

时间复杂度证明

苏王捷 2300011075

March 14, 2025

1 Proof

算法实现见下图1，整个算法由两重循环组成，外循环进行 n 次，内循环在第 i 次外循环中进行 $i-1$ 次加法运算和乘法运算，及每次内循环中进行 4 次加法运算和乘法运算，共进行 $2 \times \sum_{i=1}^n (i-1+4) = n(n+7)$ 次加法运算和乘法运算，时间复杂度为 $O(n^2)$ 。

```
for (int i = n - 1; i >= 0; i--){
    double sumT = 0.0;
    for (int j = i + 1; j < n; j++){
        sumT += T(i, j) * x[j];
    }

    double sumS = 0.0;
    for (int j = i + 1; j < n; j++){
        sumS += S(i, j) * z[j];
    }

    double denominator = S(i, i) * T(i, i) - lambda;
    double numerator = b[i] - sumS - S(i, i) * sumT;
    x[i] = numerator / denominator;

    z[i] = T(i, i) * x[i] + sumT;
}
```

Figure 1: 算法实现部分