

数值代数 第五次上机作业

姓名：樊泽羲 学号：2200010816 日期：2025年12月18日

1. 问题描述

本次上机作业包含两个主要部分：

- 算法设计**：设计并实现利用过关 Jacobi 方法 (Threshold Jacobi Method) 求解实对称矩阵全部特征值和特征向量的通用子程序。
- 数值实验**：利用上述程序，求解阶数 n 从 50 到 100 的实对称三对角矩阵 A 的全部特征值与特征向量。矩阵 A 定义如下：

$$A = \begin{bmatrix} 4 & 1 & & & \\ 1 & 4 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & 4 & 1 \\ & & & 1 & 4 \end{bmatrix} \in \mathbb{R}^{n \times n} \quad (1)$$

即 $A_{ii} = 4$, $A_{i,i+1} = A_{i+1,i} = 1$, 其余元素为 0。

2. 算法原理与推导

2.1 经典 Jacobi 方法

Jacobi 方法通过一系列平面旋转变换 (Givens 变换) 将对称矩阵 A 相似对角化。在每一步迭代中，选择非对角元素 $a_{pq} \neq 0$ ，构造平面旋转矩阵 $J(p, q, \theta)$ ：

$$J(p, q, \theta) = \begin{bmatrix} I & & & \\ & c & \cdots & s \\ & \vdots & I & \vdots \\ & -s & \cdots & c \\ & & & & I \end{bmatrix} \quad (2)$$

其中 $c = \cos \theta$, $s = \sin \theta$ 位于 (p, p) , (p, q) , (q, p) , (q, q) 位置。

令 $A' = J^T A J$, 为使 $a'_{pq} = 0$, 旋转角 θ 需满足：

$$\tan(2\theta) = \frac{2a_{pq}}{a_{qq} - a_{pp}} \quad (3)$$

令 $\tau = \cot(2\theta) = \frac{a_{qq} - a_{pp}}{2a_{pq}}$, 设 $t = \tan \theta$, 则有方程 $t^2 + 2\tau t - 1 = 0$ 。为了减小数值误差，取绝对值较小的解：

$$t = \frac{\text{sgn}(\tau)}{|\tau| + \sqrt{1 + \tau^2}} \quad (4)$$

进而得到 $c = \frac{1}{\sqrt{1+t^2}}$, $s = t \cdot c$ 。

2.2 过关 Jacobi 方法 (Threshold Strategy)

经典 Jacobi 方法每步需搜索绝对值最大的非对角元，耗时 $O(n^2)$ 。过关法采用循环扫描策略，按行或列顺序遍历所有非对角元 (p, q) ，但仅当 $|a_{pq}|$ 大于某个阈值 (Threshold) 时才进行旋转。

算法流程：

- 初始化特征向量矩阵 $V = I$ 。
- 计算初始非对角部分 Frobenius 范数 $S = \sqrt{\sum_{i \neq j} a_{ij}^2}$ 。
- 设定初始阈值 δ (例如 S/n)。
- 进行多轮扫描 (Sweep):
 - 在每一轮中，遍历所有 $p < q$ 。
 - 若 $|a_{pq}| \geq \delta$ ，执行旋转消去 a_{pq} ，更新 A 和 V 。
 - 若 $|a_{pq}| < \delta$ ，跳过。
- 每轮结束后降低阈值 (如 $\delta \leftarrow \delta/2$ 或根据剩余范数更新)。
- 当所有非对角元均满足收敛精度 (如 $\leq 10^{-10}$) 时停止。

3. 数值实验结果

实验环境基于 Python `numpy` 库。针对 $n = 50, 60, 70, 80, 90, 100$ 的矩阵进行了测试。收敛标准设定为非对角元范数 $\leq 10^{-10}$ 。

3.1 实验数据汇总

下表展示了不同维度下的运行时间、扫描次数 (Sweeps)、旋转次数 (Rotations) 以及误差分析指标。

| 维度 (n) | 时间 (s) | 扫描次数 | 旋转次数 | 是否收敛 | 最大特征值误差 | 残差范数 | 正交性误差 |
|------------|--------|------|-------|------|------------------------|------------------------|------------------------|
| 50 | 0.599 | 14 | 6187 | True | 5.33×10^{-14} | 2.25×10^{-13} | 4.64×10^{-14} |
| 60 | 1.102 | 15 | 10541 | True | 7.37×10^{-14} | 8.92×10^{-15} | 7.39×10^{-14} |
| 70 | 1.594 | 15 | 12111 | True | 7.55×10^{-14} | 1.88×10^{-12} | 9.25×10^{-14} |
| 80 | 2.805 | 16 | 18978 | True | 9.24×10^{-14} | 1.11×10^{-14} | 1.10×10^{-13} |
| 90 | 3.812 | 16 | 23700 | True | 1.04×10^{-13} | 1.42×10^{-14} | 1.45×10^{-13} |
| 100 | 5.446 | 16 | 29209 | True | 1.14×10^{-13} | 1.53×10^{-14} | 1.56×10^{-13} |

注：

- 最大特征值误差： $\max_i |\lambda_i^{(Jacobi)} - \lambda_i^{(Ref)}|$ ，参考值由 `numpy.linalg.eigh` 计算。
- 残差范数： $\|AV - V\Lambda\|_F / \|A\|_F$ 。
- 正交性误差： $\|V^T V - I\|_F$ 。

4. 结果分析

4.1 精度分析

从表中数据可以看出，算法具有极高的精度：

- 特征值精度**：最大绝对误差在 10^{-14} 量级，接近双精度浮点数的机器精度 ($\epsilon \approx 2.2 \times 10^{-16}$)，说明计算结果非常可靠。
- 正交性**：特征向量矩阵的正交性误差保持在 10^{-13} 量级，表明生成的特征向量 V 保持了良好的正交性质，未因多次旋转累积过大的舍入误差。

4.2 收敛性分析

- 扫描次数 (Sweeps)**：对于 $n = 50$ 到 100 ，算法均在 14 至 16 次扫描内收敛。这印证了 Jacobi 方法在后期具有二次收敛速度的理论性质。扫描次数并没有随着 n 的增加而显著增加，表现出良好的稳定性。
- 旋转次数 (Rotations)**：随着 n 的增加，总旋转次数显著增加。这是因为每轮扫描潜在的旋转对数量为 $n(n-1)/2$ 。

4.3 复杂度与性能分析

观察运行时间随 n 的变化：

- $n = 50$ 时，耗时约 0.60s。
- $n = 100$ 时，耗时约 5.45s。
- 维度增加 2 倍，时间增加了约 9 倍 ($5.45/0.60 \approx 9.08$)。

理论分析：

Jacobi 方法一次完整的扫描需要 $O(n^2)$ 次旋转，而每次旋转涉及行/列更新，计算量为 $O(n)$ 。因此一次扫描的复杂度为 $O(n^3)$ 。由于扫描次数基本恒定（约 15 次），总时间复杂度应为 $O(n^3)$ 。实验中， $2^3 = 8$ ，实际增长倍数约为 9，与 $O(n^3)$ 的理论复杂度基本吻合。

5. 结论

本次实验成功实现了利用过关 Jacobi 方法求解实对称矩阵特征值问题的通用程序。实验结果表明：

- 该算法能够正确求解 50-100 阶三对角矩阵的特征值与特征向量。
- 精度极高**：误差控制在 10^{-13} 至 10^{-14} 数量级。
- 收敛稳定**：扫描次数几乎不随矩阵规模变化（稳定在 15 次左右）。
- 计算效率**：运行时间符合 $O(n^3)$ 的增长规律。

该算法虽然在计算速度上不如 QR 迭代法或分治法（特别是针对三对角矩阵），但其逻辑简单、并行性好，且在求解高精度特征向量方面具有优势。