MATH693a
Dr. Peter Blomgren
HW03
Trust Region Methods

Matteo Polimeno

November 2^{nd}, 2018

Introduction

In this assignment we were given the objective function:

$$f(x) = 10(x_2 - x_1^2)^2 + (1 - x_1)^2 \tag{1}$$

and asked to draw the contour lines of the quadratic model for it, assuming that for the Hessian of $f(x)$ would hold:

$$B_k = \nabla^2 f(\mathbf{x_k}) \tag{2}$$

and the family of solutions for the Trust Region subproblem.

Part 1

In this section we will use $x_0 = [0, -1]$ as our starting point, thus as the center of the trust region. We display the results of the iterations for some values of the trust region radius D_k, between 0 and 2. We performed Cholesky Factorization on the Hessian to find the values of λ (the diagonal elements of the Hessian) to use in order to shift the 2×2 matrix away from regions of non-definiteness (i.e. with any diagonal elements less than 0).
The table below shows the results in terms of number of iterations and the elapsed time necessary to converge to the optimal value for λ, given a maximum number of iterations fixed at 100 thousands.

D_k	Iterations	Elapsed Time (sec)	Optimal λ
0.5	13778	0.248767	20.0833
1	7201	0.146302	0.0227
1.5	100000	1.659368	19.7345
2	100000	1.721085	1.0984

We can immediately appreciate that for a trust-region radius $D_k \leq 1$, we have a relatively fast convergence to an optimal value of λ. Whereas, for $D_k > 1$, we do not have convergence for the point $x_0 = [0, -1]$. Figure 1 below shows the contour of the model for such point, with the trust region and the step direction.

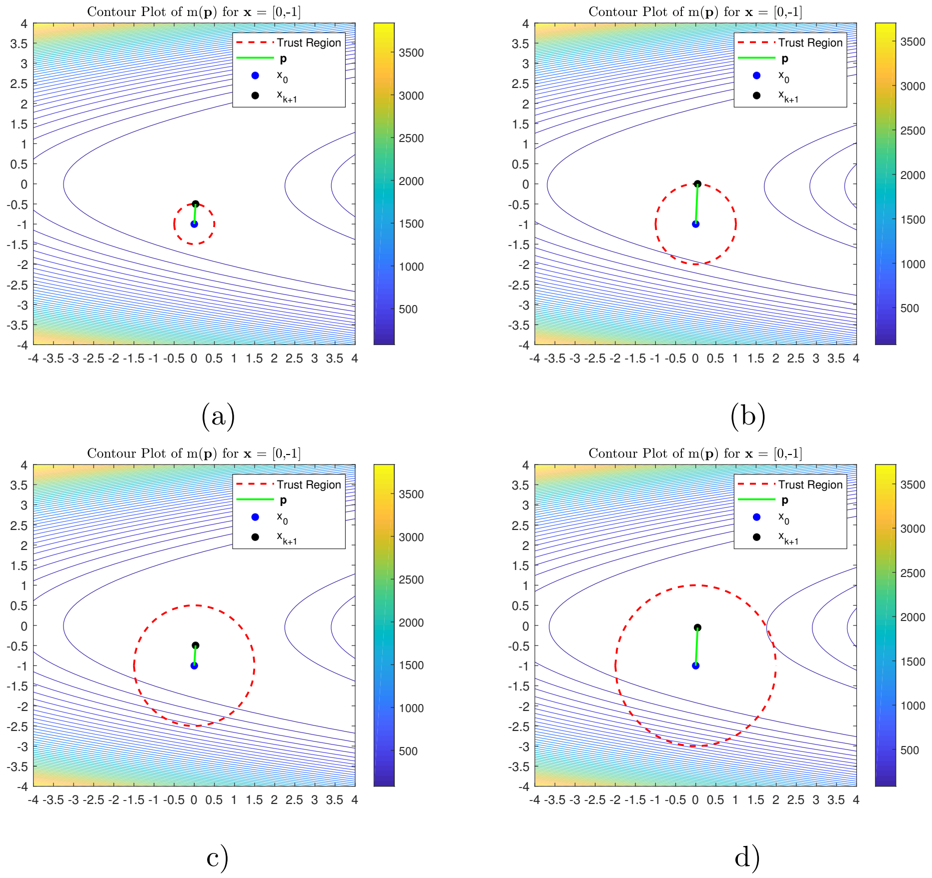


Figure 1: Trust Regions for different values of the radius D_k for the starting point $x_0 = [0, -1]$: a) $D_k = 0.5$, b) $D_k = 1$, c) $D_k = 1.5$, d) $D_k = 2$

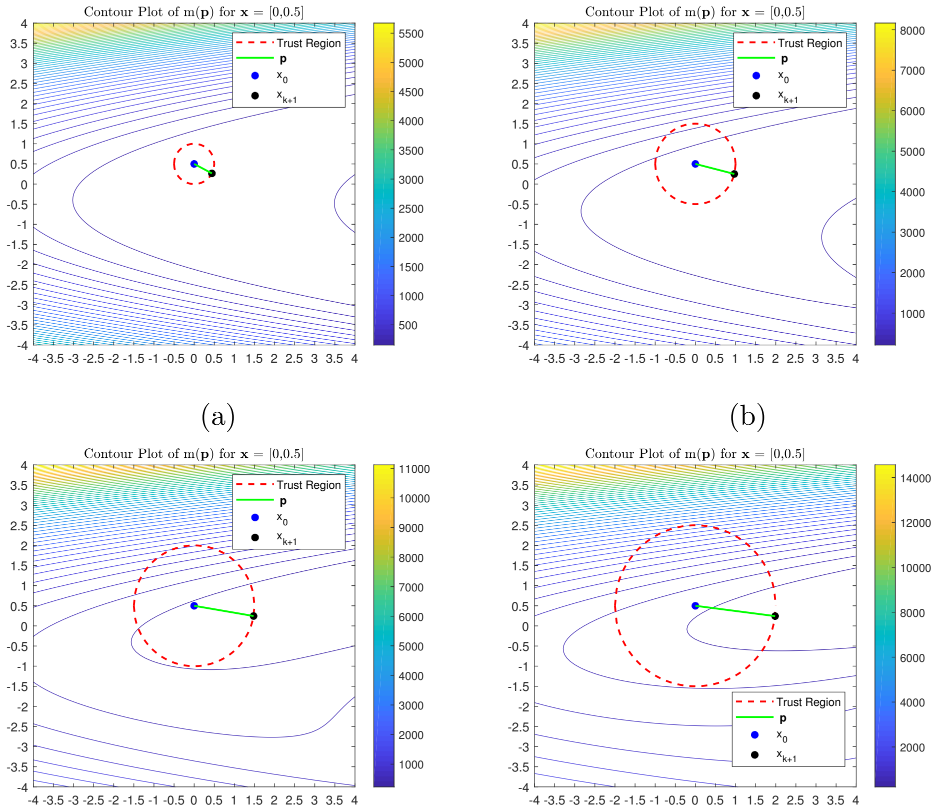
We see that in Fig1a) and b), our lambda gives a step direction from the center of the trust region to its boundary, whereas in Fig1c) and d) we see that we stay well inside the region and the step length is much shorter than the trust region radius.

Part 2

In this section we will use $x_1 = [0, 0.5]$ as our starting point, thus as the center of the trust region. We display the results of the iterations for some values of the trust region radius D_k, between 0 and 2. We performed Cholesky Factorization on the Hessian to find the values of λ (the diagonal elements of the Hessian) to use in order to shift the 2×2 matrix away from regions of non-definiteness (i.e. with any diagonal elements less than 0).
The table below shows the results in terms of number of iterations and the elapsed time necessary to converge to the optimal value for λ, given a maximum number of iterations fixed at 100 thousands.

D_k	Iterations	Elapsed Time (sec)	Optimal λ
0.5	89318	1.440988	22.5324
1	23888	0.409682	20.0654
1.5	11193	0.230730	19.3529
2	6537	0.138706	19.0083

We can immediately appreciate that, in this case, for the four trust-region radii $D_k s$ displayed, we have a relatively fast convergence to an optimal value of λ, with running time, number of iterations and λ value that decrease as D_k increases.
Below we plot the contour of the model $m(\mathbf{p})$ for the given point, as well as the trust region and the step direction.



(a) (b)

c) d)

Figure 2: Trust Regions for different various of the radius D_k for the starting point $x_1 = [0, 0.5]$: a) $D_k = 0.5$, b) $D_k = 1$, c) $D_k = 1.5$, d) $D_k = 2$

Appendix - Matlab Code

```matlab
%trust region
clear all
clc
nn = 100000;
mu = (min(diag(zeros(2))));
x = [0,-1];
Dk = 2;
tol = 1e-8;
xrange = linspace(-Dk,Dk,101);
tic
t = false;
for ii = 1:nn
    nabla = grad_f(x);
    if ii==1
        A = hess_f(x);
        mu(ii) = min(abs(diag(A)));
        if A(ii,ii) < 0
            mu(ii) = min(abs(diag(A))) + 0.1;
        end
    end
    A = hess_f(x) + mu(ii)*eye(length(hess_f(x)));
    if ii==2 && (A(ii,ii) < 0)
        mu(ii) = min(abs(diag(A))) + 0.1;
    end
    if mu(ii) < 0
        mu(ii) = min(abs(diag(A))) + 0.1;
    end
    L = cholesky(x,A);
    LT = (L)';
    LI = (L)^-1;
    q = ((LT)^-1)*(-nabla)';
    p = (LI)*q;
    mu(ii+1) = mu(ii) + (((norm(p))/(norm(q)))^2 )*((
        norm(p))-Dk)/Dk;
    if abs(mu(ii+1)-mu(ii)) < tol
        t = true;
        break
    end
    mu(ii) = mu(ii+1);
    xkp1 = x + p';
end
```

```
toc
mu(ii)
xcontour = linspace(-4,4,101);
[X,Y] = meshgrid(xcontour,xcontour);
pt = p';
m = zeros(length(xcontour),length(xcontour));

for jj=1:length(xcontour)
     x2 = xcontour;
     for kk=1:length(xcontour)
          f = h([x2(jj),x2(kk)]);
          g = grad_f([x2(jj),x2(kk)]);
          Bk = hess_f([x2(jj),x2(kk)])+mu(kk)*eye(length
              (hess_f([x2(jj),x2(kk)])));
          m(jj,kk) = f + pt*(g')+.5*pt*(Bk)*p;
     end
end

y1 = sqrt(Dk^2-(xrange-x(1)).^2)+x(2);
y2 = -sqrt(Dk^2-(xrange-x(1)).^2)+x(2);
figure(1)
mp = contour(X,Y,m,50);
hold on
circ1 = plot(xrange,y1,'r--','linewidth',1.5);
hold on
circ2 = plot(xrange,y2,'r--','linewidth',1.5);
hold on
x0 = plot(x(1),x(2),'b*','linewidth',2);
hold on
x1 = plot(xkp1(1),xkp1(2),'k*','linewidth',2);
hold on
p = plot([x(1) xkp1(1)],[x(2) xkp1(2)],'g-','linewidth
    ',1.5);
hold off
set(gca, 'Xlim',[-4,4])
set(gca,'Xtick',(-4:0.5:4))
set(gca, 'Ylim',[-4,4])
set(gca,'Ytick',(-4:0.5:4))
title('Contour Plot of m({\bf p}) for {\bf x} = [0,-1]
    ','interpreter','latex')
legend([circ1,p, x0, x1],{'Trust Region','{\bf p}','x_
    {0}','x_{k+1}'})
colorbar
axis equal
```

```

%cholesky
function L = cholesky(~,A)
n = length(A);
L = zeros(n,n);
for i = 1:n
    L(i,i) = sqrt(A(i,i));
    if A(i,i) < 0
        break
    end
    for j = i+1:n
        L(j,i) = A(j,i)/(L(i,i));
        for k = i+1:j
            A(j,k) = A(j,k) - L(j,i)*L(k,i);
        end
    end
end
end

function f = h(x)
f = 10*(x(2)-x(1)^2)^2+(1-x(1))^2;
end

function nabla = grad_f(x)
nabla = [40*x(1)*(x(1)^2-x(2))+2*(x(1)-1), 20*(x(2)-x
    (1)^2)];
end

function A = hess_f(x)
A = [120*x(1)^2-40*x(2)+2, -40*x(1);...
        -40*x(1), 20];
end
```