AMATH 515: HOMEWORK 4 COMPUTATIONAL REPORT

PRANAV DOSHETTY

Applied Mathematics Department, University of Washington, Seattle, WA
pranavrd@uw.edu

1. Introduction

This report collects the results of analysis done on optimization algorithms — Steepest Descent, Newton's Method, and two quasi-Newton methods, namely BFGS and DFP. The goal of the analysis is to see which algorithm performs the best on the objective functions. I have chosen Rosenbrock problem, Logistic Regression problem on MNIST data, and Zakharov problem to test the algorithms on. To draw conclusions on the performance of each algorithm, I will measure the number of iterations required for convergence, assess whether the norm of the gradient of the function falls within a specified tolerance, and tune the parameters for line search using the Wolfe conditions. Additionally, I will perform further tuning and analyses to determine what each algorithm requires to achieve convergence.

2. Methods

1. Line Search (with Wolfe Conditions)
All four algorithms use a bisection line search to determine step sizes, ensuring sufficient decrease and curvature conditions via Wolfe conditions:

- **(W1)**Ensures a decrease in the objective function proportionate to the product of the step size and the directional derivative, i.e., the step size is neither too large nor too small.
- **(W2)** Ensures that the directional derivative at x_{k+1} is sufficiently more negative than the directional derivative at x_k, i.e., the step size adheres to the positive curvature.

The step size α is controlled by a (lower bound), b (upper bound), and $t = t_0$ (initial guess, set to 1.0). If t_0 is too large or small, it is refined using a and b. A `NaN` in function values signals step size issues, numerical instability, or an undefined function point.

2. Steepest Descent (SD): Steepest Descent is an optimization method that updates the sequence $\{x_k\}$ using only the negative gradient direction.

3. Newton's Method: This technique uses the second-order derivative, the Hessian, to update $\{x_k\}$, achieving faster convergence for functions that are convex and have positive definite Hessian matrices.

4. BFGS: Broyden-Fletcher-Goldfarb-Shanno: Approximates the Hessian (or its inverse) using gradient evaluations, avoiding the computational cost of computing the exact Hessian while maintaining positive definiteness.

5. DFP: Davidon-Fletcher-Powell: Similar to BFGS but updates the Hessian directly instead of approximating its inverse, often at a higher computational cost.

Date: 02/15/2025.

3. Results

In this section, I will discuss how the optimization techniques behave when applied to an objective function. I have used the Zakharov function to demonstrate the typical behavior of the algorithms, i.e., it is anticipated that the algorithms converge in the order $Newton < BFGS < DFP < SD$. Following that, I test the techniques on the Rosenbrock function and Logistic Regression as well.

3.1 Zakharov : The Zakharov function is a continuous, convex, and non-separable benchmark function used in optimization, defined as

$$f(\mathbf{x}) = \sum_{i=1}^{n} x_i^2 + \left(\sum_{i=1}^{n} 0.5ix_i\right)^2 + \left(\sum_{i=1}^{n} 0.5ix_i\right)^4$$

with a global minimum at $\mathbf{x} = \mathbf{0}$. From Figureit is observed that the result is as anticipated. The order of number of iterations taken is the same as mentioned above.

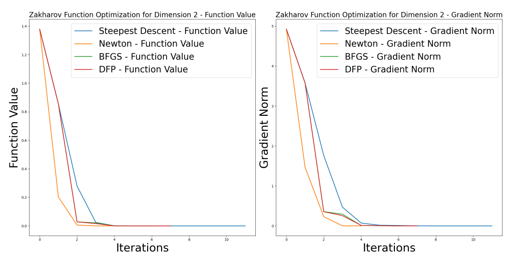
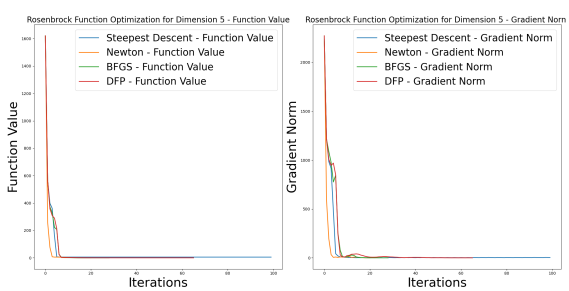
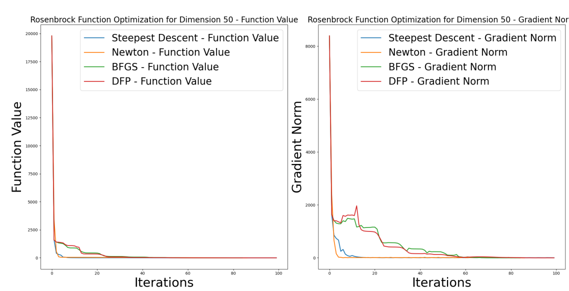


Figure 1. Plot of the Zakharov function optimization results.

3.2 Rosenbrock : On the Rosenbrock function, Newton's method and BFGS converge significantly faster than steepest descent, requiring far fewer iterations to reach a low function value and gradient norm. Steepest descent struggles with high-dimensional cases, leading to a substantial increase in iteration count and computation time. As for DFP, it takes more steps than BFGS and the cumulative time for it to reach the final value is also significantly higher. See Figure 2



(A) n=5

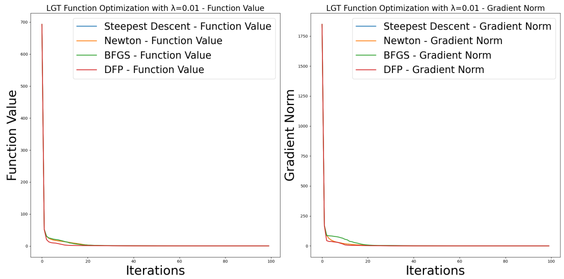


(B) n=50

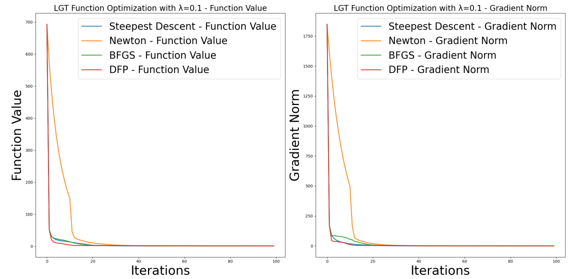
Figure 2. Plot of the Rosenbrock optimization results

3.3 Logistic Regression : For LGT it is observed that the Newton's method converges similarly to SD as the Hessian is not invertable. Of the four, The BFGS gives the best performance as regularization value increases. See Figure 3 for reference.

3.3.1 Hyperparameter tuning for LGT :. The results in Table 1 show that tuning $c1$ and $c2$ significantly impacts convergence rates. Lower values of $c1$ generally result in fewer iterations but may lead to instability in some cases. The BFGS and DFP methods exhibited large iteration counts for certain settings, suggesting sensitivity to $c2$ variations. Newton's method consistently maintained lower gradient norms but required more iterations when $c1$ was small.



(A) λ=0.01



(B) λ=0.1

FIGURE 3. Plot of the LGT function optimization results.

Method	c1	Iterations	Final Function Value	Final Gradient Norm
Newton	0.5	164	0.03127	0.0001
BFGS	0.5	172	0.03127	0.0001
DFP	0.5	912	0.03127	9e-05
Newton	0.7	143	0.03127	0.0001
BFGS	0.7	423	0.03127	0.0001
DFP	0.7	896	0.03127	0.0001
Newton	0.9	184	0.03127	0.0001
BFGS	0.9	1396	0.03127	0.0001
DFP	0.9	1172	0.03127	9e-05

TABLE 1. Results of hyperparameter tuning

4. Summary and Conclusions

Drawing from observations on both the objective functions, it is suffice to say that SD is slow but reliable, requiring significantly more iterations to converge, especially on the Rosenbrock function. Newton's method converges quickly with low final gradient norms but is computationally expensive due to Hessian evaluations. BFGS is efficient when well-tuned but sensitive to hyperparameters, sometimes leading to instability. DFP behaves similarly to BFGS but generally requires more iterations to achieve comparable accuracy.

Acknowledgements

I sincerely appreciate everyone who contributed to this report. Special thanks to Professor Bamdad Hosseini for insightful lectures and TA Alexander W. Hsu for invaluable support in resolving errors. I'm grateful to Sid for refining the objective function and Lamanisa Saputra for LaTeX assistance. Additionally, GenAI tools like ChatGPT and Perplexity helped clarify concepts.

This homework deepened my understanding of optimization algorithms and challenged me to grow in new directions.

References

[1] Dhanishtha Sharma, *Zakharov Function*, Indusmic, `https://www.indusmic.com/post/zakharov-function`, 2025.
[2] Sonja Surjanovic and Derek Bingham, *Test Functions and Datasets*, Virtual Library of Simulation Experiments, `https://www.sfu.ca/~ssurjano/optimization.html`, 2025.