Numerical Optimisation: Assignment 3

Student Number:13064947

January 2018

1 Exercise 1

The intuition behind the exercise is that we have two directions, steepest descent and the newton (quadratic point). Whereas the cauchy point (implemented in exercise 3) is a minimum in a steepest descent direction. The newton point is another direction formed by using the second order taylor series.

The idea is to search the 2 dimensional subspace comprising the span of these two vectors. If they are collinear we can use the cauchy point. If the trust radius also includes the newton point and we have a positive semi definite matrix B. Then we are done and can just give that point.

If we have $min_p m(p) = g^T p + \frac{1}{2} p^T B p$ and also that $V = span(g, B^{-1}g)$ and a trust region $||p|| \leq \Delta$. Then the newton point described above would be $p = -B^{-1}g$.

The hardest case is where the newton point is outside our trust region. In which case we hit a boundary and have a constrained optimisation problem.

Following the hint we will let $p = Va_v$, if we choose an orthonormal basis then this can be reformulated as $m(a_v) = g^T V a_v + \frac{1}{2} a_v^T V^T B V a_v$ subject to $||a_v|| \leq \Delta$.

If $B_v = V^T B V$ and $g_v = V^T g$ this gives $m(a_v) = g_v^T a_v + \frac{1}{2} a_v^T B_v a_v$.
(Note I use a_v because I later on use another 'a' later on in my proof).

Because we are now in 2-d space we can write $B_v = \begin{bmatrix} a & b \\ b & c \end{bmatrix}$ and $g_v^T = \begin{bmatrix} d & e \end{bmatrix}$ giving us the following minimisation problem.

$min_{x,y} \frac{1}{2} \begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} a & b \\ b & c \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} d & e \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$ subject to $x^2 + y^2 = \Delta^2$, with an equality condition at the boundary. (Due the the newton point being outside our trust region).

We can solve for this by forming a lagrangian and taking partial derivatives, with respect to x,y and λ and setting the partial derivatives equal to 0.

$$\mathcal{L} = ax^2 = cy^2 + 2bxy + 2dx + 2ey + \lambda(x^2 + y^2 - \Delta^2) = 0 \tag{1}$$

$$ax + by + d + \lambda x = 0 \tag{2}$$

$$cy + bx + e + \lambda y = 0 \tag{3}$$

$$x^2 + y^2 = \Delta^2 \tag{4}$$

From equation (2) we have that

$$\lambda x = \frac{-ax - by - d}{x} \tag{5}$$

and plugging equation (5) into equation (3) we can remove λ giving

$$bx^2 - by^2 + (c - a)xy - dy + ex = 0 \tag{6}$$

now it is possible to re-parameterise equation (4) using the following substitutions for x and y.

$$x = \frac{2t\Delta}{1 + t^2} \tag{7}$$

and

$$y = \frac{(1 - t^2)\Delta}{1 + t^2} \tag{8}$$

this gives

$$\frac{4bt^2\Delta^2}{(1 + t^2)^2} - \frac{b(1 - t^2)^2\Delta^2}{(1 + t^2)^2} + \frac{2t(c - a)\Delta^2(1 - t^2)}{(1 + t^2)^2} - \frac{d\Delta(1 - t^2)}{1 + t^2} + \frac{2et\Delta}{1 + t^2} = 0 \tag{9}$$

which gives us a quartic equation if we multiply throughout by $(1 + t^2)^2$.

$$4bt^2\Delta^2 - b\Delta^2(1 - 2t^2 + t^4) + (c - a)\Delta^2(2t - 2t^3) - d\Delta(1 - t^4) + 2e\Delta(t + t^3) = 0 \tag{10}$$

collecting terms this gives

$$t^4(d\Delta^2 - b\Delta^2) + t^3(-2c\Delta^2 + 2a\Delta^2 + 2e\Delta) + t^2(6b\Delta^2) + t(2c\Delta^2 - 2a\Delta^2 + 2e\Delta) + (-b\Delta^2 - d\Delta) = 0 \tag{11}$$

Notice that coefficients a,b,c are multiplied by Δ^2 and d,e by Δ.

In the implementation, then the real roots of this polynomial are gathered using matlab's roots function. This gives up to a maximum of 4 potential solutions. We substitute to get the 4 potential (x,y) points using equations (7) and

(8). Then we test each of these values in our original minimisation problem to see which gives the lowest value.

The point giving the lowest of these values is the return value from our solver.

2 Exercise 2, 3

These were cody implementations in matlab. The first an implementation of the function just described in exercise 1. Returning either the full newton point, the cauchy point or the point found in the proof just given. The latter was a full implementation subsequently tested on the Rosenbrock function.

All parameter values were as provided bar a maximum delta which didn't appear to be specified. I choose a maximum delta of 1.

3 Exercise 4

In exercise 4, we tested using two starting points (1.2,1.2) fig 1 and (-1.2,1.0) fig 2. As can be shown in fig 1, the convergence to the optimal point (1.0,1.0) was very fast. Less than 10 iterations from the easier starting point. The trust region radius immediately increased and was stable. I started with a radius of 0.2 and a maximum possible radius of 1.0

In fig 2, we still see very good convergence from a difficult starting point. Around 25 iterations with all parameters the same as given above. There was however a difference in the trust region behaviour with it both growing and shrinking at various points during the optimisation depending upon how well the quadratic model matched the actual function at each iteration.

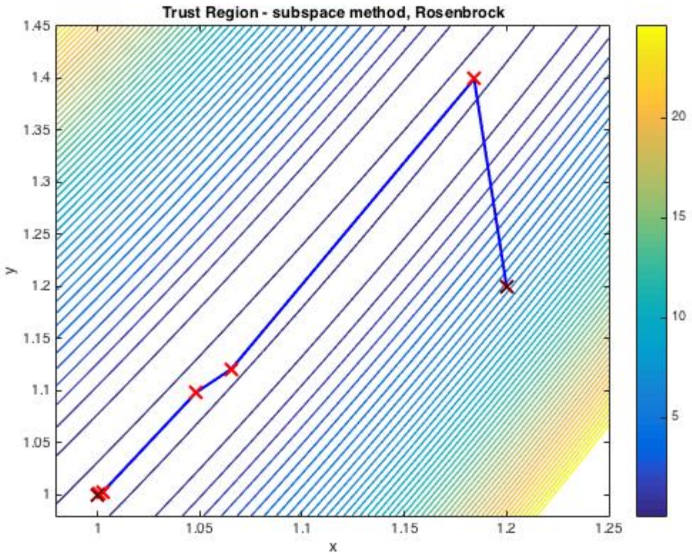

Figure 1: Trust Region Min - Subspace Method (1.2,1.2)

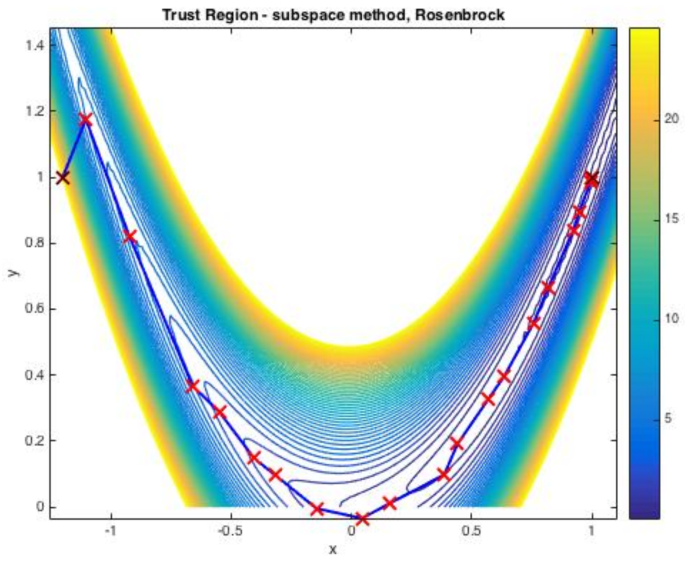

Figure 2: Trust Region Min - Subspace Method (-1.2, 1.0)