Detailed Explanation of Numerical Optimization Methods in One Variable

1 Introduction

This document presents a detailed explanation of four numerical methods used for optimizing single-variable functions. Each method is implemented in Python and applied to example problems with visualizations and iteration tables. The following methods are covered:

- Interval Halving Method
- Cubic Interpolation Method
- Gradient Descent with Constant Learning Rate
- Gradient Descent with Variable Learning Rate

2 Interval Halving Method

The interval halving method is a bracketing method used to find a local minimum of a unimodal function over an interval $[a, b]$. The procedure is:

1. Evaluate the midpoint $x_m = \frac{a+b}{2}$.
2. Compute $x_1 = a + \frac{L}{4}$ and $x_2 = b - \frac{L}{4}$, where $L = b - a$.
3. Evaluate $f(x_1)$, $f(x_2)$, and $f(x_m)$.
4. Update the interval depending on the relative magnitudes of these values.

The algorithm continues halving the interval until a desired tolerance is met.

3 Cubic Interpolation Method

This method uses function values and derivatives at two points a and b to fit a cubic polynomial:

$$p(\lambda) = A\lambda^3 + B\lambda^2 + C\lambda + D$$

Given:

- $f(a)$, $f(b)$: function values
- $f'(a)$, $f'(b)$: derivatives at a and b

  We compute:

$$Z = 3\left(\frac{f(a) - f(b)}{b - a}\right) + f'(a) + f'(b)$$

$$Q = \sqrt{Z^2 - f'(a)f'(b)}$$

$$\lambda^* = a + \frac{f'(a) + Z + Q}{f'(a) + f'(b) + 2Z}(b - a)$$

This estimated λ^* is used to update the interval for the next iteration.

4 Gradient Descent Method

Gradient descent is a first-order iterative optimization method for finding a local minimum of a function. For functions of one variable, the gradient reduces to the derivative.

4.1 Constant Learning Rate

Given an initial point x_0, the update rule is:

$$x_{k+1} = x_k - \alpha f'(x_k)$$

where $\alpha > 0$ is the learning rate (step size). The iteration stops when $|f'(x_k)| < \varepsilon$ for a tolerance ε.

4.2 Variable Learning Rate

Instead of using a constant α, a decaying learning rate is used:

$$\alpha_k = \frac{1}{\sqrt{k}}$$

This leads to the update:

$$x_{k+1} = x_k - \frac{1}{\sqrt{k}} f'(x_k)$$

This strategy reduces the step size over time, allowing for more stability near the minimum.

5 Conclusion

These methods illustrate different strategies for univariate optimization. Bracketing methods like interval halving ensure convergence in unimodal functions, while model-based methods like cubic interpolation use derivative information for efficiency. Gradient descent methods are simple and powerful, especially when derivative information is readily available.

Each method was validated with Python implementations, including iteration tables and plots to enhance understanding.