Assignment 1: Optimization of a Bivariate Quadratic function using the equal interval search method

Submitted by:
Ajinkya Suranjan Salve
Roll No.: SC24M005
Subject : Multidisciplinary Design Optimization
Branch: Aerodynamics and flight mechanics

Date:
29 January 2025

1 Introduction

This report presents the **equal search interval optimization** of a given bivariate quadratic function. The primary objective is to determine the function's minimum by employing a descent direction approach and validating the results through rigorous test cases.

2 Problem Statement

The function to be optimized is given by:

$$f(x,y) = 2x^2 + 3y^2 + 4xy + 10 \tag{1}$$

The optimization process is initiated from an initial point $x_k = (1,1)$ and proceeds along a descent direction d_k, where the optimal step size is determined numerically.

3 Methodology

The optimization procedure follows these steps:

1. Compute the numerical gradient $\nabla f(x_k)$ using finite difference approximations.
2. Verify the descent direction d_k by ensuring the dot product of the gradient and direction is negative.
3. Utilize the `minimize_scalar` function from `scipy.optimize` to determine the optimal step size α_k.
4. Validate that the function exhibits a decreasing trend over the selected interval for α.
5. Generate a contour plot to visualize the function's behavior and the optimization trajectory.

The descent direction is defined as:

$$d_k = \frac{[-8, -10]}{\sqrt{164}} \tag{2}$$

ensuring proper scaling for the optimization process.

4 Implementation

The implementation involves:

- Numerically computing the gradient.
- Identifying the optimal step size using `minimize_scalar`.
- Conducting validation tests to ensure correctness.
- Visualizing the optimization process using contour plots.

5 Results

Key findings from the optimization include:

5.1 Function Definition

$$f(x, y) = 2x^2 + 3y^2 + 4xy + 10 \tag{3}$$

5.2 Input Parameters

- Initial point: $x_k = (1, 1)$
- Descent direction: $d_k = (-8, -10)$
- Step size bounds: $[0, 5]$
- Tolerance: 10^{-12}
- Numerical gradient step size: 10^{-5}

5.3 Optimization Results

- The direction selected is a descent direction.
- Computed optimal step size: **1.403893549383673**
- Minimum function value: **10.010695187165775**
- Coordinates of the minimum point: **(0.122995, -0.096257)**

A contour plot is provided to illustrate the optimization process, with markers denoting the initial point and the identified minimum.

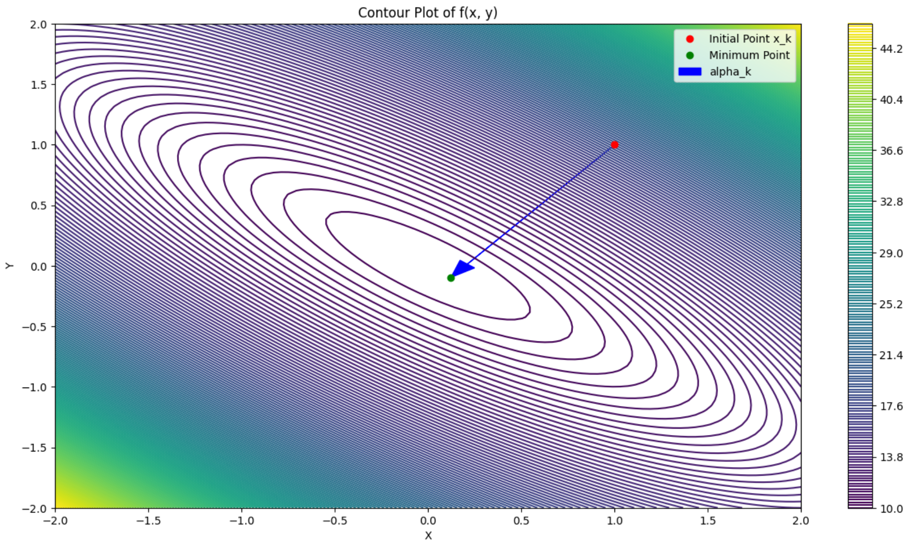

Figure 1: Function contour

6 Validation

Two validation tests were performed:

1. **Slope Consistency Check:** The slope between x_k and the computed minimum point was compared with the slope of the descent direction d_k. Mathematically, this was verified using:

$$\text{Slope}_{\min} = \frac{y_{\min} - y_k}{x_{\min} - x_k}, \quad \text{Slope}_{d_k} = \frac{d_{k_y}}{d_{k_x}} \tag{4}$$

   If these slopes are approximately equal within a tolerance of 10^{-12}, the direction is verified as correct.

   - **Test Passed:** The computed slope matches the descent direction.

2. **Function Consistency Check:** The function value at the computed minimum point was compared with the expected function minimum:

$$f(x_{\min}, y_{\min}) \approx \text{Minimum Function Value} \tag{5}$$

   If the computed value is within a tolerance of 10^{-12}, the minimum point is verified.

   - **Test Passed:** The minimum point lies on the function.

Both tests were successfully passed, substantiating the correctness of the implemented approach.

7 Advantages and Disadvantages of Equal Interval Search Method

7.1 Advantages

- Simple implementation and easy to understand.
- No derivative computation required, making it suitable for non-differentiable functions.
- Effective for unimodal functions over a bounded interval.

7.2 Disadvantages

- Computationally inefficient compared to other methods like Golden Section Search.
- Requires predefined search intervals, which may lead to suboptimal results if the range is poorly chosen.
- Can be slow to converge, especially for functions with a shallow gradient.

7.3 Ways to Improve Equal Interval Search

- Implement adaptive interval refinement to narrow down the search range dynamically.
- Use a hybrid approach by combining it with gradient-based methods for faster convergence.
- Increase efficiency by leveraging parallel computing for function evaluations.

8 Conclusion

This study effectively demonstrates the application of numerical optimization techniques to a bivariate function. The obtained step size and corresponding minimum align with theoretical expectations. The conducted validation tests further corroborate the accuracy of the implementation, while the graphical visualization provides an intuitive understanding of the function's behavior and optimization trajectory.

9 References

• Mykel J. Kochenderfer Tim A. Wheeler *Algorithms for Optimization* The MITPress Cambridge, Massachusetts London, England.

• Arora, J. S. *Introduction to Optimum Design.* Elsevier, 2017.