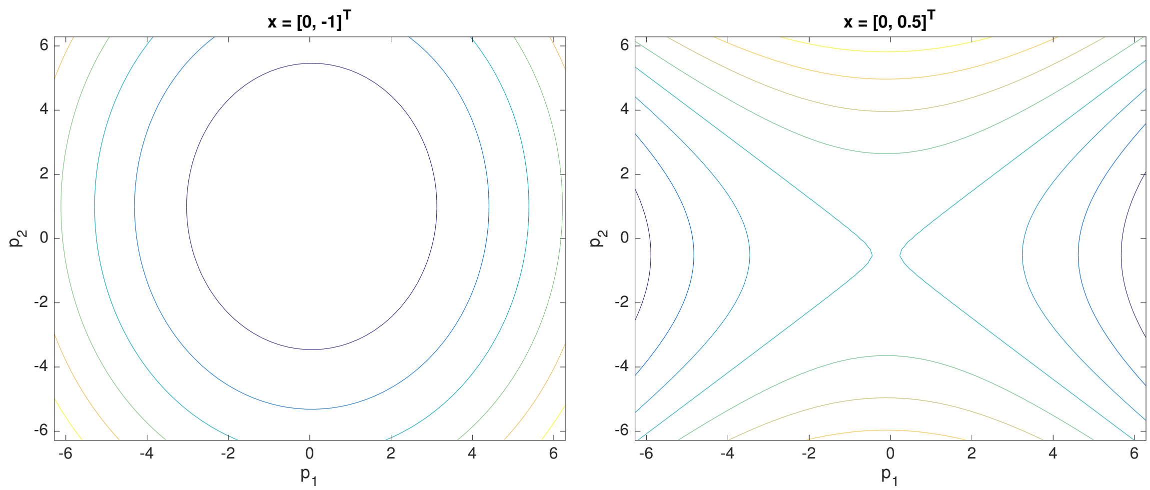
CSC 2305 Numerical Methods for Optimization Problems (Winter 2016)
Assignment # 4

1002119049 Hao Wang (UTORid: `wangh110`)

March 14, 2016

1 Problem 1

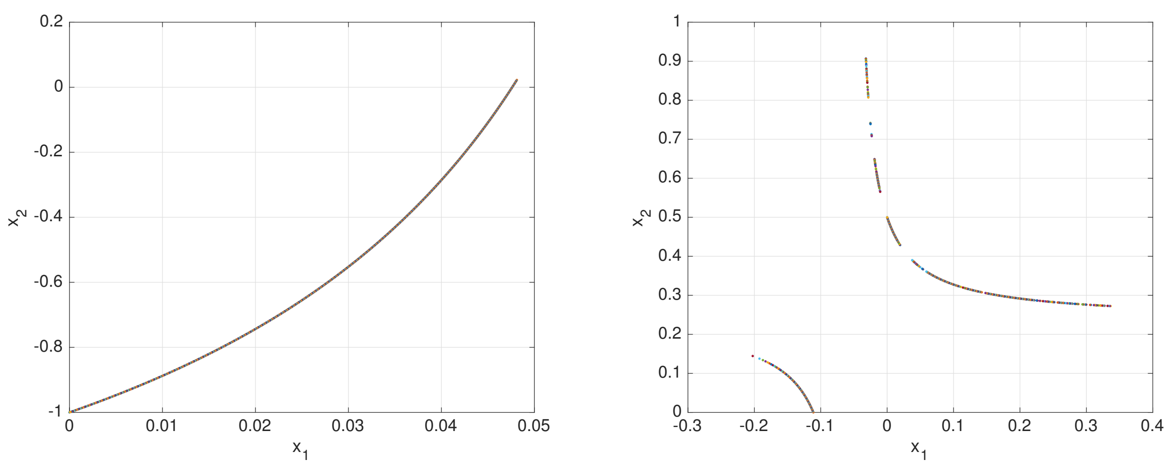
1.1 Plots



(a) Contour lines when $x = [0, -1]^T$.

(b) Contour lines when $x = [0, 0.5]^T$.

Figure 1: Contour lines of $m_k(p)$.



(a) Solution family of x when initial $x = [0, -1]^T$.

(b) Solution family of x when initial $x = [0, 0.5]^T$.

Figure 2: Family of solutions as the trust region varies from 0 to 1.001133.

1.2 Comments

(a) We use MATLAB `ezcontour` to plot the contour lines of the quadratic model.
(b) Based on the Theorem 4.1 on *Numerical Optimization, 2nd Edition, p70*, we will have:

$$(B + \lambda I)p^* = -g \tag{1}$$

$$\lambda(\Delta - \|p^*\|) = 0 \tag{2}$$

$$(B + \lambda I) \text{ is positive semidefinite.} \tag{3}$$

$$\|p^*\| \leq \Delta \tag{4}$$

$$\lambda \geq 0 \tag{5}$$

Let $p^* = [p_1, p_2]^T$. For $x = [0, -1]^T$, by equation (1), we have

$$p_1 = \frac{2}{\lambda + 42}, \; p_2 = \frac{20}{\lambda + 20} \tag{6}$$

When $\lambda > 0$, we have $\|p^*\| = \Delta$. Then we have:

$$\sqrt{(\frac{2}{\lambda + 42})^2 + (\frac{20}{\lambda + 20})^2} = \Delta \tag{7}$$

By (5) and (7), we could know the $\Delta_{max} = \sqrt{\frac{1}{21^2} + 1} = 1.001133$. So we calculate p^* when decrease Δ from Δ_{max} to 0. (Please refer to the MATLAB function `lambda` in the source code list)

For For $x = [0, 0.5]^T$, it is similar. (Please refer to the MATLAB function `lambda2` in the source code list)

1.3 Source code list

```
function a4_1()
%      x = [0, -1]';
    x = [0, 0.5]';
    %% draw contour lines
    fig = figure;
    ezcontour(@(p1, p2) mk(x, [p1, p2]'));
    set(gcf, 'Position', [0 0 500 400]);
    set(gca,'FontSize',14);
    title('x = [0, 0.5]^T');
    print(fig, '-depsc', '-r0', '../figs/q1-2');

    %% family of solutions
    d = sqrt(1+(1/21^2));
    figure;
    hold on;
    scatter(x(1), x(2), '.');

    while d >= 0
        new_x = lambda(d, x);
```

```
%          new_x = lambda2(d, x);
%          disp(new_x);
        scatter(new_x(1), new_x(2), '.');
        d = d - 0.001;
    end

    set(gcf, 'Position', [0 0 500 400]);
    set(gca,'FontSize',14);
    grid on;
    box on;
    xlabel('x_1');
    ylabel('x_2');
end
```

```
function [ y, gy, hy ] = objFunc( x )
    t = num2cell(x);
    [x1, x2] = t{:};

    y = 10 * (x2 - x1^2)^2 + (1 - x1)^2;
    gy = [ 2*x1 - 40*x1*(- x1^2 + x2) - 2;...
        - 20*x1^2 + 20*x2];
    hy = [ 120*x1^2 - 40*x2 + 2, -40*x1;...
        -40*x1,      20];
end
```

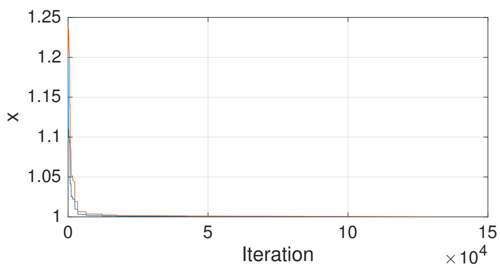
```
function [ m_k ] = mk( x_k, p )
    [y_k, gy_k, hy_k] = objFunc(x_k);
    m_k = y_k + gy_k' * p + 0.5 * p' * hy_k * p;
end
```

```
function [ y ] = lambda( delta, x )
    syms l positive;
    norm = sqrt(4/(42+l)^2 + 400/(20+l)^2);
    sol = vpasolve(norm == delta, l);
    y = [x(1) + 2 / (42 + sol); x(2) + 20 / (20 + sol)];
end
```

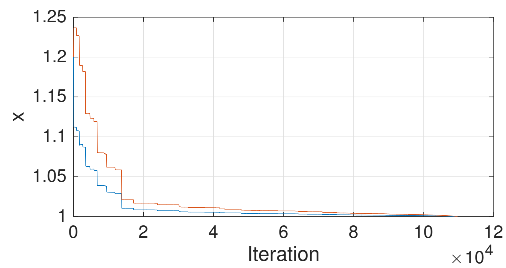
```
function [ y ] = lambda2( delta, x )
    syms l positive;
    norm = sqrt(4/(l-18)^2 + 100/(20+l)^2);
    sol = vpasolve(norm == delta, l);
    y = [x(1) + 2 / (sol-18); x(2) - 10 / (20 + sol)];
end
```

2 Problem 2

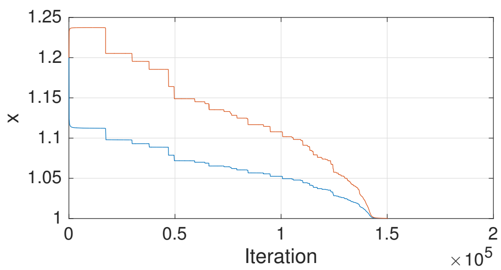
2.1 Plots



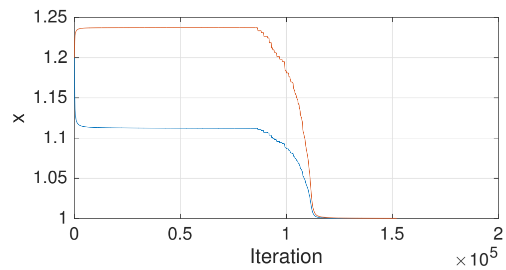
(a) $\Delta_{max} = 100$, $\eta = 0.15$ and $\Delta_0 = 0.01$.



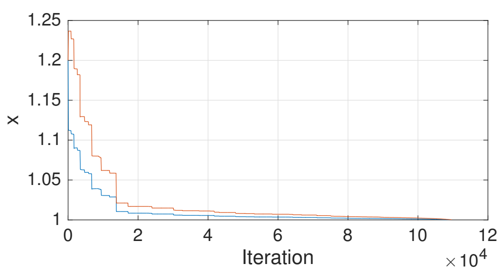
(b) $\Delta_{max} = 100$, $\eta = 0.15$ and $\Delta_0 = 0.1$.



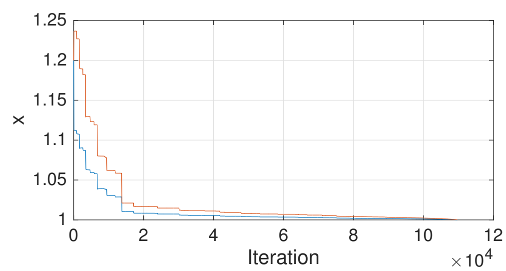
(c) $\Delta_{max} = 100$, $\eta = 0.15$ and $\Delta_0 = 2$.



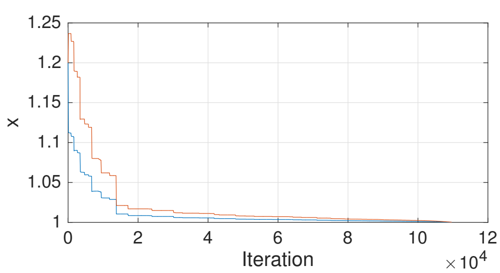
(d) $\Delta_{max} = 100$, $\eta = 0.15$ and $\Delta_0 = 10$.



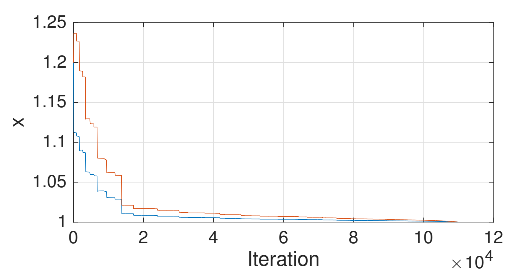
(e) $\Delta_{max} = 100$, $\eta = 0.001$ and $\Delta_0 = 0.1$.



(f) $\Delta_{max} = 100$, $\eta = 0.1$ and $\Delta_0 = 0.1$.



(g) $\Delta_{max} = 100$, $\eta = 0.2$ and $\Delta_0 = 0.1$.



(h) $\Delta_{max} = 100$, $\eta = 0.24$ and $\Delta_0 = 0.1$.

Figure 3: Convergence of x with different Δ_0 and different η.

2.2 Comments

The start point is $x = [1.2, 1.2]^T$ and the convergence is $x = [1, 1]^T$. By changing the value of Δ_0 and η, we obtain the Figure 3. As we can see, the value of Δ_0 has a dominant effect on the convergence of x, compared to η. When the Δ_0 gets larger, it takes more iterations to reach the convergence. The reason is the larger Δ_0 might not provide sufficient reduction ratio ρ_k in this case.

2.3 Output:

2.3.1 $\Delta_{max} = 100$, $\eta = 0.15$ and $\Delta_0 = 0.1$.

```
% i         xk(1)       xk(2)       pk(1)            pk(2)
  1          1.2         1.2          0               0
  2        1.12979     1.22915     -0.0702097       0.0291528
  3        1.12646     1.23061     -0.00332595       0.00145426
  4        1.12425     1.23158     -0.00221506       0.000968995
  5        1.12265     1.23227     -0.00159684       0.000698256
  6        1.12144     1.2328     -0.00121242      0.000529674
  7        1.12048     1.23322     -0.000955195       0.000416788
  8        1.11971     1.23356     -0.000773773       0.000337144
  9        1.11907     1.23384     -0.000640613       0.000278685
 10        1.11853     1.23407     -0.000539763       0.000234418
      ...
109643      1.00011     1.00023      -1.47544e-07      -2.86525e-07
109644      1.00011     1.00023      -3.02519e-08      -6.5147e-08
109645      1.00011     1.00023      -2.0697e-07      -4.23272e-07
109646      1.00011     1.00022      -2.92272e-08      -5.44762e-08
109647      1.00011     1.00022      -1.64695e-07      -3.209e-07
109648      1.00011     1.00022      -2.59721e-08      -5.61525e-08
109649      1.00011     1.00022      -1.57679e-07      -3.25178e-07
109650      1.00011     1.00022      -2.91813e-08      -5.43982e-08
109651      1.00011     1.00022      -1.64369e-07      -3.20309e-07
109652      1.00011     1.00022      -2.60094e-08      -5.62204e-08
```

2.4 Source code list

```
function a4_2()
    TOL = 0.0001;

    [xiter, output] = trustRegion([1.2, 1.2]', TOL);
    fig = draw(xiter);
    fig.PaperUnits = 'inches';
    fig.PaperPosition = [0 0 6 3];
    print(fig, '-depsc', '-r0', '../figs/q4-2-10');

    dlmwrite('../output/q4-2.txt', ...
            output, 'delimiter','\t', 'precision', '%6.6g');
end
```

```
function [ xiter, outputlist ] = trustRegion( x0, TOL )
    delta_max = 100;
    eta = 0.15;
    delta = 0.1;
    x = x0;
    it = 1;
    xiter{it} = x;
    outputlist(it, :) = [it, x(1), x(2), 0, 0];

    [~, gy, ~] = rosenbrock(x);

    while norm(gy) > TOL
        p_k = dogleg(@rosenbrock, x, delta);

        [y, gy, ~] = rosenbrock(x);
        [y1, ~, ~] = rosenbrock(x + p_k);
        rho_k = (y - y1) / (mk_rosenbrock(x, [0, 0]') - mk_rosenbrock(x, p_k));

        if rho_k < 0.25
            delta = 0.25*norm(p_k);
        elseif rho_k > 0.75 && norm(p_k) == delta
            delta = min(2*delta_old, delta_max);
        end

        if rho_k > eta
            x = x + p_k;
        end
        it = it + 1;

        xiter{it} = x;
        outputlist(it, :) = [it, x(1), x(2), p_k(1), p_k(2)];
    end
end
```

```
function [ p_k ] = dogleg( func, x, delta )
    [~, gy, hy] = func(x);

    p_u = - (gy'*gy) / (gy'*hy*gy) * gy;
    p_b = -inv(hy)*gy;

    % gauchy point (4.8)
    if gy'*hy*gy <= 0
        tau = 1;
    else
        tau = min(norm(gy)^3/(delta*gy'*hy*gy), 1);
    end

```

```
    if tau >= 0 && tau <=1
        p_k = tau * p_u;
    elseif tau > 1 && tau <= 2
        p_k = p_u + (tau - 1)*(p_b - p_u);
    end
end
```

```
function [ m_k ] = mk_rosenbrock( x, p )

    [y_k, gy_k, hy_k] = rosenbrock(x);
    m_k = y_k + gy_k' * p + 0.5 * p' * hy_k * p;

end
```

```
function [ fig ] = draw( x )
    figure;

    set(gcf, 'Position', [0 0 500 200]);
    set(gca,'FontSize',14);

    hold on;
    grid on;
    box on;

    X = NaN(length(x), 2);
    Y = NaN(length(x), 2);
    for i = 1:length(x)
        X(i,:) = [i, x{i}(1)];
        Y(i,:) = [i, x{i}(2)];
    end


    plot(X(:,1), X(:,2));
    plot(Y(:,1), Y(:,2));
    xlabel('Iteration');
    ylabel('x');

    fig = gcf;
end
```

3 Problem 3

3.1 Solution

$$\min_p m(p) = f + g^T p + \frac{1}{2} p^T B p,\ \ \|p|\ \leq \Delta,\ p \in span[g, B^{-1}g]$$

Let $p = \alpha g + \beta B^{-1} g$ and $u = (\alpha, \beta)^T$.

$$J(u) := \|p\|^2 = \alpha^2 \|g\|^2 + \beta^2 \|B^{-1}g\|^2 + 2\alpha\beta(g, B^{-1}g) = \frac{1}{2} u^T \bar{B} u$$

And,

$$\bar{B} = 2 \begin{bmatrix} \|g\|^2 & (B^{-1}g, g) \\ (B^{-1}g, g) & \|B^{-1}g\|^2 \end{bmatrix}$$

Obviously, $\bar{B}$ is symmetric positive definite. We derive the constraint for u from $\|p\| \leq \Delta$:

$$J(u) \leq \Delta^2 \tag{8}$$

Let $h(u) := m(p)$:

$$\begin{aligned} h(u) &= f + (\alpha g + \beta B^{-1} g, g) + \frac{1}{2}(\alpha g + \beta B^{-1} g, \alpha B g + \beta g) \\ &= f + \tilde{g}^T u + \frac{1}{2} u^T \tilde{B} u \end{aligned}$$

$$\tilde{g} = \begin{bmatrix} \|g\|^2 \\ (B^{-1}g, g) \end{bmatrix},\ \tilde{B} = \begin{bmatrix} (Bg, g) & \|g\|^2 \\ \|g\|^2 & (B^{-1}g, g) \end{bmatrix}$$

Therefore the problem could be reduced to the following two dimensional problem: under the constraint 8, we have:

$$\min_{u \in R^2} h(u) \tag{9}$$

Let u^* be the minimizer of 9 with no constraint:

$$u^* = -\tilde{B}^{-1} g$$

If $J(u^*) \leq \Delta^2$, then u^* is the solution.
If $J(u^*) > \Delta^2$, solve the following problem: for $\lambda \geq 0, \nabla h(u) + \lambda \nabla J(u) = 0$.
This problem is equivalent to

$$(\tilde{B} + \lambda \bar{B}) u = -\tilde{g}$$

($\tilde{B} + \lambda \tilde{B}$ is symmetric positive definite. Then the solution is:

$$-(\tilde{B} + \lambda \bar{B})^{-1} \tilde{g}$$

To this end, we have: with the constrain $J(u) \leq \Delta^2$,

$$\text{argmin } h(u) = \begin{cases} u^*, & J(u^*) \leq \Delta^2 \\ -(\tilde{B} + \lambda \bar{B})^{-1} \tilde{g} & J(u^*) > \Delta^2 \end{cases}$$

And $\lambda \leq 0$ satisfies the following equation:

$$J(-(\tilde{B} + \lambda \bar{B})^{-1} \tilde{g}) = \Delta^2$$