Newton Raphson Method

Swapnil Singh
Bank of Lithuania, KTU
November 25, 2024

Newton-Raphson Method: Overview

- **Purpose:** Find roots of equation $f(x) = 0$ or minimum/maximum of function
- **Key Idea:** Use local quadratic approximation
- **Advantages:**
  - Quadratic convergence when close to solution
  - Efficient for smooth functions
  - Works well for multi-dimensional problems
- **Disadvantages:**
  - Requires derivatives
  - Sensitive to starting point
  - May not converge for poor initial guesses

Mathematical Foundation

For Root Finding:

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

For Optimization:

- Find where gradient is zero: $\nabla f(x) = 0$
- Update formula:

$$x_{k+1} = x_k - [H(x_k)]^{-1} \nabla f(x_k)$$

where:

- $\nabla f(x_k)$ is the gradient
- $H(x_k)$ is the Hessian matrix

Convergence Properties

Quadratic Convergence:

- Error reduces quadratically:

$$\|e_{k+1}\| \leq C\|e_k\|^2$$

- Requires:
  - Smooth function
  - Good initial guess
  - Non-singular Hessian at solution

Conditions for Minimum:

- First-order: $\nabla f(x^*) = 0$
- Second-order: $H(x^*)$ is positive definite

Algorithm Implementation

[H] **Input:** Initial guess x_0, tolerance ϵ, max iterations N

Output: Optimal point x^*

$k \leftarrow 0$ **while** $k < N$ **do**
Compute gradient: $g_k = \nabla f(x_k)$ Compute Hessian: $H_k = H(x_k)$ **if** $\|g_k\| < \epsilon$ **then**
return x_k

Solve: $H_k d_k = -g_k$ Update: $x_{k+1} = x_k + d_k$ $k \leftarrow k + 1$

return x_k

MATLAB Implementation Tips

Key Components:

- **Function Evaluation:**
  - Define objective function
  - Compute gradient analytically or numerically
  - Compute Hessian analytically or numerically
- **Linear System Solution:**
  - Use `mldivide` (backslash) operator
  - Check condition number
  - Handle singular matrices
- **Convergence Check:**
  - Gradient norm
  - Step size
  - Function value change

Practical Considerations

Implementation Challenges:

- Choose appropriate stopping criteria
- Handle ill-conditioned Hessians
- Select good initial point
- Deal with non-convex functions

Improvements:

- Line search for step size
- Trust region methods
- Quasi-Newton methods (BFGS)
- Regularization for ill-conditioning