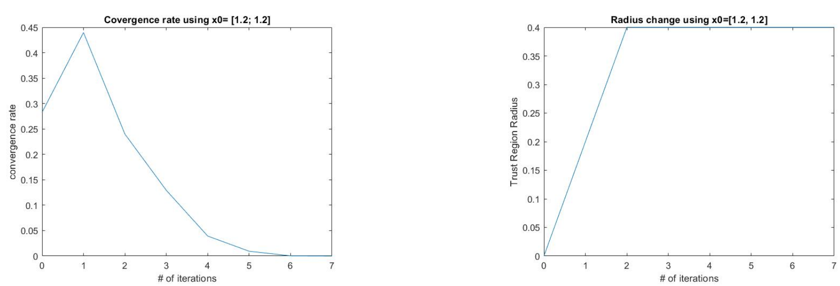
Numerical Optimisation cw2

Wenguang Lin

February 2020

1 Exercise 5

It is required to apply the trust region method to the Rosenbrock function with a nearby point (1.2, 1.2) and a remote point (-1.2, 1). I have chosen the 2D subspace implementation as a quadratic solver for this problem. The figure 1 shows how the rate of convergence and trust region radius changes with using a nearby satrting point (1.2, 1.2). The rate of convergence is obtained by $||x_k - x^*||_2^2$ for k=1,2,... As you can observe in the figure 1a, it converges at the 8^th iteration. The 2D subspace algorithm is used to capture the structure, for example it can collect the information of gradient and hessian to reach the elongated valley. The trust region radius is kept within $\triangle_k = 0.25 = \frac{1}{2}\tilde{\triangle}$ for all k. The radius is converged to 0.4 finally. Due to a small numbers of iteration in this case, it is very hard to define the convergence rate. Theoretically, it is a super linear convergence, because the hessian part is positive definite.

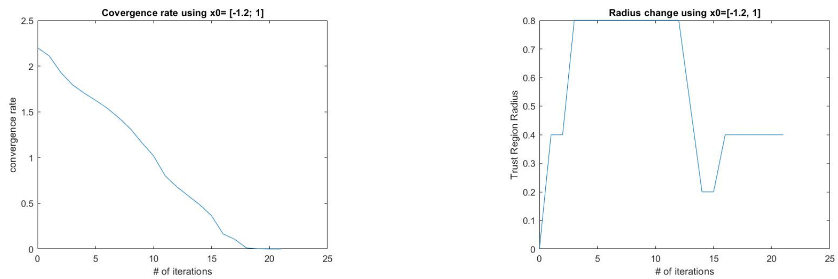


(a) Rate of Convergence

(b) Trust Region Radius

Figure 1: Starting point (1.2, 1.2)

The figure 2 shows how the rate of convergence and trust region radius changes with using a nearby starting point (-1.2, 0). For this case, it takes 22 iterations to reach the convergence which is better than the performance using line search. Since there are more iterations here, it can be clearly seen that it converges linearly in the figure 2a. The trust region radius is converged at 0.4 finally. It can be also observed that the maximum value of trust region radius is reached in the first 4 iterations and then it drops rapidly at around the $12^t h$ iteration. From this, we can see it shows a broader search firstly, then it has a finer search with a smaller radius when it approaches the minimal point.



(a) Rate of Convergence

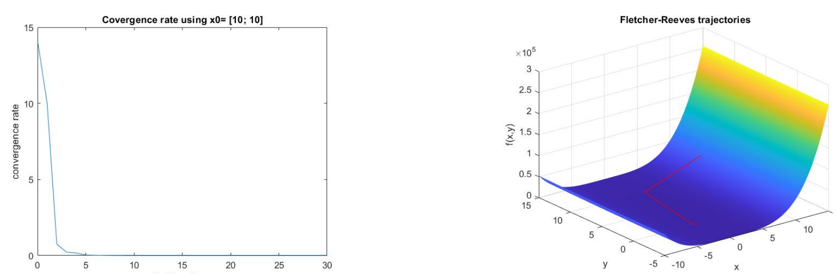
(b) Trust Region Radius

Figure 2: Starting point (-1.2, 1)

2 Exercise 7

Minimise the function using the initial point $x_0 = (10, 10)^T$ and $x_0 = (-5, 7)^T$ and tolerance $tol = 1 - 12$:

$$f(x, y) = x^2 + 5x^+10y^2 \tag{1}$$

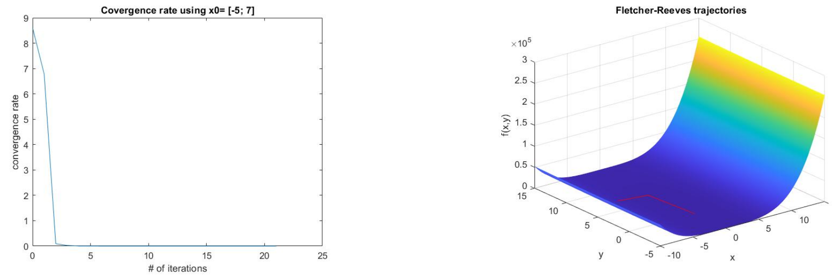


(a) Rate of Convergence

(b) trajectories

Figure 3: Fletcher-Reeves Starting point (10, 10)

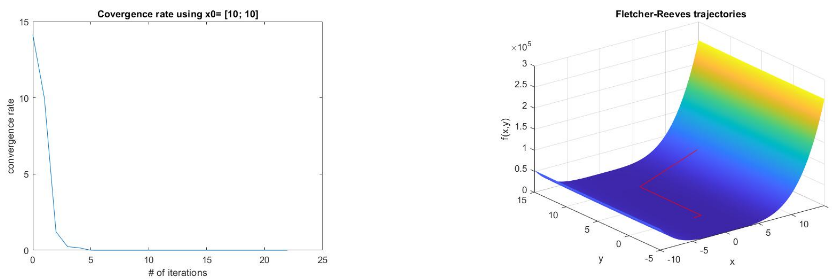
The figure 3 shows the rate of convergence and trajectories with using Fletcher-Reeves conjugate gradient method and (10, 10) as the starting point. It takes 30 iterations to converge.
The figure 4 shows the rate of convergence and trajectories with using Fletcher-Reeves conjugate gradient method and (-5, 7) as the starting point. It takes 21



(a) Rate of Convergence

(b) trajectories

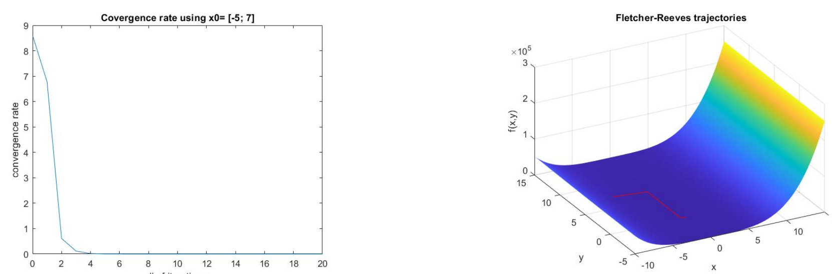
Figure 4: Fletcher-Reeves Starting point (-5, 7)



(a) Rate of Convergence

(b) trajectories

Figure 5: Fletcher-Reeves Starting point (10, 10)



(a) Rate of Convergence

(b) trajectories

Figure 6: Fletcher-Reeves Starting point (-5, 7)

iterations to converge.
The figure 5 shows the rate of convergence and trajectories with using Polak-Ribiere conjugate gradient method and (10, 10) as the starting point. It takes

22 iterations to converge.
The figure 6 shows the rate of convergence and trajectories with using Polak-Ribiere conjugate gradient method and (-5, 7) as the starting point. It takes 20 iterations to converge.

The convergence graphs here are quantified by using the same methods as the one in exercise 5.($||x_k - x^*||_2^2$). The potential problem can be detected in the graphs of convergence rate. It is very quick to find the region which is close to the minimal point. However, when it gets close to the region, it would converge slowly to the minimal point. Like what it has been shown in all convergence rate graphs, at the end of the slope, the gradient becomes smoother/ milder as it is converging to the minimal point.
The way I proposed to ensure convergence: We can use another methods to continue the optimisation problem at the moment when the convergence rate is getting slow down as using the conjugate gradient method. The slowing down phenomena can be quantified by $||x_k - x_{k-1}|| < \delta$. For example, another method can be trust region, because it is theoretically super linear.

References