = 10^3
154
155 Exiting: Maximum number of function evaluations has been exceeded
156 - increase MaxFunEvals option.
157     Current function value: -8563658281094496.000000
158
159 Elapsed time is 3127.988139 seconds.
160 Iterations or function evaluations exceeded options.

```

src/Pract1Ex3 output.txt

3.3 Interpretation

Function	n	fminsearch output description	Function value
f_a	-	termination criteria satisfied	0
f_b	-	termination criteria satisfied	0
f_c	10	termination criteria satisfied	53.625224
f_c	100	termination criteria satisfied	783.989220
f_c	1000	terminated because it took too long	-
f_d	10	termination criteria satisfied	16468.38764339
f_d	100	termination criteria satisfied	10409439951.64850044
f_d	1000	Maximum funevals exceeded	-

Disclaimer: fminsearch doesn't use the Hessian NOR the gradient, so for better efficiency we added a

"if nargout > 1" clause so that the gradient wouldn't be evaluated every time.

Both f_a and f_b converge to the desired solution very fast, though they need significantly more iterations than fminunc.

Initially, f_c with $n=10$ did not terminate with a solution since it immediately put out the message that the maximum number of function evaluations had been exceeded - "increase MaxFunEvals option". After setting this option from the default value (200 times the number of variables = 2000) to 100000, the same message was shown for the maximum number of iterations, so MaxIter was also increased. Now we got a solution, and even the same one as with fminunc.

With $n=100$, we had to increase both MaxFunEvals and MaxIter to 1.000.000 to get a result (it needed around 350.000 iterations and 380.000 evaluations). However, the function value was around 28000 in contrast to the -1850 from fminunc, so that can't be correct. Since for $n=100$, it already didn't even calculate the correct result, it took significantly longer than for $n=10$, we aborted the calculation for $n=1000$ after an hour.

In comparison to fminunc, f_d gave us different solution vectors for x , but the function values came really close to the ones from fminunc, at least until $n=100$. For $n=1000$, Matlab reached the maximum number of iterations, 1.000.000, and returned the current value, which was almost $-1.e+16$, the returned function value from fminunc.

4 Exercise 4

4.1 Source Code

```
1 options1 = optimoptions('fminunc','GradObj','off');
2 options2 = optimset('GradObj','off');
3
4 f = @(x) x(1)+10*max(x(1)^2+2*x(2)^2-1,0);
5 x0=[1,1];
6 s1 = fminunc(f,x0,options1)
7 s2 = fminsearch(f,x0,options2)
8
9 e1 = norm(s1-[-1,0])
10 e2 = norm(s2-[-1,0])
```

src/Prog1Ex4.m

```
1 >> Prog1Ex4
2
3 Local minimum possible.
4
5 fminunc stopped because it cannot decrease the objective function
6 along the current search direction.
7
8 <stopping criteria details>
9
10 s1 =
11     -0.9997    -0.0168
12
13 s2 =
14
15     -1.0000     0.0000
16
17 e1 =
18
19     0.0168
20
21 e2 =
22
23     4.3007e-05
```

4.2 Solution of fminunc

When applying `fminunc` to the problem considering the given non-continuously differentiable function

$$\min_{x \in \mathbb{R}^2} x_1 + 10 \max\{x_1^2 + 2x_2^2 - 1, 0\}$$

we get that $x = (x_1, x_2) \approx (-0.9997, -0.0168)$ solves the above equation where the exact solution should correspond to the vector $\bar{x} = (x_1, x_2) = (-1, 0)$. This implies a numerical error of $\|x - \bar{x}\| \approx 0.0168$ as can be obtained by the `MATLAB` source code in section 4.1.

4.3 Solution of fminsearch

If instead of `fminunc` the `MATLAB` command `fminsearch` is applied to the same problem as in section 4.2 we get the result of $x = (x_1, x_2) \approx (-1.0000, 0.0000)$. Consequently, we also get a numerical error much smaller in size and given by $\|x - \bar{x}\| \approx 4.3007 \cdot 10^{-5}$.

4.4 Interpretation

In the `MATLAB` output it states that `fminunc` stopped because it cannot decrease the objective function along the current search direction any further. Though, if stopping criteria details are displayed we get the following additional information.

```
1 Optimization stopped because the objective function cannot be decreased in the
2 current search direction. Either the predicted change in the objective function,
3 or the line search interval is less than eps.
```

Consequently, it must be the case that the line search algorithm in `fminunc` which makes use of numerical respectively analytic gradients is not appropriate for the considered problem. On the opposite, the simplex search method of `fminsearch` does not estimate according gradients using finite differences and therefore provides an adequate line search interval and ultimately a more reliable result.