Numerical Optimization
Re-exam Handin 5

Dmitry Serykh (qwl888)

May 29, 2020

# 1 The Setup

## 1.1 Parameters

I used following parameter values in my implementation. Some values were taken from the literature, while others were determined empirically.

# 2 Testing protocol

In order to test the effectiveness of my implementation, I came up with a testing protocol, where I used following metrics:

- The convergence plots with the number of iteration on the x-scale and the Euclidean distance between the current value of $x$ and the optimum. The resulting plot can be seen on Figure **??**.
- The convergence plots with the trust region radius. The resulting plot can be seen on Figure **??**.
- **Accuracy**. The Euclidean distance to the optimum at the termination point. The results can be seen on Table **??**. The performance of my implementation of the trust region algorithm is compared to the performance of the line search methods from the previous assignment, namely Steepest Descent and Newton's Algorithm.
- **Efficiency**. The number of steps until the gradient magnitude reaches $10^{-7}$. The results can be seen on Table **??**.

I used a random starting point taken from the uniform distribution in the interval between $-10$ to 10 and repeated each optimization 100 times for all metrics and took the average. This was done to remove the influence of the starting position from the results of the optimization. I used the mean to minimize the effect of the outliers.

As suggested, I have modified my implementation of the Log-Epsilon function and excluded the first Attractive Region function from the experiments.

# 3 Theoretical Part

In the theoretical part, I will show that Newtons Method is invariant to all invertible linear transformations of the objective function.

Assume that Newtons Method with backtracking line-search is run on $f$ with starting point $x_0$, producing a sequence of iterates $x_1, \dots x_T$ with step-candidates $p_1, \dots p_T$. Further, assume we ran the algorithm on: $g(x) = f(Ax)$, where $A$ is square and invertible and choose as starting-point $\tilde{x}_0 = A^{-1}$ leading to iterates $\tilde{x}_1, \dots \tilde{x}_T$ with step candidates $\tilde{p}_1, \dots \tilde{p}_T$.

## 3.1

I will show that the hessian is $\nabla^2 g(x) = A^T \left[\nabla^2 f(Ax)\right] A$. This can be shown by application of the chain rule. Hence, we get:

$$
\begin{aligned}
g(x) &= f(Ax)\\
\nabla g(x) &= A^T \nabla f(Ax)\\
\nabla^2 g(x) &= A^T \left[\nabla^2 f(Ax)\right] A
\end{aligned}
$$

### 3.2

Then, I assume that $\tilde{x}_k = A^{-1}x_k$ and show that $\tilde{p}_k = A^{-1}p_k$. The search direction for the Newton's method is: $-\nabla^2 f^{-1}\nabla f$, therefore:

$$\begin{aligned}
\tilde{p}_k &= -(\nabla^2 g\left(\tilde{x}_k\right)^{-1})\nabla g\left(\tilde{x}_k\right) \\
&= \left(-A^T\nabla^2 f\left(x_k\right)A\right)^{-1} A^T\nabla f\left(x_k\right) \\
&= -A^{-1}\nabla^2 f\left(x_k\right)^{-1}(A^T)^{-1}A^T\nabla f\left(x_k\right) \\
&= A^{-1}(-\nabla^2 f\left(x_k\right)^{-1})I\nabla f\left(x_k\right) \\
&= A^{-1}\left(-\nabla^2 f\left(x_k\right)^{-1}\nabla f\left(x_k\right)\right) \\
&= A^{-1}p_k
\end{aligned}$$

### 3.3

Next, I will show that $\alpha_k$ stays the same after transformation. We assumed that $\tilde{x} = A^{-1}x$ and showed that $\tilde{p}_k = A^{-1}p_k$. Therefore, we have:

$$\begin{aligned}
g(\tilde{x}_k + \alpha\tilde{p}_k) &= f(A(\tilde{x}_k + \alpha\tilde{p}_k)) \\
&= f(A\tilde{x}_k + A\alpha\tilde{p}_k) \\
&= f(AA^{-1}x + AA^{-1}\alpha p_k) \\
&= f(x + \alpha p_k)
\end{aligned}$$

$$\begin{aligned}
\nabla g(\tilde{x}_k)^T\tilde{p}_k &= \nabla g(A^{-1}x_k)^T A^{-1}p_k \\
&= (A^T\nabla f(AA^{-1}x_k))^T A_k^{-1} \\
&= \nabla f(AA^{-1}x_k)^T AA^{-1}p_k \\
&= \nabla f(x_k)^T p_k
\end{aligned}$$

$$\begin{aligned}
g(\tilde{x}_k) &= g(A^{-1}x_k) \\
&= f(AA^{-1}x_k) \\
&= f(x_k)
\end{aligned}$$

During Week 3, we have used a backtracking line search to find the value of $\alpha_k$, which was the first matching value of $\alpha$, for which it holds:

$$f(x_k + \alpha p_k) \leq f(x_k) + c\alpha\nabla f_k^T p_k$$

Let us assume that we have found a value of $\alpha$, that matches the inequality for iteration $k$ for the original function $f$. Then, by the three qualities that I have just shown, following inequlity would also hold for the same value of $\alpha$ and $c$:

$$g(\tilde{x}_k + \alpha\tilde{p}_k) \leq g(\tilde{x}_k) + c\alpha\nabla g_k^T\tilde{p}_k$$

Hence, the same value of $\alpha$ can be used as $a_k$ after the transformation and I can conclude that the step size $a_k$ stays the same.

### 3.4

I will show that $\tilde{x}_k = A^{-1}x_k$ holds for all $k \geq 0$ by induction.

We have assumed that we choose starting point $\tilde{x}_0 = A^{-1}x_0$, so the induction step holds at

$k = 0$.

Let us also assume that $\tilde{x}_k = A^{-1}x_k$ holds for $k-1$. Then, from Step 2, we have that:

$$\tilde{p}_{k-1} = A^{-1}p_{k-1}$$

Moreover, we know from Step 3 that $\alpha_k$ stays the same after transformation and hence:

$$\begin{aligned}\tilde{x}_k &= \tilde{x}_{k-1} + \alpha\tilde{p}_{k-1} \\ &= A^{-1}x_{k-1} + A^{-1}\alpha p_{k-1} \\ &= A^{-1}(x_{k-1} + \alpha p_{k-1}) \\ &= A^{-1}x_k\end{aligned}$$

$\tilde{x}_k = A^{-1}x_k$ holds for all $k \geq 0$ by weak induction.

Finally, we have:

$$g(\tilde{x}_k) = f(x_k)$$

### 3.5

The consequence of this property of the Newton's Method is that application of invertible linear transformations would not in any way affect the performance of the optimizer. Than can influence my choice of testing protocol in the future.

# 4 Conclusion