Numerical optimization : theory and applications

Ammar Mian
Associate professor, LISTIC, Université Savoie Mont Blanc

Outline

- Newton Method
- Quasi-Newton Methods
- BFGS Method
- SR1 Method
- Convergence Theory
- Exercises

Newton Method - Motivation

Key Insight

- Steepest descent: navigating with only immediate slope
- Newton method: having detailed topographic map
- Incorporates curvature information (how slope changes)
- Uses second-order Taylor approximation

Strategy

Instead of minimizing f directly, minimize simpler quadratic approximation:

$$f(\mathbf{x}_k + \mathbf{p}) \approx f(\mathbf{x}_k) + \nabla f(\mathbf{x}_k)^T \mathbf{p} + \frac{1}{2}\mathbf{p}^T \nabla^2 f(\mathbf{x}_k)\mathbf{p}$$

Newton Method - Algorithm

Derivation

Setting gradient of quadratic approximation to zero:

$$\nabla f(\mathbf{x}_k) + \nabla^2 f(\mathbf{x}_k)\mathbf{p} = \mathbf{0}$$

Solving for Newton step:

$$\mathbf{p}_k^N = -[\nabla^2 f(\mathbf{x}_k)]^{-1} \nabla f(\mathbf{x}_k)$$

Newton Iteration

$$\mathbf{x}_{k+1} = \mathbf{x}_k - [\nabla^2 f(\mathbf{x}_k)]^{-1} \nabla f(\mathbf{x}_k)$$

Newton Method - Properties

Advantages

- Recognizes elongated valley shapes via Hessian
- Takes larger steps along valley floor, smaller steps perpendicular
- Eliminates zigzag behavior of steepest descent
- Natural step size of $\alpha = 1$
- Quadratic convergence rate

Special Property

For quadratic functions: Newton method finds exact minimum in single step, regardless of conditioning!

Newton Method - Challenges

Main Drawbacks

- Requires computation of Hessian matrix $\nabla^2 f(\mathbf{x})$
- Need to solve linear system at each iteration
- Hessian may not be positive definite away from solution
- Expensive: $O(n^3)$ operations per iteration

When Newton Fails

When $\nabla^2 f_k$ is not positive definite:

- Newton direction may not be defined
- May not satisfy descent condition $\nabla f_k^T \mathbf{p}_k^N < 0$

Quasi-Newton Methods - Motivation

Core Idea

- Avoid computing exact Hessian $\nabla^2 f_k$
- Use approximation $\mathbf{B}_k \approx \nabla^2 f_k$
- Update approximation using gradient information
- Achieve superlinear convergence without Hessian computation

Quasi-Newton Direction

$$\mathbf{p}_k = -\mathbf{B}_k^{-1} \nabla f_k$$

where $\mathbf{B}_k$ is updated after each step.

The Secant Equation

Key Requirement

We want $\mathbf{B}_{k+1}$ to satisfy:

$$\mathbf{B}_{k+1}\mathbf{s}_k = \mathbf{y}_k$$

where:

- $\mathbf{s}_k = \mathbf{x}_{k+1} - \mathbf{x}_k$ (displacement)
- $\mathbf{y}_k = \nabla f_{k+1} - \nabla f_k$ (gradient change)

Curvature Condition

For positive definite updates, we need:

$$\mathbf{s}_k^T\mathbf{y}_k > 0$$

This is guaranteed by Wolfe line search conditions.

BFGS Method

Most Popular Quasi-Newton Method

Named after Broyden, Fletcher, Goldfarb, and Shanno.

BFGS Update Formula

$$\mathbf{H}_{k+1} = \left(\mathbf{I} - \rho_k \mathbf{s}_k \mathbf{y}_k^T\right) \mathbf{H}_k \left(\mathbf{I} - \rho_k \mathbf{y}_k \mathbf{s}_k^T\right) + \rho_k \mathbf{s}_k \mathbf{s}_k^T$$

where:

- $\mathbf{H}_k = \mathbf{B}_k^{-1}$ (inverse Hessian approximation)
- $\rho_k = \frac{1}{\mathbf{y}_k^T \mathbf{s}_k}$

BFGS Algorithm

Algorithm Steps

1. Choose initial $\mathbf{x}_0$ and $\mathbf{H}_0$ (often $\mathbf{H}_0 = \mathbf{I}$)
2. While $\|\nabla f_k\| > \epsilon$:
    - Compute search direction: $\mathbf{p}_k = -\mathbf{H}_k \nabla f_k$
    - Line search: find α_k satisfying Wolfe conditions
    - Update: $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k$
    - Compute: $\mathbf{s}_k = \mathbf{x}_{k+1} - \mathbf{x}_k$, $\mathbf{y}_k = \nabla f_{k+1} - \nabla f_k$
    - Update $\mathbf{H}_{k+1}$ using BFGS formula

BFGS Properties

Key Advantages

- Only $O(n^2)$ operations per iteration
- Superlinear convergence rate
- Maintains positive definiteness automatically
- Self-correcting: bad approximations get corrected
- No second derivatives required

Convergence Comparison

Method	Steepest Descent	BFGS
Iterations	5264	34
Convergence	Linear	Superlinear

Example on Rosenbrock function from $(-1.2, 1)$.

Symmetric Rank-1 (SR1) Method

Rank-1 Update

$$\mathbf{B}_{k+1} = \mathbf{B}_k + \frac{(\mathbf{y}_k - \mathbf{B}_k\mathbf{s}_k)(\mathbf{y}_k - \mathbf{B}_k\mathbf{s}_k)^T}{(\mathbf{y}_k - \mathbf{B}_k\mathbf{s}_k)^T\mathbf{s}_k}$$

Key Differences from BFGS

- Rank-1 update (vs. rank-2 for BFGS)
- Does not maintain positive definiteness
- Can handle indefinite Hessians
- Often produces better Hessian approximations

SR1 Implementation Issues

Potential Problems

- Denominator can vanish: $(\mathbf{y}_k - \mathbf{B}_k\mathbf{s}_k)^T\mathbf{s}_k = 0$
- No symmetric rank-1 update may exist
- Numerical instabilities possible

Safeguard Strategy

Skip update when:

$$|\mathbf{s}_k^T(\mathbf{y}_k - \mathbf{B}_k\mathbf{s}_k)| < r\|\mathbf{s}_k\|\|\mathbf{y}_k - \mathbf{B}_k\mathbf{s}_k\|$$

where $r \approx 10^{-8}$ is small tolerance.

SR1 - Finite Termination Property

Remarkable Property

For quadratic functions, SR1 method:

- Converges to minimizer in at most n steps
- Satisfies secant equation for **all** previous directions
- Recovers exact Hessian: $\mathbf{H}_n = A^{-1}$ after n steps

Advantage over BFGS

This property holds regardless of line search accuracy, while BFGS requires exact line search for similar guarantees.

Global Convergence

Zoutendijk's Condition

For line search methods satisfying Wolfe conditions:

$$\sum_{k=0}^{\infty} \cos^2\theta_k \|\nabla f_k\|^2 < \infty$$

where θ_k is angle between search direction and negative gradient.

Newton-like Methods

If $\mathbf{p}_k = -\mathbf{B}_k^{-1}\nabla f_k$ with bounded condition number:

$$\|\mathbf{B}_k\|\|\mathbf{B}_k^{-1}\| \leq M$$

Then: $\cos\theta_k \geq 1/M$ and $\lim_{k\to\infty} \|\nabla f_k\| = 0$.

Rate of Convergence

Convergence Rates

- **Steepest Descent:** Linear convergence
- **Newton:** Quadratic convergence (near solution)
- **Quasi-Newton:** Superlinear convergence

Practical Performance

- Newton: Fastest per iteration, but expensive
- BFGS: Good balance of speed and cost
- Steepest Descent: Slow but simple and robust

Implementation Considerations

Step Size Strategy

- Always try $\alpha = 1$ first (Newton step)
- Use Wolfe conditions for line search
- BFGS: accept $\alpha = 1$ eventually for superlinear convergence

Initial Hessian Approximation

Common choices for $\mathbf{H}_0$:

- Identity matrix: $\mathbf{H}_0 = \mathbf{I}$
- Scaled identity: $\mathbf{H}_0 = \beta \mathbf{I}$
- After first step: $\mathbf{H}_0 = \frac{\mathbf{y}_0^T \mathbf{s}_0}{\mathbf{y}_0^T \mathbf{y}_0} \mathbf{I}$

Summary

Method Comparison

Method	Cost/Iter	Convergence	Hessian
Steepest Descent	$O(n)$	Linear	Not needed
Newton	$O(n^3)$	Quadratic	Required
BFGS	$O(n^2)$	Superlinear	Approximated
SR1	$O(n^2)$	Superlinear	Approximated

Practical Recommendation

BFGS is the most widely used method due to its excellent balance of:

- Fast convergence (superlinear)
- Moderate computational cost
- Robust performance
- No second derivatives required

Exercise 1: Himmelblau Function

Problem Statement

Implement BFGS and SR1 methods to minimize the Himmelblau function: $f(x_1, x_2) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2$

Tasks

1. Compute the gradient $\nabla f(x_1, x_2)$ analytically
2. Implement both BFGS and SR1 algorithms with Wolfe line search
3. Test from starting points: $(0, 0)$, $(1, 1)$, $(-1, 1)$, $(4, 4)$
4. Compare convergence behavior, number of iterations, and final solutions
5. Plot convergence trajectories on contour plot

Exercise 2: Mixed Function

Problem Statement

Implement BFGS and SR1 methods to minimize: $f(x_1, x_2) = \frac{1}{2}x_1^2 + x_1 \cos(x_2)$

Tasks

1. Derive the gradient $\nabla f(x_1, x_2)$ and Hessian $\nabla^2 f(x_1, x_2)$
2. Implement BFGS, SR1, and exact Newton method
3. Use starting points: $(1, 0)$, $(2, \pi)$, $(-1, \pi/2)$
4. Compare all three methods in terms of:
   - Convergence speed
   - Final solutions found
   - Robustness to different starting points