



勒让德实习报告

- 标准向前递推法
- 代码解释

勒让德实习报告

标准向前递推法

首先需要给出种子点: $P_{0,0} = 1$

然后是递推公式:

1. 对于每一行最后一个数: $P_{l,l-1} = f_1 \times \sin(\theta) \times P_{l-1,l-1}$;
2. 对于每一行倒数第二个数: $P_{l,l-1} = f_2 \times \cos(\theta) \times P_{l-1,l-1}$;
3. 对于本行其他的数: $P_{l,m} = f_3 \times (f_4 * \cos(\theta) * P_{l-1,m} - f_5 * P_{l-2,m})$;

其中:

$$f_1 = \sqrt{\frac{2l-1}{2l}};$$

$$f_2 = \sqrt{2l+1};$$

$$f_3 = \sqrt{\frac{2l+1}{(l-m)(l+m)}};$$

$$f_4 = \sqrt{2l-1};$$

$$f_5 = \sqrt{\frac{(l-m-1)(l+m-1)}{2l-3}}$$

代码解释

流程图

flowchart LR A[开始] --> B[计算前
三个值] B-->|推导|C[得到每一行
最后两个数]

C==>D[每一行
前面的数]

向前递推法核心算法

```
def jisuan_P(n, sita_rad):
    """计算Plm数组的值，并返回该数组"""
    Pln = list()
    for row in range(n+1):
        Pln.append([])
        for clumn in range(row+1):
            Pln[row].append(0)
    Pln[0][0] = 1
    Pln[1][0] = math.sqrt(3) * math.cos(sita_rad)
    Pln[1][1] = math.sqrt(3) * math.sin(sita_rad)
    for row in range(2, n+1):
        coefficient_1 = math.sqrt((2 * row + 1)/(2 * row))
        coefficient_2 = math.sqrt(2 * row + 1)
        Pln[row][row] = coefficient_1 * math.sin(sita_rad) * Pln[row-1][row-1]
        Pln[row][row-1] = coefficient_2 * math.cos(sita_rad) * Pln[row-1][row-1]
    for row in range(2, n+1):
        for clumn in range(0, row-1):
            coefficient_3 = math.sqrt((2 * row + 1) / (row - clumn) / (row + clumn))
            coefficient_4 = math.sqrt(2 * row - 1)
            coefficient_5 = math.sqrt((row - clumn - 1) * (row + clumn - 1) / (2 * row - 3))
            Pln[row][clumn] = coefficient_3 * (coefficient_4 * math.cos(sita_rad) * Pln[row - 1][clumn] +
            coefficient_5 * math.sin(sita_rad) * Pln[row - 1][clumn + 1])
    return Pln
```