Numerical Optimization
Assignment 5
University of Copenhagen

Alexander Schou (drm499)
Laufey Karitas (kdc408)
Patrick Jensen (mxf667)
Jingchao Zhou (vjb382)
Contribution: Equal

March 2023

Introduction

This report documents a benchmark protocol for testing our own implementation of Steepest Descent and Newtons Algorithm under linear equality constraints. The report starts with theoretical considerations that we will use to discuss our results and testing procedures. We will discuss which steps we took to ensure that our implementation is correct, after which we will evaluate the performance of our implementations. We will create a variation of linear equality constraints to construct an initial point that fulfills the constraint, and ensure that some of the linear equalities prevent that the unconstrained minimum is feasible. Lastly, we will present and discuss the results obtained from testing.

Theory

Analytical Observations

We wish to solve the following problem:

$$\min_{\lambda} ||\nabla f(x_k) - A^T\lambda||^2$$

where $\lambda \in \mathbb{R}^m$, $A \in \mathbb{R}^{m\times n}$ and $x_k \in \mathbb{R}^n$. By using the definition of the norm we get that

$$||\nabla f(x_k) - A^T\lambda||^2 = (\nabla f(x_k) - A^T\lambda)^T(\nabla f(x_k) - A^T\lambda) = \nabla f(x_k)^T\nabla f(x_k) + \lambda^T AA^T\lambda - 2\lambda^T A\nabla f(x_k)$$

We differentiate and set it equal to zero

$$2AA^T\lambda - 2A\nabla f(x_k) = 0 \Leftrightarrow AA^T\lambda = A\nabla f(x_k)$$

We know that $AA^T \in \mathbb{R}^{m\times m}$ which means that it is symmetric. We also know that $rank(A) = m$ since we choose all linear equalities to be independent. This means that $rank(A^T) = m$ so $rank(AA^T) = m$. This means that AA^T is invertible. Hence, we get the following

$$AA^T\lambda = A\nabla f(x_k) \Leftrightarrow \lambda = (AA^T)^{-1}A\nabla f(x_k)$$

The condition $||\nabla f(x) - A^T\lambda||$ measures the distance between the gradient and the vector $A^T\lambda$ inside the cone. If we use the lambda that minimises the norm then we have a condition for how close we are to the optimum.

Lemma 7

To prove that $T_C(x) = \{p \in \mathbb{R}^2 | a_i^T p = 0\}$, we need to show each side is a subset of the other. For the first part of the proof, we prove that

$$T_C(x) \subseteq \{p \in \mathbb{R}^n | a_i^T p = 0\}.$$

Consider some $x \in \mathcal{C}$ and $p \in T_C(x)$, meaning that there exists a function $z(\alpha) : [0, u) \to \mathcal{C}, u > 0$, differentiable at $\alpha = 0$, with $z(0) = x$ and $z'(0) = p$. $\mathcal{C} = \{x \in \mathbb{R}^n | c_i(x) = 0, \forall i \in \mathcal{E}\}$ in our case, as we have no inequality constraints. The constraints are of the form

$$c_i(x) = a_i^T x + b = 0$$

Since $z(\alpha) : [0, u) \to \mathcal{C}$, we have that

$$a_i^T z(\alpha) + b = 0, \text{ for } \alpha \in [0, u)$$

This means that the gradient is 0 at those values of α, giving us

$$c_i'(z(\alpha)) = a_i^T z'(\alpha) = 0, \alpha \in [0, u)$$

More importantly,

$$a_i^T z'(0) = a_i^T p = 0$$

This shows us that

$$T_C(x) \subseteq \{p \in \mathbb{R}^n | a_i^T p = 0\}$$

For the second part of the proof, we show that

$$\{p \in \mathbb{R}^n | a_i^T p = 0\} \subseteq T_C(x)$$

Consider some function $z(\alpha)$, where $z(0) = x$ and $z'(\alpha) = p$ with $x \in \mathcal{C}$, $p \in \{p \in \mathbb{R}^n | a_i^T p = 0\}$ and $\alpha \in \mathbb{R}$. This implies that we also have $z'(0) = p$. Inserting this function into the LHS of linear equality constraint and taking the derivative w.r.t. α, we have that,

$$c_i'(z(\alpha)) = a_i^T z'(\alpha) = a_i^T p = 0, \text{ since } a_i^T p = 0 \text{ by definition.}$$

We know that since $x \in \mathcal{C}$, and $c'(z(\alpha)) = 0$ for $\alpha \in \mathbb{R}$, that $c(z(\alpha)) = 0$ for $\alpha \in \mathbb{R}$. This finally implies that $z(\alpha) : \mathbb{R} \to \mathcal{C}$ for which the restriction $z(\alpha) : [0, u) \to \mathcal{C}, u > 0$ also holds. Therefore we have shown that

$$\{p \in \mathbb{R}^n | a_i^T p = 0\} \subseteq T_C(x)$$

.

Experiments

The experiments included running the new versions of Newtons algorithm and Steepest Descent on all of the benchmark functions, with different sizes of dimensions and a varying range of starting points. We generate the constraint matrix A randomly. More precisely, we generate each element of the matrix A with a uniform distribution $\mathcal{U} \sim [-10, 10)$.

To construct an initial starting point that fulfills a constraint for a variation of linear constraints, we created a random matrix A with $m < n$ where we ensured that $rank(A) = m$. From this, we used the formula $x - A^{\dagger}(Ax + b)$ to generate a starting point that fulfills the constraint. Here b is a randomly generated vector of size n, in the range between 0 and 1. b being drawn randomly also makes it very unlikely that the constraints allow for the unconstrained optimum. We checked that the values of b did not allow for the unconstrained optimum.

We chose the stopping criteria for both algorithms to be $|\nabla f(x)^T p_k| < \epsilon$.

Correctness of algorithm

To evaluate the correctness of our code, we ran the algorithms on each of the benchmark functions over from 100 starting points in $\mathcal{N}(\mu = 0.5, \sigma^2 = 200)$ from which we observed that the algorithms converged towards our stopping criterion $|\nabla f(x)^T p_k| < 1.0e - 10$. We chose this stopping criterion, as we know that the p_k vector will be closest to the optimum when it is tangent to the lowest possible iso-curve, in which case p_k and $\nabla f(x_k)$ will be orthogonal. Furthermore, we checked in each iteration that $Ap_k < 1.0e - 10$ as well as $||\nabla f(x_k) + A^T \lambda||^2 < 1.0e - 10$, such that we are sure that we are fairly close to moving along the linear equality constraints under the limits of numerical precision.

From Figure 1, we can find that the constrained Newton Algorithm only makes steps on the hyperplane $Ax + b = 0$. So in most cases, it would not go to the global minimum.

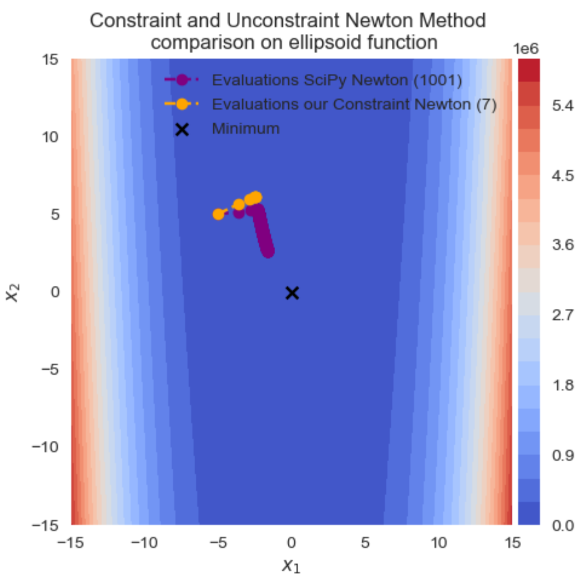

Figure 1: Comparison of convergence trajectory on Rosenbrock function, The yellow point is constrained Newton Algorithm, the purple one is the unconstrained Newton Algorithm. The starting point is $(-5, 5)$

Results

We found that the performance of Steepest Descent and Newtons Algorithms under linear equality constraint was very dependant on which function it ran on. In figure 2 we have included a plot of a comparison of the runtime between two of the benchmark functions. However it is important to note that the running time of Newton could be improved, as for the current implementation we compute the eigenvalues and vectors in each iteration, which is computationally inefficient.

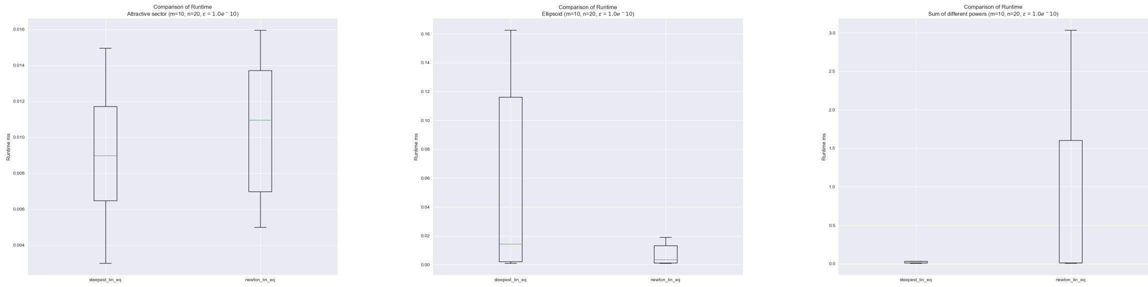

Figure 2: Computational time comparison, f4, f1 and f5. Note that the left boxes represent Steepest Descent and the Right represent Newtons Algorithm

In all the plots below, you can see that Newtons Algorithm converges to the minima in 1 to 50 iterations or so for all five benchmark functions. For Steepest Descent we observed that the algorithm is slower than Newtons as to getting towards the minimum. However in general it performs better in comparison to the prior versions of Steepest Descent, in terms of iterations. This is probably due to the stopping condition being fulfilled further away from optimum, where $||\nabla f(x_k)||$ is still fairly large.
Looking at the convergence plots compared to the performance box-plots we see that there is a connection to the convergence and how long the algorithm runs. E.g. for f5 even though Newton's Algorithm converges in fewer iterations it is a longer computational time, which is probably due to the calculations of the Hessians, and might look differently if we did not compute the eigenvalues and vectors in each iteration.

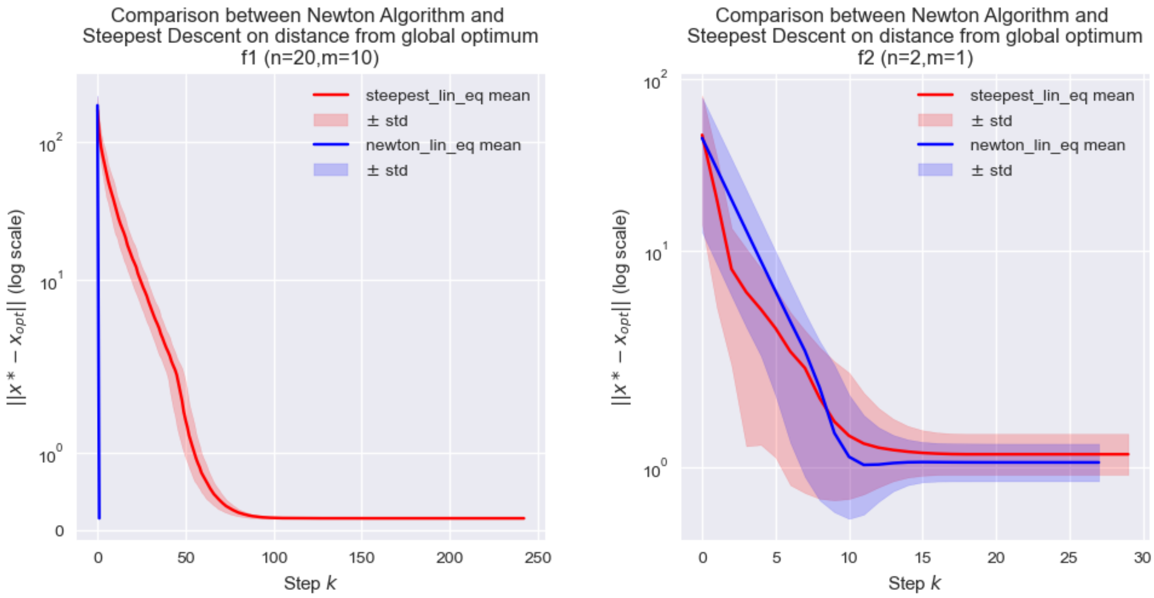

Figure 3: Convergence plots of Newtons and Steepest Descent under linear equality constraints in 20 dim, with different range of starting points, f1 and f2, respectively.

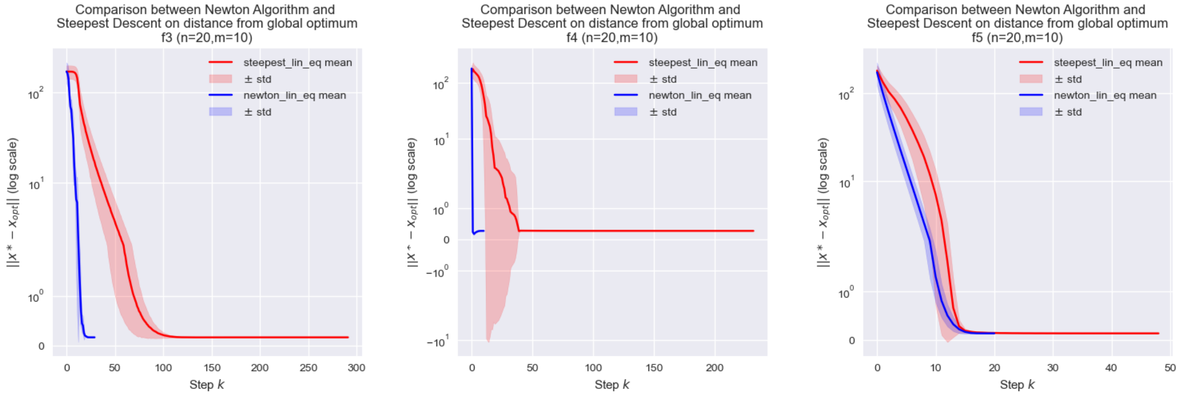

Figure 4: Convergence plots of Newtons and Steepest Descent under linear equality constraints in 20 dim, with different range of starting points, f3, f4 and f5, respectively.

We see that for steepest descent generally converges linearly on $f_1 - f_3$. However, the convergence curve kinks downward when the algorithm gets sufficiently close to the optimum. We also see superlinear convergence on f_4 and f_5. For newton, we see instant and near-instant convergence on f_1 and f_4, respectively. We also see superlinear convergence for f_2 and f_3. Finally we see that f_5 converges linearly, with a sudden somewhat of a kink when it gets sufficiently close to the optimum.

Conclusion

In conclusion we have managed to implement Steepest Descent and Newtons Algorithm under linear equality constraints and benchmarked all against the five benchmark functions. We have found that the optimizers are competitive to the prior algorithms, although dependant on the task at hand.