Optimization Module – Structured Notes

# 1. Ingredients of an Optimization Problem

• Objective function  
• Variables  
• Constraints  
  
Goal: Find variable values that minimize or maximize the objective function while satisfying constraints.

# 2. Different Optimization Techniques

• Algorithms vary widely based on problem type:  
 – Closed form vs. numerical vs. discrete  
 – Local vs. global minima  
 – Running times from O(1) to NP-hard  
• Today’s focus: Continuous numerical methods

# 3. 1-D Optimization – Bracketing a Minimum

• Bracketing is similar to root-finding  
• A bracket is formed when f(x\_mid) < f(x\_left) and f(x\_mid) < f(x\_right)  
• To establish a bracket:  
 – Start from x\_initial and increment  
 – Evaluate function  
 – Step in the direction until an increase is observed  
 – Reverse if necessary  
• For maximization: use -f(x)

# 4. Golden Section Search

• Use α = (√5 – 1)/2 ≈ 0.618 (Golden Ratio)  
• Ensures interval shrinks uniformly by ~30%  
• Converges linearly

# 5. Error Tolerance in Optimization

• Near a minimum, derivative ≈ 0  
• Use Taylor expansion to approximate function value  
• Rule: Don’t seek accuracy beyond √(machine ε)

# 6. Newton’s Method

• Update rule: x\_{k+1} = x\_k - f'(x\_k) / f''(x\_k)  
• Requires: First and second derivatives  
• Converges quadratically if close to solution

# 7. Multidimensional Optimization

• Replace derivatives with gradient and Hessian matrix  
• Update: x\_{k+1} = x\_k - H⁻¹ ∇f(x\_k)  
• Fragile if not smooth or well-initialized

# 8. Classification of Optimization Methods

• Function + Gradient + Hessian → Newton’s method  
• Function + Gradient → Gradient Descent  
• Function values only → Nelder-Mead (Simplex/Amoeba)

# 9. Steepest Descent & Quasi-Newton

• When Hessian not available, walk in direction of -∇f  
• Perform line search  
• Recompute gradient and iterate

# 10. Conjugate Gradient Methods

• Avoid repeating directions  
• Use Polak-Ribiere formula:  
 β\_k = (g\_{k+1}ᵀ(g\_{k+1} - g\_k)) / (g\_kᵀ g\_k)  
• Works well for quadratic functions

# 11. Value-Only Optimization

• Estimate gradients numerically using finite differences  
• Slower and less stable but sometimes necessary

# 12. Generic Strategies & Simulated Annealing

• Uniform Sampling: Exponential cost in high dimensions  
• Simulated Annealing:  
 – Start with large random steps  
 – Gradually reduce step size  
 – Use annealing schedule

# 13. Nelder-Mead (Downhill Simplex Method)

• Uses n+1 points in n-D space  
• Operations: Reflection, Contraction, Expansion  
• Simple, derivative-free, but may require restart

# 14. Rosenbrock’s Function

• Test function for optimization methods  
• Defined as f(x, y) = 100(y - x²)² + (1 - x)²

# 15. Constrained Optimization

• Equality Constraints → Use Lagrange Multipliers:  
 Minimize f(x) + λg(x)  
• Inequality Constraints → Use Linear Programming  
 – Simplex Method: move between vertices