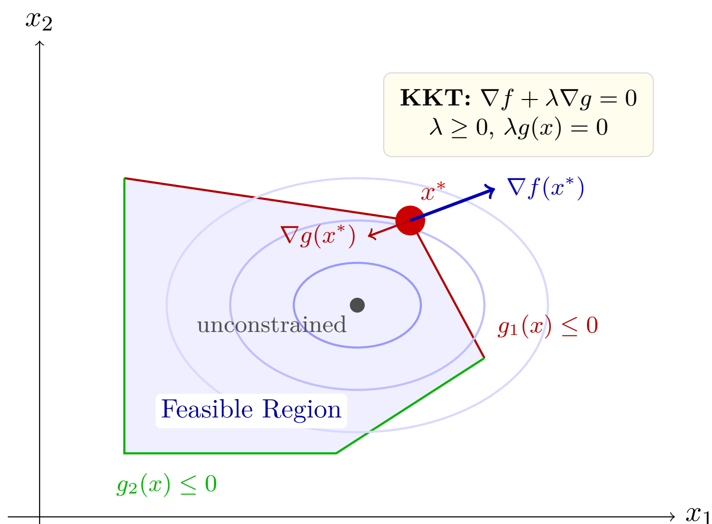
Supplementary of Mathematical Modeling

Constrained Optimization

Theory and Applications



$\min f(x) \quad \text{subject to} \quad g_i(x) \leq 0$

Kenneth, Sok Kin Cheng

Last update: July 8, 2025

Mastering Optimization with Constraints and Boundaries

Contents

Chapter 1

Foundations of Constrained Optimization

1.1 Introduction

Constrained optimization addresses the fundamental problem of finding optimal solutions when decision variables must satisfy specific restrictions. Unlike unconstrained optimization, the presence of constraints fundamentally changes the nature of optimality conditions and solution methods.

Definition 1.1 (Constrained Optimization Problem)**.** The general constrained optimization problem is formulated as:

$$\min_{x \in \mathbb{R}^n} \quad f(x) \tag{1.1}$$

$$\text{subject to} \quad g_i(x) \leq 0, \quad i = 1, 2, \ldots, m \tag{1.2}$$

$$h_j(x) = 0, \quad j = 1, 2, \ldots, p \tag{1.3}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is the objective function, $g_i : \mathbb{R}^n \to \mathbb{R}$ are inequality constraints, and $h_j : \mathbb{R}^n \to \mathbb{R}$ are equality constraints.

1.2 Feasible Sets and Constraint Qualification

Definition 1.2 (Feasible Set)**.** The feasible set $\mathcal{S}$ is defined as:

$$\mathcal{S} = \{x \in \mathbb{R}^n : g_i(x) \leq 0 \text{ for all } i, \text{ and } h_j(x) = 0 \text{ for all } j\}$$

Definition 1.3 (Active Set)**.** At a point x^*, the active set $\mathcal{A}(x^*)$ consists of:

$$\mathcal{A}(x^*) = \{i : g_i(x^*) = 0\} \cup \{j : h_j(x^*) = 0\}$$

Definition 1.4 (Linear Independence Constraint Qualification (LICQ))**.** At a feasible point x^*, LICQ holds if the gradients $\{\nabla g_i(x^*) : i \in \mathcal{A}(x^*)\}$ and $\{\nabla h_j(x^*) : j = 1, \ldots, p\}$ are linearly independent.

1.3 Lagrange Multipliers and KKT Conditions

The method of Lagrange multipliers extends the concept of stationary points to constrained problems.

Definition 1.5 (Lagrangian Function)**.** The Lagrangian function is defined as:

$$\mathcal{L}(x, \lambda, \mu) = f(x) + \sum_{i=1}^{m} \lambda_i g_i(x) + \sum_{j=1}^{p} \mu_j h_j(x)$$

where $\lambda_i \geq 0$ are multipliers for inequality constraints and μ_j are multipliers for equality constraints.

Theorem

Karush-Kuhn-Tucker (KKT) Conditions
Let x^* be a local minimum of the constrained optimization problem, and assume LICQ holds at x^*. Then there exist multipliers $\lambda^* \in \mathbb{R}^m$ and $\mu^* \in \mathbb{R}^p$ such that:
Stationarity: $\nabla_x \mathcal{L}(x^*, \lambda^*, \mu^*) = 0$
Primal feasibility: $g_i(x^*) \leq 0$, $h_j(x^*) = 0$
Dual feasibility: $\lambda_i^* \geq 0$
Complementary slackness: $\lambda_i^* g_i(x^*) = 0$

1.3.1 Applications and Examples

Example 1.1 (Classical Lagrange Multipliers - Equality Constraints)**.** Consider the problem of minimizing the distance from the origin to a line:

$$\min \quad f(x_1, x_2) = x_1^2 + x_2^2 \tag{1.4}$$
$$\text{s.t.} \quad h(x_1, x_2) = x_1 + 2x_2 - 3 = 0 \tag{1.5}$$

Solution Process:
Step 1: Form the Lagrangian:

$$\mathcal{L}(x_1, x_2, \mu) = x_1^2 + x_2^2 + \mu(x_1 + 2x_2 - 3)$$

Step 2: Apply first-order necessary conditions:

$$\frac{\partial \mathcal{L}}{\partial x_1} = 2x_1 + \mu = 0 \quad \Rightarrow \quad x_1 = -\frac{\mu}{2} \tag{1.6}$$
$$\frac{\partial \mathcal{L}}{\partial x_2} = 2x_2 + 2\mu = 0 \quad \Rightarrow \quad x_2 = -\mu \tag{1.7}$$
$$\frac{\partial \mathcal{L}}{\partial \mu} = x_1 + 2x_2 - 3 = 0 \tag{1.8}$$

Step 3: Solve the system: Substituting into the constraint: $-\frac{\mu}{2} + 2(-\mu) - 3 = 0$

$$-\frac{\mu}{2} - 2\mu = 3 \quad \Rightarrow \quad -\frac{5\mu}{2} = 3 \quad \Rightarrow \quad \mu = -\frac{6}{5}$$

Step 4: Find the optimal point:

$$x_1^* = -\frac{\mu}{2} = \frac{3}{5}, \quad x_2^* = -\mu = \frac{6}{5}$$

Verification: $\frac{3}{5} + 2 \cdot \frac{6}{5} = \frac{15}{5} = 3$
Interpretation: The Lagrange multiplier $\mu = -\frac{6}{5}$ represents the rate of change of the optimal objective value with respect to the constraint right-hand side.

Example 1.2 (KKT Conditions with Inequality Constraints)**.** Solve the constrained quadratic programming problem:

$$\min \quad f(x_1, x_2) = x_1^2 + x_2^2 - 2x_1 - 4x_2 \tag{1.9}$$
$$\text{s.t.} \quad g_1(x_1, x_2) = x_1 + x_2 - 2 \leq 0 \tag{1.10}$$
$$g_2(x_1, x_2) = -x_1 \leq 0 \tag{1.11}$$
$$g_3(x_1, x_2) = -x_2 \leq 0 \tag{1.12}$$

Solution Process:

Step 1: Form the Lagrangian:

$$\mathcal{L}(x, \lambda) = x_1^2 + x_2^2 - 2x_1 - 4x_2 + \lambda_1(x_1 + x_2 - 2) + \lambda_2(-x_1) + \lambda_3(-x_2)$$

Step 2: Write the KKT conditions:

$$\text{Stationarity:} \quad \begin{cases} \frac{\partial \mathcal{L}}{\partial x_1} = 2x_1 - 2 + \lambda_1 - \lambda_2 = 0 \\ \frac{\partial \mathcal{L}}{\partial x_2} = 2x_2 - 4 + \lambda_1 - \lambda_3 = 0 \end{cases} \tag{1.13}$$
$$\text{Primal feasibility:} \quad x_1 + x_2 \leq 2, \quad x_1 \geq 0, \quad x_2 \geq 0 \tag{1.14}$$
$$\text{Dual feasibility:} \quad \lambda_i \geq 0, \quad i = 1, 2, 3 \tag{1.15}$$
$$\text{Complementary slackness:} \quad \begin{cases} \lambda_1(x_1 + x_2 - 2) = 0 \\ \lambda_2 x_1 = 0 \\ \lambda_3 x_2 = 0 \end{cases} \tag{1.16}$$

Step 3: Case analysis:

Case 1: Check unconstrained solution ($\lambda_1 = \lambda_2 = \lambda_3 = 0$) From stationarity: $x_1 = 1$, $x_2 = 2$ Check feasibility: $1 + 2 = 3 > 2 \quad$ (violates constraint 1)

Case 2: Constraint 1 active ($\lambda_1 > 0$, $x_1 + x_2 = 2$) Assume $\lambda_2 = \lambda_3 = 0$ (interior w.r.t. bounds):

$$2x_1 - 2 + \lambda_1 = 0 \quad \Rightarrow \quad \lambda_1 = 2 - 2x_1 \tag{1.17}$$
$$2x_2 - 4 + \lambda_1 = 0 \quad \Rightarrow \quad \lambda_1 = 4 - 2x_2 \tag{1.18}$$

Setting equal: $2 - 2x_1 = 4 - 2x_2$ From constraint: $x_2 = 2 - x_1$ Substituting: $2 - 2x_1 = 4 - 2(2 - x_1) = 2x_1$ Solving: $2 = 4x_1 \Rightarrow x_1 = \frac{1}{2}$, $x_2 = \frac{3}{2}$

Check: $\lambda_1 = 2 - 2 \cdot \frac{1}{2} = 1 > 0$

Step 4: Verify all KKT conditions:

- Stationarity: $2 \cdot \frac{1}{2} - 2 + 1 = 0$, $2 \cdot \frac{3}{2} - 4 + 1 = 0$
- Primal feasibility: $\frac{1}{2} + \frac{3}{2} = 2$, $x_1, x_2 \geq 0$
- Dual feasibility: $\lambda = (1, 0, 0) \geq 0$
- Complementary slackness: $1 \cdot 0 = 0$, $0 \cdot \frac{1}{2} = 0$, $0 \cdot \frac{3}{2} = 0$

Optimal solution: $x^* = \left(\frac{1}{2}, \frac{3}{2}\right)$ with multipliers $\lambda^* = (1, 0, 0)$

Example 1.3 (Portfolio Optimization with Mixed Constraints)**.** Consider a simplified portfolio optimization problem:

$$\min \quad \frac{1}{2}(w_1^2 + w_2^2) \quad \text{(minimize variance/risk)} \tag{1.19}$$

$$\text{s.t.} \quad 0.08w_1 + 0.12w_2 \geq 0.10 \quad \text{(minimum expected return)} \tag{1.20}$$

$$w_1 + w_2 = 1 \quad \text{(budget constraint)} \tag{1.21}$$

$$w_1, w_2 \geq 0 \quad \text{(no short selling)} \tag{1.22}$$

Solution Process:

Step 1: Convert to standard form:

$$\min \quad \frac{1}{2}(w_1^2 + w_2^2) \tag{1.23}$$

$$\text{s.t.} \quad g_1(w) = -0.08w_1 - 0.12w_2 + 0.10 \leq 0 \tag{1.24}$$

$$h_1(w) = w_1 + w_2 - 1 = 0 \tag{1.25}$$

$$g_2(w) = -w_1 \leq 0, \quad g_3(w) = -w_2 \leq 0 \tag{1.26}$$

Step 2: Form the Lagrangian:

$$\mathcal{L} = \frac{1}{2}(w_1^2 + w_2^2) + \lambda_1(-0.08w_1 - 0.12w_2 + 0.10) + \mu(w_1 + w_2 - 1) + \lambda_2(-w_1) + \lambda_3(-w_2)$$

Step 3: KKT conditions:

$$\text{Stationarity:} \quad \begin{cases} w_1 - 0.08\lambda_1 + \mu - \lambda_2 = 0 \\ w_2 - 0.12\lambda_1 + \mu - \lambda_3 = 0 \end{cases} \tag{1.27}$$

$$\text{Constraints:} \quad w_1 + w_2 = 1, \quad 0.08w_1 + 0.12w_2 \geq 0.10, \quad w_1, w_2 \geq 0 \tag{1.28}$$

Step 4: Case analysis:

Assume interior solution ($\lambda_2 = \lambda_3 = 0$) and return constraint active ($\lambda_1 > 0$):

$$w_1 = 0.08\lambda_1 - \mu \tag{1.29}$$

$$w_2 = 0.12\lambda_1 - \mu \tag{1.30}$$

From budget constraint: $w_1 + w_2 = 0.20\lambda_1 - 2\mu = 1$ From return constraint: $0.08w_1 + 0.12w_2 = 0.0208\lambda_1 - 0.20\mu = 0.10$

Solving the system:

$$\begin{bmatrix} 0.20 & -2 \\ 0.0208 & -0.20 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \mu \end{bmatrix} = \begin{bmatrix} 1 \\ 0.10 \end{bmatrix}$$

Using Cramer's rule: $\det = 0.20 \cdot (-0.20) - (-2) \cdot 0.0208 = -0.04 + 0.0416 = 0.0016$

$\lambda_1 = \frac{\begin{vmatrix} 1 & -2 \\ 0.10 & -0.20 \end{vmatrix}}{0.0016} = \frac{-0.20+0.20}{0.0016} = 0$

Since $\lambda_1 = 0$, the return constraint is not active. Check if unconstrained optimum satisfies return constraint:

With $\lambda_1 = 0$: $w_1 = w_2 = \mu$ From budget: $2\mu = 1 \Rightarrow \mu = 0.5$ So $w_1 = w_2 = 0.5$

Check return: $0.08 \cdot 0.5 + 0.12 \cdot 0.5 = 0.10$

Optimal portfolio: $w^* = (0.5, 0.5)$ - equal weights in both assets.

Economic interpretation: When the minimum return constraint is exactly met by the minimum-variance portfolio, no trade-off between risk and return is needed.

Python Code

```
import numpy as np
from scipy.optimize import minimize
import matplotlib.pyplot as plt

def solve_kkt_examples():
    """
    Solve the KKT examples numerically for verification
    """

    # Example 1: Equality constraint problem
    def obj1(x):
        return x[0]**2 + x[1]**2

    def constraint1(x):
        return x[0] + 2*x[1] - 3

    constraints1 = [{'type': 'eq', 'fun': constraint1}]
    x0_1 = np.array([1.0, 1.0])

    result1 = minimize(obj1, x0_1, method='SLSQP', constraints=constraints1)

    print("Example 1 - Equality Constraint:")
    print(f"Numerical solution: x = {result1.x}")
    print(f"Analytical solution: x = [0.6, 1.2]")
    print(f"Objective value: {result1.fun}")
    print(f"Constraint violation: {abs(constraint1(result1.x))}")
    print()

    # Example 2: Inequality constraints
    def obj2(x):
        return x[0]**2 + x[1]**2 - 2*x[0] - 4*x[1]

    def grad2(x):
        return np.array([2*x[0] - 2, 2*x[1] - 4])

    constraints2 = [
        {'type': 'ineq', 'fun': lambda x: 2 - x[0] - x[1]},  # x1 + x2 <= 2
        {'type': 'ineq', 'fun': lambda x: x[0]},              # x1 >= 0
        {'type': 'ineq', 'fun': lambda x: x[1]}               # x2 >= 0
    ]

    x0_2 = np.array([0.5, 0.5])
    result2 = minimize(obj2, x0_2, method='SLSQP', jac=grad2,
                      constraints=constraints2)

    print("Example 2 - Inequality Constraints:")
    print(f"Numerical solution: x = {result2.x}")
    print(f"Analytical solution: x = [0.5, 1.5]")
    print(f"Objective value: {result2.fun}")

    # Check constraint activity
```

```
    x_opt = result2.x
    g1 = x_opt[0] + x_opt[1] - 2
    print(f"Constraint 1 (x1+x2-2): {g1:.6f} {'(active)' if abs(g1) < 1e-6 else
    ↪ '(inactive)'}")
    print()

    # Example 3: Portfolio optimization
    def portfolio_obj(w):
        return 0.5 * (w[0]**2 + w[1]**2)

    constraints3 = [
        {'type': 'ineq', 'fun': lambda w: 0.08*w[0] + 0.12*w[1] - 0.10},
        {'type': 'eq', 'fun': lambda w: w[0] + w[1] - 1},
        {'type': 'ineq', 'fun': lambda w: w[0]},
        {'type': 'ineq', 'fun': lambda w: w[1]}
    ]

    w0 = np.array([0.5, 0.5])
    result3 = minimize(portfolio_obj, w0, method='SLSQP', constraints=constraints3)

    print("Example 3 - Portfolio Optimization:")
    print(f"Optimal weights: w = {result3.x}")
    print(f"Portfolio variance: {result3.fun:.6f}")

    # Calculate expected return
    expected_return = 0.08 * result3.x[0] + 0.12 * result3.x[1]
    print(f"Expected return: {expected_return:.4f} = {expected_return*100:.1f}%")

    return result1, result2, result3

# Visualization function
def plot_constrained_optimization():
    """
    Visualize the second example with contours and constraints
    """
    x1 = np.linspace(-0.5, 3, 100)
    x2 = np.linspace(-0.5, 3, 100)
    X1, X2 = np.meshgrid(x1, x2)

    # Objective function
    F = X1**2 + X2**2 - 2*X1 - 4*X2

    plt.figure(figsize=(10, 8))

    # Contour lines
    contours = plt.contour(X1, X2, F, levels=15, alpha=0.7, colors='blue')
    plt.clabel(contours, inline=True, fontsize=8)

    # Constraint boundaries
    x1_line = np.linspace(0, 2.5, 100)
    x2_line = 2 - x1_line

    # Feasible region
```

```python
    x1_fill = np.linspace(0, 2, 100)
    x2_upper = 2 - x1_fill
    x2_lower = np.zeros_like(x1_fill)

    plt.fill_between(x1_fill, x2_lower, x2_upper, alpha=0.3,
                     color='lightgreen', label='Feasible Region')

    # Boundaries
    plt.plot(x1_line, x2_line, 'r-', linewidth=2, label='$x_1 + x_2 = 2$')
    plt.axhline(y=0, color='black', linewidth=1.5, label='$x_2 \\geq 0$')
    plt.axvline(x=0, color='black', linewidth=1.5, label='$x_1 \\geq 0$')

    # Optimal points
    plt.plot(0.5, 1.5, 'ro', markersize=12, label='Constrained Optimum')
    plt.plot(1, 2, 'bs', markersize=10, label='Unconstrained Optimum')

    # Add arrows showing gradients at optimum
    plt.arrow(0.5, 1.5, 0.3, 0.6, head_width=0.1, head_length=0.1,
              fc='red', ec='red', alpha=0.7)
    plt.text(0.9, 2.2, '$\\nabla f$', fontsize=12, color='red')

    plt.xlim(-0.2, 2.8)
    plt.ylim(-0.2, 2.8)
    plt.xlabel('$x_1$', fontsize=14)
    plt.ylabel('$x_2$', fontsize=14)
    plt.title('KKT Example: Constrained Quadratic Optimization', fontsize=16)
    plt.legend(fontsize=12)
    plt.grid(True, alpha=0.3)
    plt.show()

# Execute examples
results = solve_kkt_examples()
plot_constrained_optimization()
```

Exploration

Advanced KKT Analysis

Explore these concepts to deepen your understanding:

1. **Constraint Qualification Violations**: Construct examples where LICQ fails and observe how KKT conditions may not provide necessary conditions for optimality.

2. **Multiple Optimal Solutions**: Find problems where KKT conditions are satisfied at multiple points, investigating when this occurs.

3. **Economic Interpretation**: In the portfolio example, interpret the Lagrange multipliers as shadow prices and analyze their sensitivity to constraint changes.

4. **Active Set Changes**: Study how the active set changes as problem parameters vary, particularly in the transition between different constraint combinations.

Exercise 1.1. Solve the following constrained optimization problem using KKT conditions:

$$\min \quad x_1^2 + x_2^2 + x_3^2 \tag{1.31}$$
$$\text{s.t.} \quad x_1 + x_2 + x_3 = 3 \tag{1.32}$$
$$x_1^2 + x_2^2 \leq 2 \tag{1.33}$$
$$x_1, x_2, x_3 \geq 0 \tag{1.34}$$

Identify all possible active sets and determine which one corresponds to the optimal solution.

Exercise 1.2. For the general quadratic programming problem:

$$\min \quad \frac{1}{2}x^T Q x + c^T x \tag{1.35}$$
$$\text{s.t.} \quad Ax \leq b \tag{1.36}$$
$$Ex = d \tag{1.37}$$
$$x \geq 0 \tag{1.38}$$

where $Q \succ 0$, derive the complete KKT system and show how it can be written as a linear complementarity problem.

1.4 Geometric Interpretation

The KKT conditions have a clear geometric interpretation: at the optimum, the gradient of the objective function must be a linear combination of the gradients of the active constraints.

1.4.1 Fundamental Geometric Principle

Theorem

Geometric Interpretation of KKT Conditions
At a constrained optimum x^*, the gradient of the objective function $\nabla f(x^*)$ lies in the cone generated by the gradients of the active constraints. Specifically:

$$\nabla f(x^*) = -\sum_{i \in \mathcal{A}(x^*)} \lambda_i \nabla g_i(x^*) - \sum_{j=1}^{p} \mu_j \nabla h_j(x^*)$$

where $\mathcal{A}(x^*)$ is the active set and all $\lambda_i \geq 0$.

This geometric condition ensures that we cannot find a feasible direction that decreases the objective function, which is precisely the optimality condition.

Example 1.4 (Simple Constrained Problem)**.** Consider the problem:

$$\min \quad f(x) = x_1^2 + x_2^2 \tag{1.39}$$
$$\text{s.t.} \quad x_1 + x_2 - 1 = 0 \tag{1.40}$$

The Lagrangian is: $\mathcal{L}(x, \mu) = x_1^2 + x_2^2 + \mu(x_1 + x_2 - 1)$

KKT conditions give:

$$\frac{\partial \mathcal{L}}{\partial x_1} = 2x_1 + \mu = 0 \tag{1.41}$$
$$\frac{\partial \mathcal{L}}{\partial x_2} = 2x_2 + \mu = 0 \tag{1.42}$$
$$\frac{\partial \mathcal{L}}{\partial \mu} = x_1 + x_2 - 1 = 0 \tag{1.43}$$

Solving: $x_1 = x_2 = \frac{1}{2}$, $\mu = -1$.

Geometric Verification: - $\nabla f(\frac{1}{2}, \frac{1}{2}) = (1, 1)$ - $\nabla h(\frac{1}{2}, \frac{1}{2}) = (1, 1)$ - Indeed: $\nabla f = -\mu \nabla h = -(-1)(1, 1) = (1, 1)$

1.4.2 Case Studies: Different Geometric Scenarios

Example 1.5 (Inequality Constraint Becomes Active)**.** Consider the problem:

$$\min \quad f(x) = (x_1 - 2)^2 + (x_2 - 2)^2 \tag{1.44}$$
$$\text{s.t.} \quad g(x) = x_1 + x_2 - 2 \leq 0 \tag{1.45}$$

Geometric Analysis:

Case 1: Unconstrained optimum at $(2, 2)$ Check constraint: $2 + 2 - 2 = 2 > 0 \quad \text{(infeasible)}$

Case 2: Constrained optimum on boundary $x_1 + x_2 = 2$ The constraint is active, so we need: $\nabla f(x^*) = \lambda \nabla g(x^*)$ where $\lambda \geq 0$

At any point (x_1, x_2) on the constraint: - $\nabla f = (2(x_1 - 2), 2(x_2 - 2))$ - $\nabla g = (1, 1)$

For the KKT condition: $(2(x_1 - 2), 2(x_2 - 2)) = \lambda(1, 1)$

This gives: $x_1 - 2 = x_2 - 2 = \frac{\lambda}{2}$, so $x_1 = x_2$

From constraint: $2x_1 = 2$, so $x_1 = x_2 = 1$

Solution: $x^* = (1, 1)$ with $\lambda = -2$

Since $\lambda < 0$, this violates dual feasibility. The constraint is not active at the optimum.

Correct Analysis: The unconstrained optimum is infeasible, so we project onto the constraint boundary. The optimal point is where the objective function gradient is orthogonal to the constraint boundary, giving $x^* = (1, 1)$.

Example 1.6 (Multiple Active Constraints - Corner Solution)**.** Consider the problem:

$$\min \quad f(x) = x_1 + 2x_2 \tag{1.46}$$
$$\text{s.t.} \quad g_1(x) = -x_1 \leq 0 \tag{1.47}$$
$$g_2(x) = -x_2 \leq 0 \tag{1.48}$$
$$g_3(x) = x_1 + x_2 - 3 \leq 0 \tag{1.49}$$

Geometric Analysis:

The feasible region is a triangle with vertices at $(0, 0)$, $(3, 0)$, and $(0, 3)$.

At the origin $(0, 0)$: - $\nabla f = (1, 2)$ (points northeast) - $\nabla g_1 = (-1, 0)$ (points west) - $\nabla g_2 = (0, -1)$ (points south)

For KKT: $\nabla f + \lambda_1 \nabla g_1 + \lambda_2 \nabla g_2 = 0$ $(1, 2) + \lambda_1(-1, 0) + \lambda_2(0, -1) = (0, 0)$

This gives: $\lambda_1 = 1 > 0$ and $\lambda_2 = 2 > 0$

Geometric Insight: The objective gradient $(1, 2)$ lies in the positive cone generated by $(-1, 0)$ and $(0, -1)$, confirming that $(0, 0)$ is optimal.

1.4.3 Visualization of Geometric Relationships

Example 1.7 (Tangency Condition for Inequality Constraints)**.** Consider the classical problem:

$$\min \quad f(x) = x_1^2 + x_2^2 \tag{1.50}$$
$$\text{s.t.} \quad g(x) = x_1^2 + x_2^2 - 1 \leq 0 \tag{1.51}$$

Geometric Analysis:

The unconstrained minimum is at $(0, 0)$, which satisfies the constraint since $0+0-1 = -1 < 0$.

Since the constraint is inactive ($\lambda = 0$), the KKT conditions reduce to $\nabla f = 0$, which is satisfied at $(0, 0)$.

Geometric Insight: When the unconstrained optimum lies in the interior of the feasible region, the constraint has no effect on the solution.

Example 1.8 (Non-tangency Case with Active Constraint)**.** Consider:

$$\min \quad f(x) = -x_1 - x_2 \tag{1.52}$$
$$\text{s.t.} \quad g(x) = x_1^2 + x_2^2 - 1 \leq 0 \tag{1.53}$$

Analysis:

The objective gradient $\nabla f = (-1, -1)$ points toward increasing x_1 and x_2. The unconstrained optimum would be at infinity, so the constraint must be active.

On the circle $x_1^2 + x_2^2 = 1$: - $\nabla f = (-1, -1)$ - $\nabla g = (2x_1, 2x_2)$

For KKT: $(-1, -1) + \lambda(2x_1, 2x_2) = (0, 0)$

This gives: $-1 + 2\lambda x_1 = 0$ and $-1 + 2\lambda x_2 = 0$

So $x_1 = x_2 = \frac{1}{2\lambda}$

From constraint: $2\left(\frac{1}{2\lambda}\right)^2 = 1$, giving $\lambda = \frac{1}{\sqrt{2}}$

Solution: $x^* = \left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right)$

Geometric Insight: At the optimum, ∇f and ∇g are parallel and point in opposite directions, indicating tangency between level curves of the objective and the constraint boundary.

1.4.4 Normal Cone and Tangent Cone Interpretation

Definition 1.6 (Tangent Cone)**.** The tangent cone to the feasible set $\mathcal{S}$ at a point x^* is:

$$T_{\mathcal{S}}(x^*) = \{d : \exists t_k \downarrow 0, \exists d_k \to d \text{ such that } x^* + t_k d_k \in \mathcal{S}\}$$

Definition 1.7 (Normal Cone)**.** The normal cone to the feasible set $\mathcal{S}$ at a point x^* is:

$$N_{\mathcal{S}}(x^*) = \{g : g^T d \leq 0 \text{ for all } d \in T_{\mathcal{S}}(x^*)\}$$

Theorem

Geometric Characterization of Optimality
A feasible point x^* is optimal if and only if:

$$\nabla f(x^*) \in N_{\mathcal{S}}(x^*)$$

This means the objective gradient must be in the normal cone to the feasible set.

1.4.5 Fritz John vs. KKT Conditions

Example 1.9 (Constraint Qualification Failure)**.** Consider the pathological case:

$$\min \quad f(x) = x_2 \tag{1.54}$$
$$\text{s.t.} \quad g_1(x) = x_1^3 - x_2 \leq 0 \tag{1.55}$$
$$g_2(x) = -x_1^3 - x_2 \leq 0 \tag{1.56}$$

At $x^* = (0,0)$: - Both constraints are active: $g_1(0,0) = g_2(0,0) = 0$ - $\nabla g_1(0,0) = (0,-1)$ and $\nabla g_2(0,0) = (0,-1)$

The constraint gradients are linearly dependent, violating LICQ.

Fritz John Conditions: There exist $\lambda_0, \lambda_1, \lambda_2 \geq 0$, not all zero, such that:

$$\lambda_0 \nabla f + \lambda_1 \nabla g_1 + \lambda_2 \nabla g_2 = 0$$

This gives: $\lambda_0(0,1) + (\lambda_1 + \lambda_2)(0,-1) = (0,0)$

Solution: $\lambda_0 = \lambda_1 + \lambda_2$ with any non-negative values.

Geometric Insight: When constraint qualification fails, the normal cone may have "gaps" that prevent the standard KKT conditions from holding.

Real-World Application

Resource Allocation

A manufacturing company must allocate limited resources among different products. The problem can be formulated as:

$$\max \quad \sum_{i=1}^{n} p_i x_i - \sum_{i=1}^{n} c_i x_i^2 \tag{1.57}$$
$$\text{s.t.} \quad \sum_{i=1}^{n} a_{ji} x_i \leq b_j, \quad j = 1, \ldots, m \tag{1.58}$$
$$x_i \geq 0, \quad i = 1, \ldots, n \tag{1.59}$$

where x_i is the production level, p_i is the unit price, c_i represents production costs, and a_{ji} is the resource requirement.

Geometric Interpretation: At the optimal allocation, the marginal profit gradient is a non-negative combination of the resource constraint gradients, meaning no profitable direction exists that satisfies all resource limitations.

Python Code

```
import numpy as np
from scipy.optimize import minimize
import matplotlib.pyplot as plt

def geometric_kkt_visualization():
    """
    Visualize the geometric interpretation of KKT conditions
    """
```

```python
# Example: min (x1-2)^2 + (x2-2)^2 subject to x1 + x2 <= 2

def objective_constrained(x):
    return (x[0] - 2)**2 + (x[1] - 2)**2

def constraint_ineq(x):
    return 2 - x[0] - x[1]  # x1 + x2 <= 2

# Solve the optimization problem
constraints = [
    {'type': 'ineq', 'fun': constraint_ineq},
    {'type': 'ineq', 'fun': lambda x: x[0]},  # x1 >= 0
    {'type': 'ineq', 'fun': lambda x: x[1]}   # x2 >= 0
]

x0 = np.array([1.0, 1.0])
result = minimize(objective_constrained, x0, method='SLSQP',
                  constraints=constraints)

# Create visualization
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(15, 6))

# Plot 1: Objective function contours and constraints
x1 = np.linspace(-0.5, 3, 100)
x2 = np.linspace(-0.5, 3, 100)
X1, X2 = np.meshgrid(x1, x2)
Z = (X1 - 2)**2 + (X2 - 2)**2

# Contour plot
contours = ax1.contour(X1, X2, Z, levels=15, alpha=0.7, colors='blue')
ax1.clabel(contours, inline=True, fontsize=8)

# Constraint boundary
x1_constraint = np.linspace(0, 2.5, 100)
x2_constraint = 2 - x1_constraint

# Feasible region
x1_fill = np.linspace(0, 2, 100)
x2_fill = 2 - x1_fill
x2_lower = np.zeros_like(x1_fill)

ax1.fill_between(x1_fill, x2_lower, x2_fill, alpha=0.3,
                 color='lightgreen', label='Feasible Region')
ax1.plot(x1_constraint, x2_constraint, 'r-', linewidth=2,
        label='$x_1 + x_2 = 2$')

# Mark points
ax1.plot(2, 2, 'bs', markersize=10, label='Unconstrained Opt (2,2)')
ax1.plot(result.x[0], result.x[1], 'ro', markersize=10,
        label=f'Constrained Opt ({result.x[0]:.1f},{result.x[1]:.1f})')

ax1.set_xlim(-0.5, 3)
```

```
ax1.set_ylim(-0.5, 3)
ax1.set_xlabel('$x_1$')
ax1.set_ylabel('$x_2$')
ax1.set_title('Constrained Optimization Problem')
ax1.legend()
ax1.grid(True, alpha=0.3)

# Plot 2: Gradient vectors at optimal point
x_opt = result.x

# Objective gradient at optimal point
grad_f = 2 * (x_opt - np.array([2, 2]))

# Constraint gradient
grad_g = np.array([1, 1])  # gradient of x1 + x2

# Plot gradient vectors
ax2.quiver(x_opt[0], x_opt[1], grad_f[0], grad_f[1],
          angles='xy', scale_units='xy', scale=1, color='blue',
          width=0.005, label='$\\nabla f(x^*)$')
ax2.quiver(x_opt[0], x_opt[1], -grad_g[0], -grad_g[1],
          angles='xy', scale_units='xy', scale=1, color='red',
          width=0.005, label='$-\\nabla g(x^*)$')

# Show that gradients are parallel (KKT condition)
lambda_val = np.dot(grad_f, grad_g) / np.dot(grad_g, grad_g)
scaled_grad_g = lambda_val * grad_g

ax2.quiver(x_opt[0], x_opt[1], scaled_grad_g[0], scaled_grad_g[1],
          angles='xy', scale_units='xy', scale=1, color='green',
          width=0.005, linestyle='--',
          label=f'$\\lambda \\nabla g(x^*)$ ($\\lambda$={lambda_val:.2f})')

# Plot constraint line near optimal point
x1_local = np.linspace(x_opt[0]-0.5, x_opt[0]+0.5, 100)
x2_local = 2 - x1_local
ax2.plot(x1_local, x2_local, 'k-', linewidth=2, alpha=0.7,
        label='Constraint Boundary')

# Mark optimal point
ax2.plot(x_opt[0], x_opt[1], 'ro', markersize=10, label='Optimal Point')

ax2.set_xlim(x_opt[0]-0.8, x_opt[0]+0.8)
ax2.set_ylim(x_opt[1]-0.8, x_opt[1]+0.8)
ax2.set_xlabel('$x_1$')
ax2.set_ylabel('$x_2$')
ax2.set_title('KKT Condition: $\\nabla f + \\lambda \\nabla g = 0$')
ax2.legend()
ax2.grid(True, alpha=0.3)
ax2.set_aspect('equal')

plt.tight_layout()
plt.show()
```

```
    return result

def tangent_normal_cone_example():
    """
    Illustrate tangent and normal cones
    """

    # Consider feasible set: x1^2 + x2^2 <= 1, x1 >= 0
    fig, ax = plt.subplots(1, 1, figsize=(10, 8))

    # Draw feasible region
    theta = np.linspace(0, np.pi, 100)
    x1_circle = np.cos(theta)
    x2_circle = np.sin(theta)

    # Add bottom part
    x1_full = np.concatenate([x1_circle, [0]])
    x2_full = np.concatenate([x2_circle, [0]])

    ax.fill(x1_full, x2_full, alpha=0.3, color='lightblue',
           label='Feasible Region')
    ax.plot(x1_circle, x2_circle, 'b-', linewidth=2, label='Constraint Boundary')
    ax.axvline(x=0, color='black', linewidth=2, ymin=0, ymax=0.5)

    # Point on boundary
    point = np.array([0, 1])
    ax.plot(point[0], point[1], 'ro', markersize=10, label='Point on Boundary')

    # Tangent cone directions
    tangent_directions = np.array([[1, 0], [0, -1], [0.5, -0.866]])
    for i, direction in enumerate(tangent_directions):
        ax.arrow(point[0], point[1], 0.3*direction[0], 0.3*direction[1],
                head_width=0.05, head_length=0.05, fc='green', ec='green',
                alpha=0.7)

    # Normal cone (outward normals)
    normal_directions = np.array([[0, 1], [-1, 0]])
    for direction in normal_directions:
        ax.arrow(point[0], point[1], 0.4*direction[0], 0.4*direction[1],
                head_width=0.05, head_length=0.05, fc='red', ec='red',
                alpha=0.7, linewidth=2)

    # Add text annotations
    ax.text(0.2, 0.8, 'Tangent Cone', color='green', fontsize=12, fontweight='bold')
    ax.text(-0.3, 1.2, 'Normal Cone', color='red', fontsize=12, fontweight='bold')

    ax.set_xlim(-1.5, 1.5)
    ax.set_ylim(-0.5, 1.5)
    ax.set_xlabel('$x_1$')
    ax.set_ylabel('$x_2$')
    ax.set_title('Tangent and Normal Cones at Boundary Point')
    ax.legend()
```

```
    ax.grid(True, alpha=0.3)
    ax.set_aspect('equal')

    plt.show()

# Execute visualizations
print("Geometric KKT Visualization:")
result = geometric_kkt_visualization()
print(f"Optimal solution: {result.x}")
print(f"Optimal value: {result.fun}")

print("\nTangent and Normal Cone Illustration:")
tangent_normal_cone_example()
```

Exploration

Advanced Geometric Investigations
Explore these geometric concepts further:

1. **Constraint Curvature Effects**: Investigate how the curvature of constraint boundaries affects the geometry of optimal solutions, particularly for nonlinear constraints.

2. **Degeneracy Analysis**: Study cases where multiple constraints are active and constraint gradients are nearly linearly dependent. Observe how small perturbations can change the active set.

3. **Second-Order Geometry**: Examine the role of the Hessian of the Lagrangian in determining the nature of critical points (minimum, maximum, or saddle point).

4. **Parametric Analysis**: Investigate how the optimal solution path varies as constraint parameters change, observing bifurcation points where the active set changes.

Exercise 1.3. Consider the problem: minimize $x_1^2 + x_2^2$ subject to $x_1^2 + x_2^2 \leq 1$ and $x_1 + x_2 \geq 1$. Set up the KKT conditions and solve graphically. Identify the geometric relationship between the objective gradient and constraint gradients at the optimal point.

Exercise 1.4. Prove that if x^* satisfies the KKT conditions for a convex optimization problem, then x^* is a global minimum.

Chapter 2

Linear Programming

2.1 Introduction to Linear Programming

Linear Programming (LP) is a fundamental class of constrained optimization problems where both the objective function and constraints are linear.

Definition 2.1 (Linear Programming Problem)**.** The standard form of a linear programming problem is:

$$\min \quad c^T x \tag{2.1}$$
$$\text{s.t.} \quad Ax = b \tag{2.2}$$
$$x \geq 0 \tag{2.3}$$

where $c \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$, and $b \in \mathbb{R}^m$.

2.2 Fundamental Theorem of Linear Programming

Theorem

Fundamental Theorem of Linear Programming
If a linear programming problem has an optimal solution, then there exists an optimal solution that is a basic feasible solution (vertex of the feasible region).

2.2.1 Illustrative Examples of the Fundamental Theorem

Example 2.1 (Two-Variable Linear Program)**.** Consider the problem:

$$\max \quad 3x_1 + 2x_2 \tag{2.4}$$
$$\text{s.t.} \quad x_1 + 2x_2 \leq 4 \tag{2.5}$$
$$2x_1 + x_2 \leq 4 \tag{2.6}$$
$$x_1, x_2 \geq 0 \tag{2.7}$$

The feasible region is a polygon with vertices at:

- $(0, 0)$: objective value $= 0$

- $(0, 2)$: objective value $= 4$
- $\left(\frac{4}{3}, \frac{4}{3}\right)$: objective value $= 3 \cdot \frac{4}{3} + 2 \cdot \frac{4}{3} = \frac{20}{3}$
- $(2, 0)$: objective value $= 6$

The maximum occurs at vertex $(2, 0)$ with value 6, confirming the fundamental theorem.

Example 2.2 (Unbounded Linear Program)**.** Consider:

$$\begin{aligned} \max \quad & x_1 + x_2 && (2.8) \\ \text{s.t.} \quad & -x_1 + x_2 \leq 1 && (2.9) \\ & x_1, x_2 \geq 0 && (2.10) \end{aligned}$$

The feasible region is unbounded in the direction $(1, 1)$, which is also the direction of objective improvement. Hence, the problem is unbounded above.

Geometric verification: Any point $(t, t + 1)$ for $t \geq 0$ is feasible and gives objective value $2t + 1 \to \infty$ as $t \to \infty$.

Example 2.3 (Infeasible Linear Program)**.** Consider:

$$\begin{aligned} \min \quad & x_1 + x_2 && (2.11) \\ \text{s.t.} \quad & x_1 + x_2 \leq 1 && (2.12) \\ & x_1 + x_2 \geq 2 && (2.13) \\ & x_1, x_2 \geq 0 && (2.14) \end{aligned}$$

The constraints $x_1 + x_2 \leq 1$ and $x_1 + x_2 \geq 2$ are contradictory, making the feasible set empty.

Example 2.4 (Degenerate Basic Feasible Solution)**.** Consider the problem:

$$\begin{aligned} \max \quad & x_1 + x_2 && (2.15) \\ \text{s.t.} \quad & x_1 \leq 2 && (2.16) \\ & x_2 \leq 2 && (2.17) \\ & x_1 + x_2 \leq 3 && (2.18) \\ & x_1, x_2 \geq 0 && (2.19) \end{aligned}$$

At the point $(1, 2)$, three constraints are active: $x_2 = 2$, $x_1 + x_2 = 3$, and the implied constraint from these two. This represents a degenerate basic feasible solution where more than n constraints are active.

2.3 Multiple Solution Methods for Linear Programming

2.3.1 The Simplex Method

The simplex method is the classical algorithm for solving linear programming problems, but it is not the only approach.

Definition 2.2 (Basic Feasible Solution)**.** A basic feasible solution corresponds to a vertex of the feasible region where exactly n constraints are active (including non-negativity constraints).

Example 2.5 (Simplex Method Illustration)**.** For the standard form problem:

$$\min \quad -3x_1 - 2x_2 \tag{2.20}$$
$$\text{s.t.} \quad x_1 + 2x_2 + s_1 = 4 \tag{2.21}$$
$$2x_1 + x_2 + s_2 = 4 \tag{2.22}$$
$$x_1, x_2, s_1, s_2 \geq 0 \tag{2.23}$$

Initial tableau:

	x_1	x_2	s_1	s_2	RHS
s_1	1	2	1	0	4
s_2	2	1	0	1	4
z	3	2	0	0	0

Iteration 1: x_1 enters (most negative reduced cost), s_2 leaves (minimum ratio test: $\min\{4/1, 4/2\} = 2$).

After pivoting:

	x_1	x_2	s_1	s_2	RHS
s_1	0	$\frac{3}{2}$	1	$-\frac{1}{2}$	2
x_1	1	$\frac{1}{2}$	0	$\frac{1}{2}$	2
z	0	$\frac{1}{2}$	0	$-\frac{3}{2}$	-6

Since all reduced costs are non-negative, the optimal solution is $x_1 = 2$, $x_2 = 0$ with objective value 6.

2.3.2 Interior Point Methods

Interior point methods approach the optimal solution through the interior of the feasible region.

Definition 2.3 (Primal-Dual Interior Point Method for LP)**.** The primal-dual method simultaneously solves the primal and dual problems:

$$\text{Primal:} \quad \min c^T x \text{ s.t. } Ax = b, x \geq 0 \tag{2.24}$$
$$\text{Dual:} \quad \max b^T y \text{ s.t. } A^T y + s = c, s \geq 0 \tag{2.25}$$

The central path is parameterized by $\mu > 0$ and satisfies:

$$Ax = b \tag{2.26}$$
$$A^T y + s = c \tag{2.27}$$
$$xs = \mu e \tag{2.28}$$
$$x, s > 0 \tag{2.29}$$

where xs denotes component-wise multiplication.

Example 2.6 (Interior Point Path)**.** For the problem:

$$\min \quad -x_1 - x_2 \tag{2.30}$$
$$\text{s.t.} \quad x_1 + x_2 \leq 2 \tag{2.31}$$
$$x_1, x_2 \geq 0 \tag{2.32}$$

The central path solutions for decreasing μ are:

- $\mu = 1$: $(x_1, x_2) \approx (0.5, 0.5)$
- $\mu = 0.1$: $(x_1, x_2) \approx (0.9, 0.9)$
- $\mu \to 0$: $(x_1, x_2) \to (1, 1)$ (optimal vertex)

The path stays in the interior until $\mu \to 0$, then converges to the optimal vertex.

2.3.3 Dual Simplex Method

The dual simplex method maintains dual feasibility while seeking primal feasibility.

Example 2.7 (Dual Simplex Application)**.** Consider the problem after adding a constraint:

$$\begin{aligned} \min \quad & x_1 + x_2 && (2.33)\\ \text{s.t.} \quad & x_1 + 2x_2 \geq 4 && (2.34)\\ & 2x_1 + x_2 \geq 4 && (2.35)\\ & x_1, x_2 \geq 0 && (2.36) \end{aligned}$$

Converting to standard form with slack variables $s_1, s_2 \leq 0$:

$$\begin{aligned} \min \quad & x_1 + x_2 && (2.37)\\ \text{s.t.} \quad & x_1 + 2x_2 - s_1 = 4 && (2.38)\\ & 2x_1 + x_2 - s_2 = 4 && (2.39)\\ & x_1, x_2 \geq 0, \quad s_1, s_2 \leq 0 && (2.40) \end{aligned}$$

The dual simplex method starts with $x_1 = x_2 = 0$, giving $s_1 = s_2 = -4 < 0$ (primal infeasible but dual feasible), then iteratively restores primal feasibility.

2.3.4 Network Simplex Method

For network flow problems, the network simplex method exploits the special structure.

Example 2.8 (Minimum Cost Flow Problem)**.** Consider a transportation network with:

- Supply nodes: $s_1 = 10$, $s_2 = 15$
- Demand nodes: $d_1 = 8$, $d_2 = 12$, $d_3 = 5$
- Arc costs: c_{ij} for flow from node i to node j

The problem becomes:

$$\begin{aligned} \min \quad & \sum_{(i,j)\in A} c_{ij} x_{ij} && (2.41)\\ \text{s.t.} \quad & \sum_{j:(i,j)\in A} x_{ij} - \sum_{j:(j,i)\in A} x_{ji} = b_i \quad \forall i && (2.42)\\ & x_{ij} \geq 0 \quad \forall (i,j) \in A && (2.43) \end{aligned}$$

The network simplex method maintains a spanning tree solution and performs pivots by adding and removing arcs to maintain the tree structure.

2.3.5 Ellipsoid Method

The ellipsoid method has theoretical importance as the first polynomial-time algorithm for linear programming.

Example 2.9 (Ellipsoid Method Concept)**.** Starting with an ellipsoid E_0 containing the optimal solution, the method:

1. **Query the center** x_k of current ellipsoid E_k
2. **Check feasibility**: If x_k is optimal, stop
3. **Generate cutting plane**: Find violated constraint $a^T x \leq b$ with $a^T x_k > b$
4. **Update ellipsoid**: Construct smaller ellipsoid E_{k+1} containing the intersection of E_k with the half-space $a^T x \leq b$

For the constraint $x_1 + x_2 \leq 2$ violated at center $(3, 1)$:

- Current ellipsoid center: $(3, 1)$
- Cutting plane: $x_1 + x_2 \leq 2$
- New ellipsoid center shifts toward the feasible region

The method converges in polynomial time but is not practical due to poor constant factors.

2.4 Duality Theory

Theorem

Strong Duality Theorem
If the primal problem has an optimal solution with objective value z^*, then the dual problem also has an optimal solution with the same objective value.

Example 2.10 (Duality Analysis)**. Primal:**

$$\begin{aligned} \max \quad & 3x_1 + 2x_2 && (2.44) \\ \text{s.t.} \quad & x_1 + 2x_2 \leq 4 && (2.45) \\ & 2x_1 + x_2 \leq 4 && (2.46) \\ & x_1, x_2 \geq 0 && (2.47) \end{aligned}$$

Dual:

$$\begin{aligned} \min \quad & 4y_1 + 4y_2 && (2.48) \\ \text{s.t.} \quad & y_1 + 2y_2 \geq 3 && (2.49) \\ & 2y_1 + y_2 \geq 2 && (2.50) \\ & y_1, y_2 \geq 0 && (2.51) \end{aligned}$$

Solution process:
First, solve the primal by checking vertices:

- $(0, 0)$: objective value $= 0$
- $(0, 2)$: objective value $= 4$
- $(2, 0)$: objective value $= 6$
- $\left(\frac{4}{3}, \frac{4}{3}\right)$: objective value $= 3 \cdot \frac{4}{3} + 2 \cdot \frac{4}{3} = \frac{20}{3} \approx 6.67$

The primal optimal is $x^* = \left(\frac{4}{3}, \frac{4}{3}\right)$ with value $\frac{20}{3}$.

For the dual, using complementary slackness since both primal constraints are tight:

$$y_1 + 2y_2 = 3 \quad (2.52)$$

$$2y_1 + y_2 = 2 \quad (2.53)$$

Solving this system: From the second equation: $y_1 = 1 - \frac{y_2}{2}$ Substituting: $\left(1 - \frac{y_2}{2}\right) + 2y_2 = 3$ Simplifying: $1 + \frac{3y_2}{2} = 3$, so $y_2 = \frac{4}{3}$ and $y_1 = \frac{1}{3}$

Dual value: $4 \cdot \frac{1}{3} + 4 \cdot \frac{4}{3} = \frac{4}{3} + \frac{16}{3} = \frac{20}{3}$ ✓

Example 2.11 (Economic Interpretation of Duality)**.** Consider a production planning problem:

Primal interpretation: Maximize profit from producing x_1 units of product 1 and x_2 units of product 2, subject to resource limitations.

Dual interpretation: The dual variables y_1, y_2 represent shadow prices of resources. The dual problem asks: what is the minimum cost of the resources needed, where resources are valued at their shadow prices?

At optimality, the shadow price y_j^* represents the marginal value of one additional unit of resource j.

Real-World Application

Multi-Method Approach in Practice

Real-world linear programming often employs multiple methods:

1. **Preprocessing**: Problem reduction and reformulation
2. **Method Selection**:
   - Simplex for small to medium problems
   - Interior point for large-scale problems
   - Network simplex for network flow structures
   - Dual simplex for sensitivity analysis
3. **Postprocessing**: Solution interpretation and sensitivity analysis
4. **Warm Starting**: Using previous solutions when problem parameters change

Modern LP solvers automatically choose the most appropriate method based on problem characteristics.

Exploration

Comparative Analysis of LP Methods
Investigate the following aspects across different LP solution methods:

1. **Theoretical Complexity**: Compare worst-case and average-case complexity
2. **Practical Performance**: Benchmark on different problem classes
3. **Numerical Stability**: Analyze behavior with poorly conditioned problems
4. **Parallelization Potential**: Examine which methods can benefit from parallel computing
5. **Memory Requirements**: Compare storage needs for large-scale problems

Exercise 2.1. Consider the linear program:

$$\max \quad 2x_1 + 3x_2 + x_3 \tag{2.54}$$
$$\text{s.t.} \quad x_1 + x_2 + x_3 \leq 6 \tag{2.55}$$
$$2x_1 + x_2 \leq 8 \tag{2.56}$$
$$x_2 + 2x_3 \leq 7 \tag{2.57}$$
$$x_1, x_2, x_3 \geq 0 \tag{2.58}$$

Formulate the dual problem and use complementary slackness to verify that $x^* = (3, 2, 1)$ and $y^* = (1, 1, 1)$ are optimal for primal and dual respectively.

Exercise 2.2. Prove that if both primal and dual linear programs have feasible solutions, then both have optimal solutions with equal objective values (Strong Duality Theorem).

Exercise 2.3. For a network flow problem with n nodes and m arcs, explain why the network simplex method maintains exactly $n - 1$ basic variables in a spanning tree structure, and describe how degeneracy is handled in this context.

Chapter 3

Quadratic Programming and Advanced Methods

3.1 Quadratic Programming

Quadratic Programming (QP) extends linear programming by allowing quadratic terms in the objective function while maintaining linear constraints.

Definition 3.1 (Quadratic Programming Problem)**.** The standard quadratic programming problem is:

$$\min \quad \frac{1}{2}x^TQx + c^Tx \tag{3.1}$$
$$\text{s.t.} \quad Ax \leq b \tag{3.2}$$
$$Ex = d \tag{3.3}$$
$$x \geq 0 \tag{3.4}$$

where $Q \in \mathbb{R}^{n\times n}$ is symmetric.

3.1.1 Classification of Quadratic Programs

Definition 3.2 (Types of Quadratic Programming)**.** Based on the properties of matrix Q:

- **Convex QP**: $Q \succeq 0$ (positive semidefinite)
- **Strictly Convex QP**: $Q \succ 0$ (positive definite)
- **Indefinite QP**: Q has both positive and negative eigenvalues
- **Concave QP**: $Q \preceq 0$ (negative semidefinite)

Theorem

Global Optimality for Convex QP
For a convex quadratic program where $Q \succeq 0$, any local minimum is a global minimum. Furthermore, if $Q \succ 0$, the global minimum is unique (if it exists).

3.1.2 Convex Quadratic Programming

When $Q \succeq 0$ (positive semidefinite), the problem is convex and the KKT conditions are necessary and sufficient for optimality.

Theorem

KKT Conditions for Quadratic Programming
For the QP problem with $Q \succeq 0$, a point x^* is optimal if and only if there exist multipliers $\lambda^* \geq 0$, μ^*, and $\nu^* \geq 0$ such that:

$$\text{Stationarity:} \quad Qx^* + c + A^T\lambda^* + E^T\mu^* - \nu^* = 0 \tag{3.5}$$
$$\text{Primal feasibility:} \quad Ax^* \leq b, \quad Ex^* = d, \quad x^* \geq 0 \tag{3.6}$$
$$\text{Dual feasibility:} \quad \lambda^* \geq 0, \quad \nu^* \geq 0 \tag{3.7}$$
$$\text{Complementary slackness:} \quad \lambda_i^*(A_i x^* - b_i) = 0, \quad \nu_j^* x_j^* = 0 \tag{3.8}$$

Example 3.1 (Simple Quadratic Program)**.** Consider the problem:

$$\min \quad \frac{1}{2}(x_1^2 + x_2^2) + x_1 + x_2 \tag{3.9}$$
$$\text{s.t.} \quad x_1 + x_2 \leq 1 \tag{3.10}$$
$$x_1, x_2 \geq 0 \tag{3.11}$$

Here $Q = I$, $c = (1,1)^T$, $A = (1,1)$, $b = 1$.
Solution process:
The Lagrangian is:

$$\mathcal{L} = \frac{1}{2}(x_1^2 + x_2^2) + x_1 + x_2 + \lambda(x_1 + x_2 - 1) - \nu_1 x_1 - \nu_2 x_2$$

KKT conditions:

$$x_1 + 1 + \lambda - \nu_1 = 0 \tag{3.12}$$
$$x_2 + 1 + \lambda - \nu_2 = 0 \tag{3.13}$$
$$x_1 + x_2 \leq 1, \quad \lambda \geq 0, \quad \lambda(x_1 + x_2 - 1) = 0 \tag{3.14}$$
$$x_1, x_2 \geq 0, \quad \nu_1, \nu_2 \geq 0, \quad \nu_1 x_1 = \nu_2 x_2 = 0 \tag{3.15}$$

Case 1: Interior solution ($\nu_1 = \nu_2 = 0$, $\lambda = 0$) From stationarity: $x_1 = x_2 = -1 < 0$ (violates non-negativity)

Case 2: Constraint active ($x_1 + x_2 = 1$, $\nu_1 = \nu_2 = 0$) From stationarity: $x_1 = x_2 = -1 - \lambda$ From constraint: $2(-1-\lambda) = 1 \Rightarrow \lambda = -\frac{3}{2} < 0$

Case 3: $x_1 = 0$ (so $\nu_1 \geq 0$), $\nu_2 = 0$ From stationarity: $1 + \lambda - \nu_1 = 0$ and $x_2 + 1 + \lambda = 0$ So $x_2 = -1 - \lambda$ and $\nu_1 = 1 + \lambda$

For feasibility: $x_2 \geq 0 \Rightarrow \lambda \leq -1$ and $\nu_1 \geq 0 \Rightarrow \lambda \geq -1$ Thus $\lambda = -1$, giving $x_2 = 0$, $\nu_1 = 0$.

By symmetry, optimal solution is $x^* = (0,0)$ with objective value 0.

Example 3.2 (Portfolio Optimization QP)**.** The classical Markowitz portfolio optimization problem:

$$\begin{aligned} \min \quad & \frac{1}{2} w^T \Sigma w && (3.16)\\ \text{s.t.} \quad & \mathbf{1}^T w = 1 && (3.17)\\ & \mu^T w \geq r_{\min} && (3.18)\\ & w \geq 0 && (3.19) \end{aligned}$$

where w is the portfolio weight vector, Σ is the covariance matrix, μ is the expected return vector.

Economic interpretation: Minimize portfolio variance subject to budget constraint, minimum return requirement, and no short-selling.

KKT conditions:

$$\Sigma w^* + \lambda \mathbf{1} - \nu \mu - \gamma = 0 \quad (3.20)$$

$$\mathbf{1}^T w^* = 1 \quad (3.21)$$

$$\mu^T w^* \geq r_{\min}, \quad \nu \geq 0, \quad \nu(\mu^T w^* - r_{\min}) = 0 \quad (3.22)$$

$$w^* \geq 0, \quad \gamma \geq 0, \quad \gamma_i w_i^* = 0 \quad (3.23)$$

where λ is the Lagrange multiplier for the budget constraint, ν for the return constraint, and γ for non-negativity.

Example 3.3 (Quadratic Program with Equality Constraints Only)**.** Consider:

$$\begin{aligned} \min \quad & \frac{1}{2} x^T Q x + c^T x && (3.24)\\ \text{s.t.} \quad & Ax = b && (3.25) \end{aligned}$$

where $Q \succ 0$. The KKT system becomes:

$$\begin{bmatrix} Q & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} x \\ \lambda \end{bmatrix} = \begin{bmatrix} -c \\ b \end{bmatrix}$$

This system has a unique solution since $Q \succ 0$ makes the matrix nonsingular.

Explicit solution:

$$x^* = Q^{-1}(A^T(AQ^{-1}A^T)^{-1}b - c)$$
$$\lambda^* = (AQ^{-1}A^T)^{-1}(AQ^{-1}c + b)$$

3.2 Active Set Methods

Active set methods solve QP by iteratively working with subsets of constraints that are active at the current iterate.

Definition 3.3 (Active Set Method Algorithm)**.**

1. **Initialization**: Start with a feasible point x_0 and working set $\mathcal{W}_0$
2. **Solve Equality QP**: Solve subproblem with working set constraints as equalities
3. **Optimality Check**: Check KKT conditions for current working set
4. **Add/Remove Constraints**: Update working set based on violations
5. **Iteration**: Repeat until convergence

Theorem

Finite Convergence of Active Set Method
For a strictly convex quadratic program, the active set method converges to the optimal solution in a finite number of iterations.

Example 3.4 (Active Set Method Trace)**.** Consider the QP:

$$\min \quad \frac{1}{2}(x_1^2 + x_2^2) - 2x_1 - 6x_2 \tag{3.26}$$
$$\text{s.t.} \quad x_1 + x_2 \leq 2 \tag{3.27}$$
$$-x_1 + 2x_2 \leq 2 \tag{3.28}$$
$$2x_1 + x_2 \leq 3 \tag{3.29}$$
$$x_1, x_2 \geq 0 \tag{3.30}$$

Iteration 0: Start with $x_0 = (0, 0)$, working set $\mathcal{W}_0 = \{4, 5\}$ (both non-negativity constraints)

Solve equality QP with $x_1 = x_2 = 0$: Lagrangian gradient: $\nabla\mathcal{L} = (1 - \nu_1, 6 - \nu_2) = (0, 0)$ This gives $\nu_1 = 1 > 0$, $\nu_2 = 6 > 0$

Check optimality: Compute d to improve objective while maintaining active constraints. Direction d must satisfy $d_1 = d_2 = 0$, so $d = 0$. Check if we can remove constraints.

Since $\nu_2 = 6$ is largest, remove constraint $x_2 \geq 0$ from working set.

Iteration 1: Working set $\mathcal{W}_1 = \{4\}$ (only $x_1 \geq 0$)

Solve for optimal x with $x_1 = 0$: $\min \frac{1}{2}x_2^2 - 6x_2$ gives $x_2 = 6$

Check constraints: $x_1 = 0$, $x_2 = 6$ violates $x_1 + x_2 \leq 2$.

Add most violated constraint to working set.

Continue until convergence...

3.3 Interior Point Methods for QP

Interior point methods approach the optimal solution through the interior of the feasible region.

Definition 3.4 (Logarithmic Barrier Function for QP)**.** For the QP with inequality constraints $Ax \leq b$, the barrier function is:

$$\phi(x) = -\sum_{i=1}^{m} \log(b_i - A_i x)$$

The barrier subproblem is:

$$\min_x \frac{1}{2}x^T Q x + c^T x + \mu\phi(x)$$

where $\mu > 0$ is the barrier parameter.

Theorem

Central Path for QP
For a strictly convex QP, the central path $x(\mu)$ is well-defined for all $\mu > 0$ and converges to the optimal solution as $\mu \to 0^+$.

Example 3.5 (Barrier Method for QP)**.** Consider:

$$\min \quad \frac{1}{2}(x_1^2 + x_2^2) \tag{3.31}$$
$$\text{s.t.} \quad x_1 + x_2 \leq 1 \tag{3.32}$$
$$x_1, x_2 \geq 0 \tag{3.33}$$

The barrier subproblem is:

$$\min \frac{1}{2}(x_1^2 + x_2^2) - \mu \log(1 - x_1 - x_2) - \mu \log x_1 - \mu \log x_2$$

First-order conditions:

$$x_1 + \frac{\mu}{1 - x_1 - x_2} - \frac{\mu}{x_1} = 0 \tag{3.34}$$
$$x_2 + \frac{\mu}{1 - x_1 - x_2} - \frac{\mu}{x_2} = 0 \tag{3.35}$$

By symmetry, $x_1 = x_2$ on the central path. Let $x_1 = x_2 = t$:

$$t + \frac{\mu}{1 - 2t} - \frac{\mu}{t} = 0$$

Solving: $t^2(1 - 2t) + \mu t - \mu(1 - 2t) = 0$

As $\mu \to 0$: $t^2(1 - 2t) = 0$, giving $t = 0$ or $t = \frac{1}{2}$. Since we need $t > 0$, the limit point is $x^* = (0, 0)$.

3.3.1 Primal-Dual Interior Point Method

The primal-dual method solves the KKT system directly while maintaining positivity of inequality multipliers.

Theorem

Newton Direction for Primal-Dual Method
For the QP in standard form, the Newton direction is obtained by solving:

$$\begin{bmatrix} Q & A^T & E^T \\ \text{diag}(\lambda)A & \text{diag}(Ax - b) & 0 \\ E & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta \lambda \\ \Delta \mu \end{bmatrix} = - \begin{bmatrix} Qx + c + A^T\lambda + E^T\mu \\ \text{diag}(\lambda)(Ax - b) - \sigma\mu e \\ Ex - d \end{bmatrix} \tag{3.36}$$

where $\sigma \in (0, 1)$ is a centering parameter.

Example 3.6 (Primal-Dual Step Calculation)**.** For the system above, when $E = 0$ (no equality constraints), the simplified Newton system becomes:

$$\begin{bmatrix} Q & A^T \\ \Lambda A & S \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta \lambda \end{bmatrix} = \begin{bmatrix} -Qx - c - A^T\lambda \\ \sigma\mu e - \Lambda S e \end{bmatrix}$$

where $\Lambda = \text{diag}(\lambda)$ and $S = \text{diag}(s)$ with $s = b - Ax$.

The step sizes are chosen to maintain positivity: $\alpha = 0.95 \min\{1, \min_i\{-\lambda_i/\Delta\lambda_i, -s_i/\Delta s_i\}\}$.

3.4 Sequential Quadratic Programming (SQP)

SQP methods solve nonlinear programming problems by solving a sequence of quadratic programming subproblems.

Definition 3.5 (SQP Subproblem)**.** At iteration k, solve:

$$\min \quad \nabla f(x_k)^T d + \frac{1}{2} d^T B_k d \tag{3.37}$$
$$\text{s.t.} \quad \nabla g_i(x_k)^T d + g_i(x_k) \leq 0, \quad i = 1, \ldots, m \tag{3.38}$$
$$\nabla h_j(x_k)^T d + h_j(x_k) = 0, \quad j = 1, \ldots, p \tag{3.39}$$

where B_k is an approximation to the Hessian of the Lagrangian.

Theorem

Superlinear Convergence of SQP
Under appropriate regularity conditions, if B_k converges to the true Hessian of the Lagrangian, then SQP converges superlinearly to the optimal solution.

Example 3.7 (SQP for Nonlinear Problem)**.** Consider the nonlinear program:

$$\min \quad x_1^2 + x_2^2 \tag{3.40}$$
$$\text{s.t.} \quad x_1^2 + x_2^2 - 1 = 0 \tag{3.41}$$

Starting from $x_0 = (0.5, 0.5)$:
Iteration 1: $\nabla f(x_0) = (1, 1)$, $\nabla h(x_0) = (1, 1)$, $h(x_0) = -0.5$
SQP subproblem:

$$\min \quad d_1 + d_2 + \frac{1}{2}(d_1^2 + d_2^2) \tag{3.42}$$
$$\text{s.t.} \quad d_1 + d_2 - 0.5 = 0 \tag{3.43}$$

Solution: $d_1 = d_2 = 0.25$, giving $x_1 = (0.75, 0.75)$.
Check constraint: $0.75^2 + 0.75^2 = 1.125 \neq 1$.
Continue iterations until convergence to $x^* = (1/\sqrt{2}, 1/\sqrt{2})$.

3.5 Specialized QP Algorithms

3.5.1 Conjugate Gradient for QP

For large-scale QP with $Q \succ 0$, the conjugate gradient method can be applied directly.

Definition 3.6 (CG Method for QP)**.** The conjugate gradient method for solving $Qx = -c$ (unconstrained QP) generates search directions d_k that are Q-conjugate:

$$d_i^T Q d_j = 0 \quad \text{for } i \neq j$$

Example 3.8 (CG for Unconstrained QP)**.** For $\min \frac{1}{2}x^T Q x + c^T x$ with $Q \succ 0$:

Algorithm:

$$x_0 = \text{initial guess} \tag{3.44}$$
$$r_0 = Qx_0 + c, \quad d_0 = -r_0 \tag{3.45}$$
$$\alpha_k = \frac{r_k^T r_k}{d_k^T Q d_k} \tag{3.46}$$
$$x_{k+1} = x_k + \alpha_k d_k \tag{3.47}$$
$$r_{k+1} = r_k + \alpha_k Q d_k \tag{3.48}$$
$$\beta_k = \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k} \tag{3.49}$$
$$d_{k+1} = -r_{k+1} + \beta_k d_k \tag{3.50}$$

The method converges in at most n steps for an $n \times n$ system.

3.5.2 Gradient Projection Method

For box-constrained QP: $\min \frac{1}{2}x^T Q x + c^T x$ s.t. $l \leq x \leq u$.

Definition 3.7 (Gradient Projection)**.** The gradient projection method alternates between:

1. **Gradient step**: $y = x - \alpha \nabla f(x)$
2. **Projection step**: $x^+ = P_{[l,u]}(y)$ where $P_{[l,u]}(y)_i = \max\{l_i, \min\{y_i, u_i\}\}$

Example 3.9 (Projected Gradient for Box QP)**.** Consider:

$$\min \quad \frac{1}{2}(x_1^2 + x_2^2) + x_1 + x_2 \tag{3.51}$$
$$\text{s.t.} \quad 0 \leq x_1, x_2 \leq 1 \tag{3.52}$$

Starting from $x_0 = (0.5, 0.5)$:
$\nabla f(x_0) = (1.5, 1.5)$ $y = (0.5, 0.5) - 0.5(1.5, 1.5) = (-0.25, -0.25)$ $x_1 = P_{[0,1]}(y) = (0, 0)$

At $x_1 = (0, 0)$: $\nabla f(x_1) = (1, 1)$ Since both components are positive and we're at the lower bound, this is optimal.

Real-World Application

Support Vector Machines
The dual formulation of SVM leads to a quadratic programming problem:

$$\max \quad \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j K(x_i, x_j) \tag{3.53}$$
$$\text{s.t.} \quad \sum_{i=1}^{n} \alpha_i y_i = 0 \tag{3.54}$$
$$0 \leq \alpha_i \leq C, \quad i = 1, \ldots, n \tag{3.55}$$

where $K(x_i, x_j)$ is the kernel function and C is the regularization parameter.
This QP has a special structure that can be exploited by specialized algorithms like SMO (Sequential Minimal Optimization).

Real-World Application

Model Predictive Control
In MPC, the optimal control problem at each time step becomes a QP:

$$\min \quad \sum_{k=0}^{N-1} \left[x_k^T Q x_k + u_k^T R u_k\right] + x_N^T P x_N \tag{3.56}$$
$$\text{s.t.} \quad x_{k+1} = A x_k + B u_k \tag{3.57}$$
$$u_{\min} \le u_k \le u_{\max} \tag{3.58}$$
$$x_{\min} \le x_k \le x_{\max} \tag{3.59}$$

The real-time nature of MPC requires fast QP solvers, often using warm-starting techniques.

Exploration

Advanced QP Topics
Explore these advanced aspects of quadratic programming:

1. **Parametric QP**: Study how the optimal solution changes as problem parameters vary
2. **Robust QP**: Investigate QP formulations under parameter uncertainty
3. **Integer QP**: Examine mixed-integer quadratic programming problems
4. **Multi-parametric QP**: Analyze QP problems with multiple varying parameters
5. **Stochastic QP**: Consider QP problems with random variables

Exercise 3.1. Consider the QP:

$$\min \quad \frac{1}{2}(x_1^2 + 4x_2^2) + x_1 - 2x_2 \tag{3.60}$$
$$\text{s.t.} \quad x_1 + x_2 = 1 \tag{3.61}$$
$$x_1 - x_2 \le 0 \tag{3.62}$$
$$x_1, x_2 \ge 0 \tag{3.63}$$

Solve this problem by:

1. Writing the complete KKT system
2. Identifying all possible active sets
3. Solving each case systematically
4. Verifying the optimal solution

Exercise 3.2. Derive the KKT conditions for the portfolio optimization problem:

$$\begin{aligned} \min \quad & \frac{1}{2} w^T \Sigma w && (3.64) \\ \text{s.t.} \quad & \mathbf{1}^T w = 1 && (3.65) \\ & \mu^T w \geq r_{\min} && (3.66) \\ & w \geq 0 && (3.67) \end{aligned}$$

and interpret the economic meaning of each Lagrange multiplier. Show that when the return constraint is not active, the optimal portfolio has the closed-form solution:

$$w^* = \frac{\Sigma^{-1}\mathbf{1}}{\mathbf{1}^T \Sigma^{-1} \mathbf{1}}$$

Exercise 3.3. For the barrier method applied to QP, prove that the central path $x(\mu)$ satisfies:

$$\lim_{\mu \to 0^+} x(\mu) = x^*$$

where x^* is the optimal solution of the original QP. Discuss the rate of convergence.

Exercise 3.4. Implement the active set method for solving the general QP problem. Your algorithm should handle:

1. Finding an initial feasible point
2. Determining which constraint to add/remove from the working set
3. Handling degeneracy when multiple constraints have the same multiplier values
4. Detecting infeasibility and unboundedness

Test your implementation on the portfolio optimization problem with different risk-return preferences.