MA40050 Coursework

mje45

Problem 1: Inequality

(1) For B SPD and $x \in \mathbb{R}^N$ with $x \neq 0$, prove:

$$(x^T x)^2 \leq (x^T B x)(x^T B^{-1} x) \tag{1.1}$$

Proof. Consider the Cauchy inequality:

$$(\overline{x}^T \overline{y})^2 \leq (\overline{x}^T \overline{x})(\overline{y}^T \overline{y}) \tag{1.2}$$

Setting $\overline{x} = B^{\frac{1}{2}} x$ and $\overline{y} = B^{-\frac{1}{2}} x$:

$$\begin{aligned}(x^T x)^2 = ((B^{\frac{1}{2}} x)^T (B^{-\frac{1}{2}} x))^2 = (\overline{x}^T \overline{y})^2 &\leq ((B^{\frac{1}{2}} x)^T (B^{\frac{1}{2}} x))((B^{-\frac{1}{2}} x)^T (B^{-\frac{1}{2}} x)) \\ &= (x^T B^{\frac{1}{2}} B^{\frac{1}{2}} x)(x^T B^{-\frac{1}{2}} B^{-\frac{1}{2}} x) = (x^T B x)(x^T B^{-1} x)\end{aligned} \tag{1.3}$$

This holds as we have that if B is SPD then so are $B^{\pm\frac{1}{2}}$ and that the inverse of a SPD matrix is also SPD. The $\leq$ represents (1.2). Moving onto equality. We get equality iff $x = Bx$ and $x = B^{-1}x$. I.e. if x is an eigenvector of B with the eigenvalue of 1.

□

Problem 2: Theory Work

For this problem we consider H_n as solely SPD and that $x_n^N \neq x_n^U$

(1) Prove that the following inequality holds for the above assumptions.

$$\gamma \ = (x_n^N - x_n^U)^T (x_n - x_n^U) < 0 \tag{2.1}$$

Proof. To start, let:

- $\nabla f_n = g_n \neq 0$
- $\nabla f_n^T H_n \nabla f_n = g_n^T H_n g_n = \eta \neq 0$

First substitute the values of x_n^N and x_n^U into γ (using the list above for simplification of notation) and work through:

$$\begin{aligned}\gamma &= \left(\frac{|g_n|^2}{\eta} g_n - H_n^{-1} g_n\right)^T \cdot \left(\frac{|g_n|^2}{\eta} g_n\right) \\ &= \frac{|g_n|^4}{\eta^2} g_n^T g_n - \frac{|g_n|^2}{\eta} g_n^T H_n^{-1} g_n \\ &\leq \frac{|g_n|^2}{\eta^2} (g_n^T H_n g_n)(g_n^T H_n^{-1} g_n) - \frac{|g_n|^2}{\eta} g_n^T H_n^{-1} g_n \\ &= \frac{|g_n|^2}{\eta} (g_n^T H_n^{-1} g_n) - \frac{|g_n|^2}{\eta} (g_n^T H_n^{-1} g_n) \\ &= 0\end{aligned} \tag{2.2}$$

The inequality is due to Problem 1, with $x = g_n$ and $B = H_n$. This is still a less than equals until we consider if g_n is an eigenvector of H_n. This is not possible due to the orthogonality of the Hessian applied to vectors and the gradient. Hence the inequality is strict. □

(2) Deduce $|x_n^U - x_n| < |x_n^N - x_n|$ and that there exists a unique point $x_n^D(\delta)$ on the dogleg path, such that $|x_n^D(\delta) - x_n| = \delta$. Compute $x_n^D(\delta)$ for $|x_n^U - x_n| \leq \delta \leq |x_n^N - x_n|$.

Proof. Assume that $|\nabla f_n| \neq 0$ and let $g_n = \nabla f_n$

$$\begin{aligned} |g_n| \cdot |x_n^N - x_n| &= |H_n^{-1} g_n| \cdot |g_n| \\ &\geq g_n^T H_n^{-1} g_n \\ &= \frac{(g_n^T H_n^{-1} g_n)(g_n^T H_n g_n)}{g_n^T H_n g_n} \\ &\geq \frac{|g_n|^4}{g_n^T H_n g_n} \\ &= |g_n| \cdot |x_n^U - x_n| \end{aligned} \tag{2.3}$$

Where the second inequality is true by the statement in Problem 1. $\gamma < 0$ implies that the angle between $x_n^U - x_n$ and $x_n^N - x_n^U$ is more than $\frac{\pi}{2}$ and hence the equality is not possible (see figure 4 for more detail). This shows that, assuming $|\nabla f_n| \neq 0$, that $|x_n^U - x_n| < |x_n^N - x_n|$. For the unique intersection part of the proof, we consider both cases: $\delta \leq |x_n^U - x_n|$ and $|x_n^U - x_n| < \delta \leq |x_n^N - x_n|$. For the first case we clearly have a unique intersection as $x_n^U - x_n$ is a radius of the trust region of radius δ. As $\gamma < 0$ we have that the vectors $x_n^U - x_n$ and $x_n^N - x_n^U$ have a separation of $\theta \in (\frac{\pi}{2}, \frac{3\pi}{2})$ due to the dot-product & cos identity. Adding this to (2.1) we finally have that there is a unique intersection when $\delta \leq |x_n^U - x_n|$. For the second case ($|x_n^U - x_n| < \delta \leq |x_n^N - x_n|$), we need a more robust approach. Consider δ_1 and δ_2 with $\delta_1 < \delta_2$, such that $x_n^D(\delta_{1/2})$ (not equal) lie on the line segment from x_n^U to x_n^N as shown in the images below:

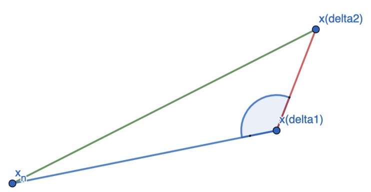

Figure 1: Zoomed View of $x_n^D(\delta_{1/2})$

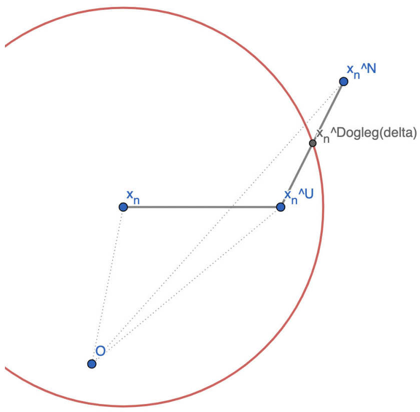

Figure 2: Overall View of the Setup

The angle in the fig.1 will be $\phi \in (\frac{\pi}{2}, \frac{3\pi}{2})$ (as $\theta \in (\frac{\pi}{2}, \frac{3\pi}{2})$) and hence $cos(\phi) \in (-1, 0)$. We get (with use of the cosine triangle formula):

$$\begin{aligned} |x_n^D(\delta_2) - x_n|^2 &= |x_n^D(\delta_1) - x_n|^2 + |x_n^D(\delta_2) - x_n^D(\delta_1)|^2 \\ &\quad - 2|x_n^D(\delta_1) - x_n| \cdot |x_n^D(\delta_2) - x_n^D(\delta_1)| \cdot cos(\phi) \\ &> |x_n^D(\delta_1) - x_n|^2 \end{aligned} \tag{2.4}$$

I.e. as we increase the radius of the trust region for $|x_n^U - x_n| < \delta \leq |x_n^N - x_n|$ we get a unique intersection. Hence it is shown that there is a unique point on the dogleg path such that $|x_n^D(\delta) - x_n| = \delta$.

To calculate an explicit formula for $x_n^D(\delta)$, it is necessary to draw a plot to help:

To start the proof of the formula for $x_n^D(\delta)$, we need to consider the "percentage" of which $x_n^D(\delta)$ is along $x_n^N - x_n^U$. To start this we consider the dot product formula:

$$(x_n^N - x_n^U) \cdot (x_n - x_n^U) = \gamma = |x_n^N - x_n^U| \cdot |x_n - x_n^U| cos(\theta) \tag{2.5}$$

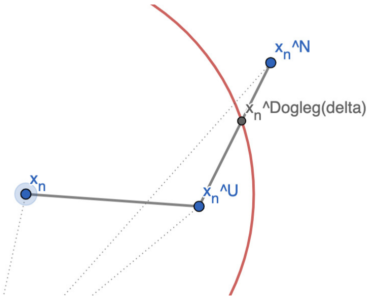

Figure 3: Zoomed View of $x_n^D(\delta)$

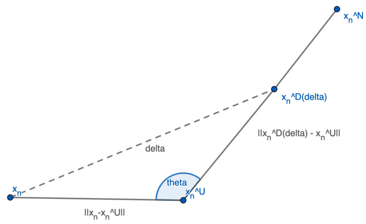

Figure 4: Set up for formula proof

This in turn implies that ($|x_n^N - x_n^U| \neq 0$):

$$|x_n - x_n^U|cos(\theta) = \frac{\gamma}{|x_n^N - x_n^U|} \tag{2.6}$$

Next consider the triangle with corners: x_n,x_n^U and $x_n^D(\delta)$. We can then use the general cosine triangle formula;

$$c^2 = a^2 + b^2 - 2abcos(C) \tag{2.7}$$

We can input in then:

- $a = |x_n^U - x_n|$
- $b = |x_n^D(\delta) - x_n^U|$
- $c = \delta$
- $C = \theta$

We then get a quadratic equation in $|x_n^D(\delta) - x_n^U|$:

$$\delta^2 = |x_n^U - x_n|^2 + |x_n^D(\delta) - x_n^U|^2 - 2|x_n^U - x_n| * |x_n^D(\delta) - x_n^U| * cos(\theta) \tag{2.8}$$

Using the quadratic formula then we get the solution for $|x_n^D(\delta) - x_n^U|$ as (must take positive square root as it's the only way that $|x_n^D(\delta) - x_n^U|$ will be positive, which is a must):

$$|x_n^D(\delta) - x_n^U| = |x_n^U - x_n|cos(\theta) + \sqrt{|x_n^U - x_n|^2 cos^2(\theta) + (\delta^2 - |x_n^U - x_n|^2)} \tag{2.9}$$

Substituting in the (2.5) where appropriate, we get:

$$|x_n^D(\delta) - x_n^U| = \frac{\gamma}{|x_n^N - x_n^U|} + \sqrt{\left(\frac{\gamma}{|x_n^N - x_n^U|}\right)^2 + (\delta^2 - |x_n^U - x_n|^2)} \tag{2.10}$$

So dividing this by $|x_n^N - x_n^U|$ gives the percentage $x_n^D(\delta)$ is along $x_n^N - x_n^U$ (set equal to t_{min}):

$$t_{min} = \frac{|x_n^D(\delta) - x_n^U|}{|x_n^N - x_n^U|} = \frac{\gamma + \sqrt{\gamma^2 + (\delta^2 + |x_n^U - x_n|^2)}}{|x_n^N - x_n^U|^2} \tag{2.11}$$

Finally we get using a linear interpolation of the line from x_n^U to x_n^N:

$$x_n^D(\delta) = x_n^U + t_{min}(x_n^N - x_n^U) \tag{2.12}$$

□

(3) Prove if $\gamma < 0$, the function $\delta \mapsto m_n(x_n^D(\delta))$ is strictly decreasing for $|x_n^U - x_n| \leq \delta \leq |x_n^N - x_n|$.

Proof. First, let's consider the steps made at each iteration. So in this proof only we consider:

- x_n^N as $x_n^N - x_n$ and;
- x_n^U as $x_n^U - x_n$

This is because we have $x_n^N - x_n^U = (x_n + x_n^N) - (x_n + x_n^U)$. Next, we consider the function:

$$\phi(t) = m_n(x_n + x_n^U + t(x_n^N - x_n^U)) \tag{2.13}$$

And we take the derivative of this w.r.t. t and let $z(t) = x_n + x_n^U + t(x_n^N - x_n^U)$:

$$\begin{aligned} \frac{d}{dt}\phi(t) &= \frac{d}{dt}m_n(z(t)) \\ &= g_n^T \cdot (x_n^N - x_n^U) + (x_n^N - x_n^U)^T H_n x_n^U + t(x_n^N - x_n^U)^T H_n (x_n^N - x_n^U) \\ &= (x_n^N - x_n^U)^T \cdot (g_n + H_n x_n^U) + t(x_n^N - x_n^U)^T H_n (x_n^N - x_n^U) \\ &= (x_n^N - x_n^U)^T H_n (H_n^{-1} g_n + x_n^U) + t(x_n^N - x_n^U)^T H_n (x_n^N - x_n^U) \\ &= -(x_n^N - x_n^U)^T H_n (x_n^N - x_n^U) + t(x_n^N - x_n^U)^T H_n (x_n^N - x_n^U) \\ &= -(1-t)(x_n^N - x_n^U)^T H_n (x_n^N - x_n^U) \end{aligned} \tag{2.14}$$

So we have that for $t \in (0,1)$ that $\frac{d}{dt}\phi(t) < 0$ ($H_n \geq 0$) and hence the model is decreasing as t increases. As t is a strictly increasing function of δ (by (2.3)), we have that $\delta \mapsto m_n(x_n^D(\delta))$ is strictly decreasing for $|x_n^U - x_n| \leq \delta \leq |x_n^N - x_n|$. □

Problem 3: Implement the dogleg method for solving the trust region subproblem

For this section, I tested the code in several ways. I first looked individual values of the output of the dogleg function and compared these to pre-described setups I had written to be able to see if the output was correct. These included setups where;

- $g_n' * H_n * g_n <= 0$
- H_n was either ill-conditioned (here $cond(H_n) >= 1e9$) or singular (here $det(H_n) < 1e-10$)
- $|x_n^N| <= \Delta_n$
- $|x_n^U| >= \Delta_n$

This was done with random points on 4 different surfaces to be able to get all the required tests. The surfaces are:

- Rosenbrock Function: $z(x,y) = (1-x)^2 + 10(y-x^2)^2$
- A 2D parabola: $z(x,y) = x^2 + y^2$
- A 2D exponential: $z(x,y) = 1 - e^{(x^2+y^2)}$
- The sin-sqrt function: $z(x,y) = sin(sqrt(x^2+y^2))$

Secondly, I tested this as part of the total *tr_dogleg* function that will be introduced later. This was to be able to use the visualise function properly and to be able to see where all the values of the dogleg end up.

In addition to these two main testing methods, I also displayed what sub-method in the dogleg method was used. This allowed me to pinpoint the sections where repeated errors were occurring on the visuals. I also used breakpoints in MATLAB to be able to pinpoint errors further and to be able to see what variables were doing what and how they were interacting step at a time. This was used in conjunction with the option to pause at any errors or warnings in the code.

Problem 4: SR1 Quasi-Newton Updates Theory

(1) Prove H_{n+1} satisfies the Secant condition.

Proof. Consider $H_{n+1}d_n$:

$$\begin{aligned} H_{n+1}d_n &= H_n d_n + \frac{(y_n - H_n d_n)(y_n - H_n d_n)^T}{d_n^T(y_n - H_n d_n)} d_n \\ &= H_n d_n + \frac{(y_n - H_n d_n)(y_n - H_n d_n)^T d_n}{d_n^T(y_n - H_n d_n)} \\ &= H_n d_n + (y_n - H_n d_n) \\ &= y_n \end{aligned} \tag{4.1}$$

□

(2) Prove that H_{n+1} is invertible and that:

$$H_{n+1}^{-1} = H_n^{-1} + \frac{(d_n - H_n^{-1}y_n)(d_n - H_n^{-1}y_n)^T}{(d_n - H_n^{-1}y_n)^T y_n} \tag{4.2}$$

Proof. First consider the Sherman–Morrison–Woodbury Formula (SMW) with both $U, V \in \mathbb{R}^{\mathbb{N}}$. First we need to show that H_{n+1} invertible. We do this by considering the statement: "$B + UV^T$ is invertible iff $\mathbb{I} + V^T B^{-1} U$ is invertible". Consider:

$$B = H_n, \quad \alpha = (d_n^T(y_n - H_n d_n))^{-1}, \quad U = \alpha(y_n - H_n d_n), \quad V = (y_n - H_n d_n) \tag{4.3}$$

Then substituting the values in for $\mathbb{I} + V^T B^{-1} U$:

$$\mathbb{I} + V^T B^{-1} U = 1 + \alpha(y_n - H_n d_n)^T H_n^{-1}(y_n - H_n d_n) \neq 0 \tag{4.4}$$

The last $\neq$ in (4.4) is because H_n is SPD $\Rightarrow$ H_n^{-1} is SPD. Also it cannot equal 0 unless $y_n = 0$. This is not possible as it would break the assumption: $(d_n - H_n^{-1}y_n)^T y_n \neq 0$. Therefore as this value can't be equal to 0, it is invertible in the reals, and hence so is $B + UV^T$. When we substitute in the values for B,U and V we get our formula for H_{n+1}, H_{n+1} is invertible.

As $H_{n+1} = B + UV^T$, we use the SMW Formula for the inverse of H_{n+1}^{-1};

$$H_{n+1}^{-1} = (B + UV^T)^{-1} = B^{-1} - \frac{B^{-1}UV^T B^{-1}}{\mathbb{I} + V^T B^{-1} U} \tag{4.5}$$

This is because both $U, V \in \mathbb{R}$. Substituting and working the algebra:

$$\begin{aligned} H_{n+1}^{-1} &= H_n^{-1} - \frac{H_n^{-1}\alpha(y_n - H_n d_n)(y_n - H_n d_n)^T H_n^{-1}}{1 + (y_n - H_n d_n)^T H_n^{-1}\alpha(y_n - H_n d_n)} \\ &= H_n^{-1} - \frac{(H_n^{-1}y_n - d_n)(y_n^T H_n^{-T} - d_n^T H_n^{-1} H_n)}{\alpha^{-1} + (y_n^T H_n^{-T} - d_n^T H_n^{-1} H_n)(y_n - H_n d_n)} \\ &= H_n^{-1} - \frac{(d_n - H_n^{-1}y_n)(d_n^T - y_n^T H_n^{-T})}{\alpha^{-1} + (y_n^T H_n^{-1} y_n - y_n^T d_n - d_n^T y_n + d_n^T H_n d_n)} \\ &= H_n^{-1} - \frac{(d_n - H_n^{-1}y_n)(d_n - H_n^{-1}y_n)^T}{(d_n^T y_n - d_n^T H_n d_n) + (y_n^T H_n^{-1} y_n - y_n^T d_n - d_n^T y_n + d_n^T H_n d_n)} \\ &= H_n^{-1} - \frac{(d_n - H_n^{-1}y_n)(d_n - H_n^{-1}y_n)^T}{(d_n^T y_n - y_n^T H_n^{-1} y_n)} \\ &= H_n^{-1} - \frac{(d_n - H_n^{-1}y_n)(d_n - H_n^{-1}y_n)^T}{(d_n - H_n^{-1}y_n)^T y_n} \end{aligned} \tag{4.6}$$

□

Problem 5: Implement the SR1 Quasi-Newton Updates

Similar to the way I tested Problem 3, I used a variety of different matrices to be able to cover all possibilities. This was to be able to check that the if statement worked properly and that the SR1 update algorithm agreed with the different scenarios that I had set up. This ended up throwing no errors or warnings in the diverse scenarios that I had inputted. After this, It was again tested in the final *tr_dogleg* function and I was able to pinpoint errors within it using the necessary breakpoints.

Problem 6: Implementing trust region with dogleg and SR1 update

(1) This was tested over the functions listed earlier. I added the option to have a random input around a circle of suitably small radius. (Done to make use the sin-sqrt function was aptly defined and had a unique minimiser locally, due to the sinusoidal behaviour of the sin function). This was the same for the exponential function, due to the target ($|g_n| <= tol$) being met for more values than just the function's minimum. The other two functions didn't have this problem, but still had a random input entered from the 5x5 square centered at the origin. This allowed me to be able to see where the errors were and not just in the more 'scripted' inputs that I had lined up to do basic testing.

Again as with the previous two coding problems, I was able to test all the coding in one go with aid of the visualise function showing where seemingly different values were being generated.

Further, using the visual function, I was able to see the paths made by the functions and how close they appeared to what, theoretically, would be expected. The ideal path would reduce the gradient as soon as possible and then follow the gradient down. For our Rosenbrock function as shown in fig.10 below, we have that the algorithm 'falls' into the trough of the function and carries on towards the minimiser. This is also why Δ_n also stays larger closer to the minimiser as the gradient is really shallow so it needs a larger radius to be able to decrease the total number of iterations. This matches up with what is predicted.

(2) First is the considering the trust region radius. This appears to have a general down trend as shown in fig.5 below.

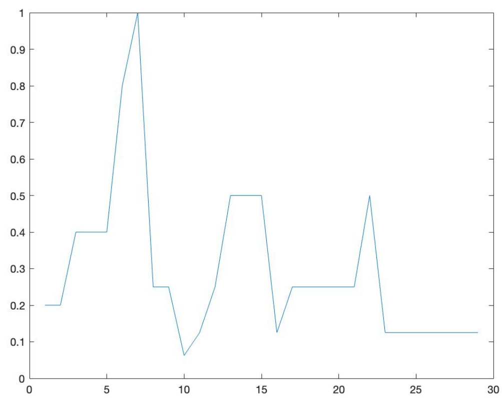

Figure 5: Δ_n

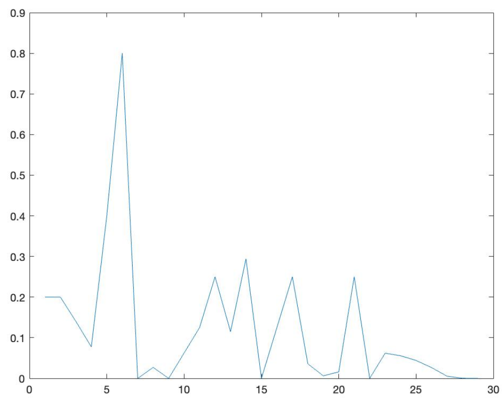

Figure 6: $|x_{n+1} - x_n|$

This is in agreement with the theory as the plot should only decrease if the ρ_n is lower than 0.25 but this only happens when the model value is far from the functions value, and hence needing a tighter search area for an appropriate iteration. We see here that overall the trust region radius only halves over the entire process. Considering the norm: $|x_{n+1} - x_n|$ over the iterations, we have the following graph in fig.6.This shows that this norm appears to approach zero over the dogleg process, while also taking larger steps where necessary to be able to approach the minimiser as quick as possible. This is consistent with results from the problem sheets and that the whole process converges super-linearly as explained now.

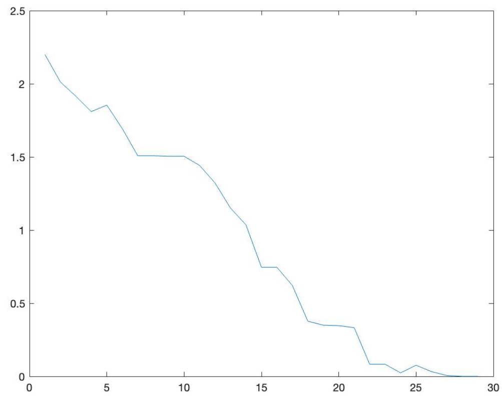

Figure 7: $|x_n - x^*|$

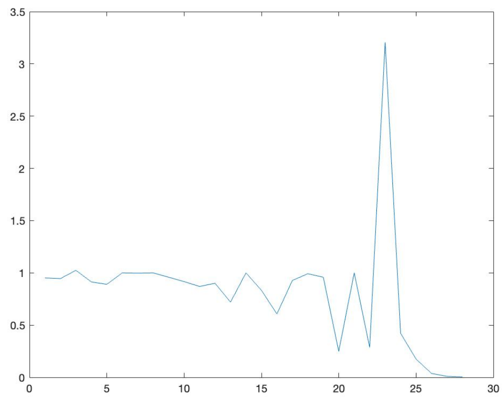

Figure 8: $\frac{|x_{n+1}-x^*|}{|x_n-x^*|}$

Fig.7 is iteration against $|x_n - x^*|$ and fig.8 is the ratio of subsequent $|x_n - x^*|$, i.e. $\frac{|x_{n+1}-x^*|}{|x_n-x^*|}$. As we can see the plot on the right approaches 0 (all bar one value is less than 1)(there is a steep slope there but this is due to the very small gradient close to the minimum of the Rosenbrock function). Hence super-linearity is displayed. This aligns with methods implemented and tested in the problem sheets.

(3) Sensitivity Analysis Below are three plots each with different Δ_{max} values. (Fig.10 is the plot asked for in question 6b). Interestingly the difference was quite substantial. For $\Delta_{max} = 0.5$ there was 6 less iterations to be able to be within the tolerance. The number of iterations were 23,29,28 resp.

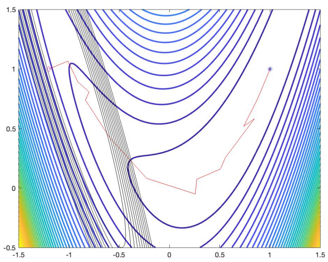

Figure 9: $\Delta_{max} = 0.5$

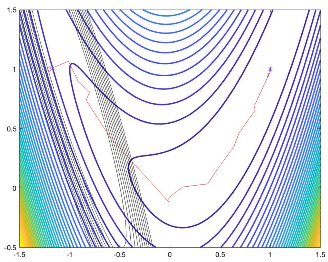

Figure 10: $\Delta_{max} = 1$

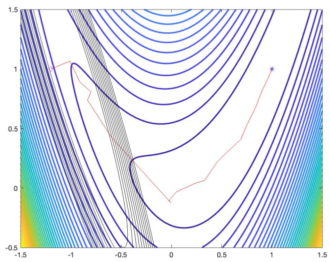

Figure 11: $\Delta_{max} = 10$

I would argue that the substantial drop in iterations when $\Delta_{max} = 0.5$ would suggest that the method is sensitive to changes in Δ_{max}.

Next is looking at the sensitivity on the ρ_{ac} parameter. Again, following are three plots with different values for ρ_{ac}. The case where $\rho_{ac} = 0.125$ is the same as Fig.10.

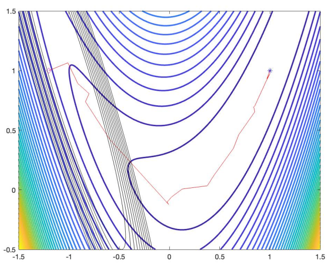

Figure 12: $\rho_{ac} = 0.25$

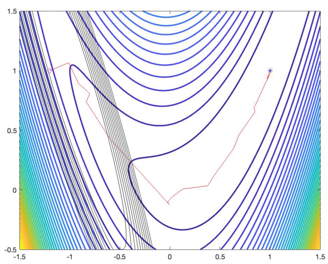

Figure 13: $\rho_{ac} = 0.5$

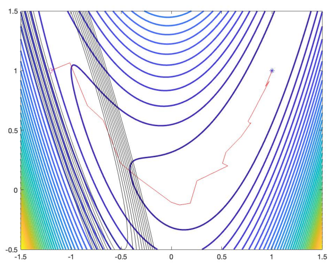

Figure 14: $\rho_{ac} = 0.75$

Interestingly there was no changed in the number of iterations for the changes in ρ_{ac} until $\rho_{ac} = 0.75$. This is because most of the values for ρ_n are more than 0.6 and hence a difference is marginal at the lower values of ρ_{ac}.

Finally onto the sensitivity of η. Again the 3 graphs below show a variety of values for η (1e-7,1e-2,1). As before Fig.10 represents $\eta = 1e-5$.

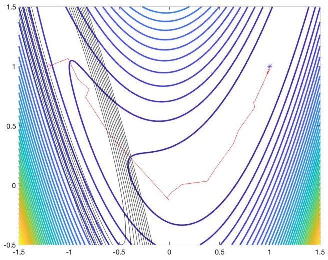

Figure 15: $\eta = 1e-7$

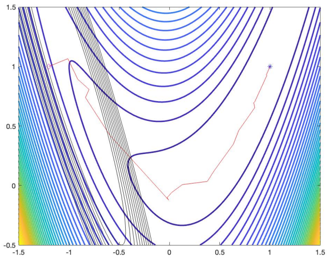

Figure 16: $\eta = 1e-2$

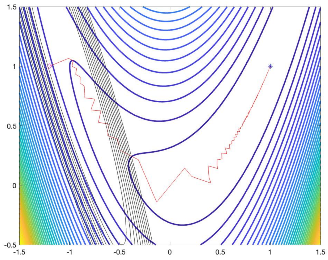

Figure 17: $\eta = 1$

The number of iterations doesn't change for $\eta < 1e-1$ apart from $\eta = 1e-2$ which appears to be the "sweet spot" where the number of iterations drops down by 1 to 28. When $\eta >= 1$ we get that the number of iterations shoots up to over 1200. This is because more of the SR1 updates are just outputting the inputs and there seems to be a clustering point around 1. So in all, there doesn't seem to be much of a sensitivity around the value of η unless it is over 1.