Numerical Optimization Assignment 2

Dilnaz-str989, Hamsa-nbp737, Rasmus-kpn134, Parth-tdh903

February 16, 2026

1 Introduction

This report answers the given theoretical question (described in Section 2), and the minimizer benchmarking (described in Section 3). In the theory section we prove that Newtons method has Q-linear, but not Q-superlinear convergence on the function assuming that $\alpha = 1$ and $x_{k,2} \neq 0$, on the case study function f_5, first with $d = 2$ then $d > 2$. In the benchmarking section, we showcase performance in terms of distance to the optimum value and number of convergence steps, before visualizing the Newton descent for each iteration, and finally discussing the results.

For the project, all team-members have contributed equally.

2 Theory

For $d = 2$

The given function $f_5(x_k) = \sum_{i=1}^{d} (x_{k,i}^2)^{(1+\frac{i-1}{d-1})}$ with $d = 2$, we get

$$f_5(x_k) = x_{k,1}^2 + x_{k,2}^4$$

$\bigtriangledown f_5(x_k) = (2x_{k,1}, 4x_{k,2}^3)^T$ and $\bigtriangledown^2 f_5(x_k) = \begin{bmatrix} 2 & 0 \\ 0 & 12x_{k,2}^2 \end{bmatrix}$

As $\bigtriangledown^2 f_5(x_k) > 0$ is positive definite $\forall x > 0$, we 'll use equation (11)[NumOp] Newton step direction and we find the inverse of 2×2 matrix by using the formula: $\begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \frac{1}{ad-bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$

$p_k = -(\bigtriangledown^2 f_5(x_k))^{-1} \bigtriangledown f_5(x_k) = -\begin{bmatrix} 2 & 0 \\ 0 & 12x_{k,2}^2 \end{bmatrix}^{-1} \begin{bmatrix} 2x_{k,1} \\ 4x_{k,2}^3 \end{bmatrix} = -\frac{1}{24x_{k,2}^2} \begin{bmatrix} 12x_{k,2}^2 & 0 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} 2x_{k,1} \\ 4x_{k,2}^3 \end{bmatrix} = \begin{bmatrix} -x_{k,1} \\ -\frac{1}{3}x_{k,2} \end{bmatrix}$

when $\alpha_k = 1$, then Newton iteration is , $x_{k+1} = x_k + 1 \cdot p_k$

$$x_{k+1} = \begin{bmatrix} x_{k,1} \\ x_{k,2} \end{bmatrix} + \begin{bmatrix} -x_{k,1} \\ -\frac{1}{3}x_{k,2} \end{bmatrix} = \begin{bmatrix} 0 \\ \frac{2}{3}x_{k,2} \end{bmatrix}$$

Since the function is always $f_5(x_k) \geq 0$, and it equals 0 at $x = (0, 0)$, there cannot be any point with a smaller value. Therefore, $x^* = (0, 0)$ is the global minimum. That is why we take $x^* = (0, 0)$ as the optimum for this function. After the first iteration $x_{k,1} = 0$ for all $k \geq 1$. Therefore, for all $k \geq 1$, $\|x_k - x^*\| = |x_{k,2}|$

$$\frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} = \frac{\sqrt{0^2 + (\frac{2}{3}x_{k,2})^2}}{\sqrt{x_{k,1}^2 + x_{k,2}^2}} = \frac{2/3|x_{k,2}|}{|x_{k,2}|} = \frac{2}{3}, \quad \text{where} \quad x_{k,2} \neq 0$$

By definition 2 [NumOp] The sequence $\{x_k\}$ converges Q-linearly to x^* if there exists a constant r with $0 < r < 1$ such that $\frac{\|x_{k+1}-x^*\|}{\|x_k-x^*\|} \leq r$

In our case, we obtained $\frac{\|x_{k+1}-x^*\|}{\|x_k-x^*\|} = \frac{2}{3}$. Since $0 < \frac{2}{3} < 1$, the Newtons method converges Q-linearly.

Q-superlinear convergence requires $\frac{|x_{k+1}-x^*|}{|x_k-x^*|} \to 0$ as $k \to \infty$. However, in our case $\frac{\|x_{k+1}-x^*\|}{\|x_k-x^*\|} = \frac{2}{3}$ which is a constant and does not converge to zero. Therefore, the convergence is not Q-superlinear.

For $d > 2$

$$f_5(x_k) = \sum_{i=1}^{d} (x_{k,i})^{2(1+\frac{i-1}{d-1})}$$

$$f_5(x_k) = x_{k,1}^{2(1+\frac{0}{d-1})} + x_{k,2}^{2(1+\frac{1}{d-1})} + ... + x_{k,d}^{2(1+\frac{d-1}{d-1})}$$

$$\bigtriangledown f_5(x_k) = \big(2(1+\frac{0}{d-1})x_{k,1}^{2*\frac{0}{d-1}+1},\ 2(1+\frac{1}{d-1})x_{k,2}^{2\frac{1}{d-1}+1},\ \cdots,\ 2(1+\frac{d-1}{d-1})x_{k,d}^{2\frac{d-1}{d-1}+1}\big)^T$$

$\bigtriangledown^2 f_5(x_k)$ is diagonal because the function is separable. Therefore

$$\bigtriangledown^2 f_5(x_k) = diag\big(2,\ 2(1+\frac{1}{d-1})(2\frac{1}{d-1}+1)x_{k,2}^{2\frac{1}{d-1}},\ \cdots,\ 2(1+\frac{d-1}{d-1})(2\frac{d-1}{d-1}+1)x_{k,d}^{2\frac{d-1}{d-1}}\big)$$

Since the Hessian is diagonal, its inverse is obtained by taking the reciprocal of each diagonal entry and because Hessian is diagonal, we compute Newton step direction coordinate by coordinate :

$$p_k = -(\bigtriangledown^2 f_5(x_k))^{-1} \bigtriangledown f_5(x_k) = -\left(x_{k,1},\ \frac{1}{2\frac{1}{d-1}+1}x_{k,2},\ \frac{1}{2\frac{2}{d-1}+1}x_{k,3},\ \cdots,\ \frac{1}{2\frac{d-1}{d-1}+1}x_{k,d}\right)^T$$

$$= -(\beta_1 x_{k,1},\ \beta_2 x_{k,2},\ \beta_3 x_{k,3},\ \cdots,\ \beta_d x_{k,d})^T d \quad \text{we define} \quad \beta_i = \frac{1}{2\frac{i-1}{d-1}+1}, \quad i = 1, \cdots, d$$

$$x_{k+1} = x_k + p_k\alpha_k = x_k + 1 \cdot p_k = ((1-\beta_1)x_{k,1},\ (1-\beta_2)x_{k,2},\ \cdots,\ (1-\beta_d)x_{k,d})^T$$

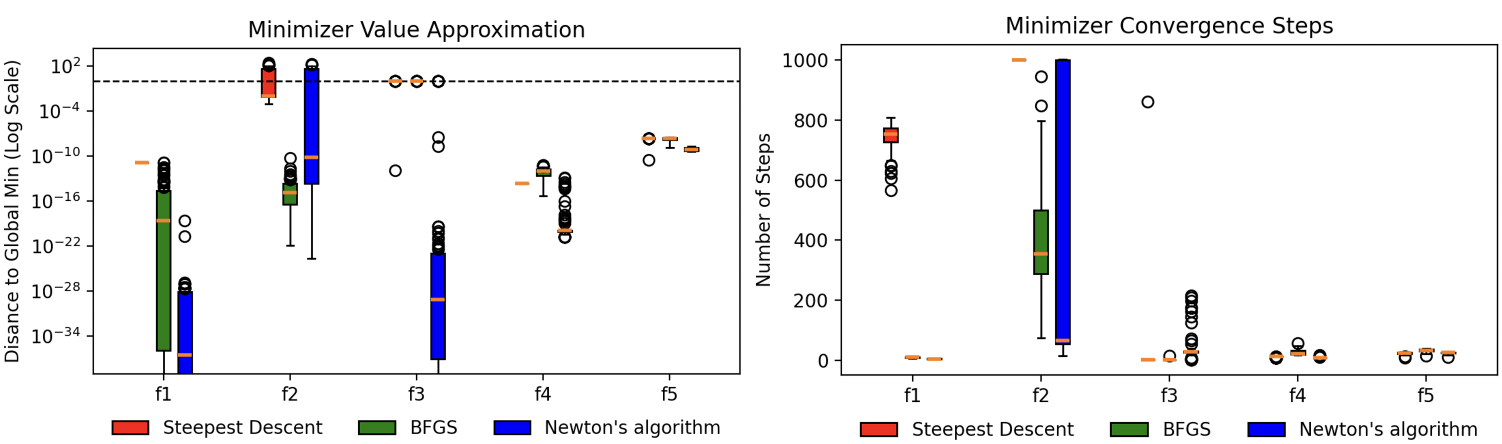
since $x^* = \vec{0}$ then we get,

$$\frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} = \frac{\|x_{k+1}\|}{\|x_k\|} = \frac{\sqrt{(1-\beta_1)^2 x_{k,1}^2 + (1-\beta_2)^2 x_{k,2}^2 + \cdots + (1-\beta_d)^2 x_{k,d}^2}}{\sqrt{x_{k,1}^2 + x_{k,2}^2 + \cdots + x_{k,d}^2}}$$

Let $1 - \beta^* = \max\{1-\beta_1, \ldots, 1-\beta_d\}$. Then$(1-\beta_i) \leq (1-\beta^*)$

$$= \frac{\sqrt{\sum_{i=1}^{d}(1-\beta_i)^2 x_{k,i}^2}}{\sqrt{x_{k,1}^2 + x_{k,2}^2 + \cdots + x_{k,d}^2}} \leq \frac{(1-\beta^*)\sqrt{x_{k,1}^2 + x_{k,2}^2 + \cdots + x_{k,d}^2}}{\sqrt{x_{k,1}^2 + x_{k,2}^2 + \cdots + x_{k,d}^2}} = (1-\beta^*)$$

As i increases, the denominator increases, hence β_i decreases. The smallest β_i occurs at $i = d$, so $\beta_d = \frac{1}{2+1} = \frac{1}{3}$. Thus $\beta^* = \frac{1}{3}$. Therefore $1 - \beta^* = \frac{2}{3}$. Thus, $\frac{\|x_{k+1}-x^*\|}{\|x_k - x^*\|} \leq \frac{2}{3}$, the result still holds.



Meds1	Vars1	Meds2	Vars2	Meds3	Vars3	Meds4	Vars4	Meds5	Vars5
0.00	0.00	0.01	3124.64	1.00	0.01	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	2485.15	0.00	0.09	0.00	0.00	0.00	0.00

Meds1	Vars1	Meds2	Vars2	Meds3	Vars3	Meds4	Vars4	Meds5	Vars5
754.50	1978.80	1000.00	0.00	1.00	7356.14	13.00	4.18	23.50	10.32
10.00	1.34	354.00	32665.62	1.00	1.67	22.50	71.72	33.50	19.84
3.00	1.15	66.00	173492.00	28.00	1821.87	8.00	3.71	25.00	7.75

Figure 1: Box-plot benchmarks of value distances (left) and convergence steps (right).

3 Methodology & Results

As shown on Figure 1, we benchmarked our implementation of Newton's algorithm against the `scipy` default BFGS, and our custom Steepest Descent for all five case functions (f_1-f_5). To provide a complete estimate of performance, we measured both the *optimum distance* $|f(x) - f(x^*)|$ (the difference between the optimizer output minus the optimal value) *and* the number of steps before convergence. We executed each estimator with 100 different starting points (of dimension 2 to support f_2) randomly sampled from a normal distribution $\mathcal{N}(0, 100)$, which provided a good tradeoff between sample size, point diversity, and execution time when running the benchmarks. To further limit the execution time, we set the maximum iterations to 1000 and gradient threshold to $\epsilon = 1 \cdot 10^{-5}$.

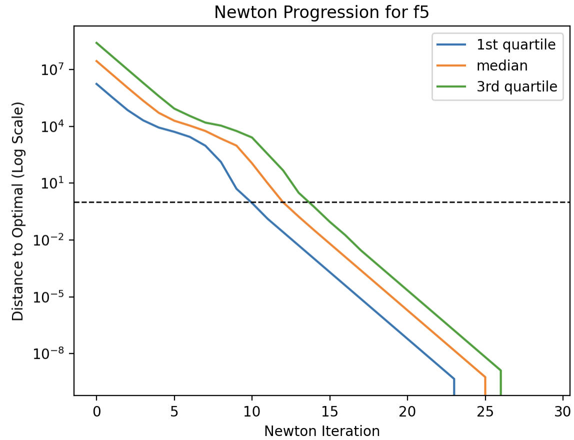


Figure 2: Newton convergence progression

On Figure 1, we visualized the resulting distributions of distances and steps (for the 100 points) using box-plots, in order to show the general trends, range of results, and extent of outliers. The figure is organized with the functions along the x-axis, which each include the three plots of Steepest Descent, BFGS, and Newton's algorithm in that order from left to right. Outliers are shown as circular dots and medians as orange bars. Below each plot the corresponding medians Meds_i and variances Vars_i are written explicitly. For convenience, we marked the $y = 1$ value on plots with logarithmic scale, as a dotted horizontal line.

To study the convergence rate, we additionally plotted the optimum distance as a function of the minimizer iterations for our Newton's method on f_5. The result is shown on Figure 2, which shows the 1st quartile, the median, and the 3rd quartile for the distances at each iteration of the algorithm (for 100 randomly sampled points).

4 Discussion & Conclusion

As seen in Figure 2, the convergence rate exhibits an approximately linear decrease for the 1st, 2nd, and 3rd distance quartiles. A slight curvature is observed between iterations 5–10, which may be attributed to the backtracking line search reducing the step length α to satisfy the first Wolfe condition. Aside from this effect, the progression remains consistent with the theoretically derived Q-linear convergence, supporting the correctness of our Newton implementation.

To further validate correctness, we benchmarked our implementation against the default `scipy` Newton method and obtained identical plots.

The overall performance of Newton's algorithm relative to Steepest Descent and BFGS is shown in Figure 1. In the left distance plot (log scale), values below the dotted $y = 1$ line correspond to solutions close to the optimum. Newton's method performs comparably to, and in some cases better than, the other methods. For f_3, Steepest Descent and BFGS stagnate at a distance of 1, whereas Newton converges further within 27 additional iterations. Overall, Newton's method demonstrates strong performance on the tested problems.