In [2]:

```
import numpy as np
from scipy.optimize import minimize

# Definierte Datenpunkte und Klassen
X = np.array([[1, 1], [-1, 1], [2, 1], [1, -2], [-2, 1]])  # A, B, C, D, E
y = np.array([1, 1, -1, -1, -1])  # Labels: +1 für A und B und -1 für C, D un
d E

# Polynomialer Kernel vom Grad 2: K(x_i, x_j) = (x_i * x_j + 1)^2
def polynomial_kernel(x_i, x_j, degree=2):
    return (np.dot(x_i, x_j) + 1) ** degree

# Kernel-Matrix berechnen
def compute_kernel_matrix(X, degree=2):
    n_samples = X.shape[0]
    K = np.zeros((n_samples, n_samples))
    for i in range(n_samples):
        for j in range(n_samples):
            K[i, j] = polynomial_kernel(X[i], X[j], degree)
    return K

# Berechne die Kernel-Matrix für die Datenpunkte
K = compute_kernel_matrix(X, degree=2)
print("Kernel-Matrix:\n", K)

# Lagrange-Funktion: L = Sum(alpha_i) - 1/2 * Sum(alpha_i * alpha_j * y_i * y
_j * K(x_i, x_j))
def lagrange_function(alpha, y, K):
    return np.sum(alpha) - 0.5 * np.sum(alpha * alpha[:, None] * y * y[:, Non
e] * K)

# Randbedingungen: Sum(alpha_i * y_i) = 0
def equality_constraint(alpha, y):
    return np.dot(alpha, y)

# Bounds für die Multiplikatoren alpha (alpha_i >= 0)
bounds = [(0, None) for _ in range(X.shape[0])]

# Anfangswerte für alpha
initial_alpha = np.zeros(X.shape[0])

# Optimierung der Lagrange-Funktion
result = minimize(lambda alpha: -lagrange_function(alpha, y, K), initial_alph
a,
                  constraints={'type': 'eq', 'fun': lambda alpha: equality_co
nstraint(alpha, y)},
                  bounds=bounds)

# Optimierte Lagrange-Multiplikatoren
optimal_alpha = result.x
print("Optimierte Lagrange-Multiplikatoren:\n", optimal_alpha)

# Ausgabe der Lagrange-Funktion
lagrange_value = lagrange_function(optimal_alpha, y, K)
print("Wert der Lagrange-Funktion:\n", lagrange_value)
```

```
Kernel-Matrix:
 [[ 9.  1. 16.  0.  0.]
 [ 1.  9.  0.  4. 16.]
 [16.  0. 36.  1.  4.]
 [ 0.  4.  1. 36.  9.]
 [ 0. 16.  4.  9. 36.]]
Optimierte Lagrange-Multiplikatoren:
 [0.13421372 0.14362519 0.11577962 0.05555319 0.10650611]
Wert der Lagrange-Funktion:
 0.27777769335516167
```