第五章 矩阵特征值计算 编程实验

> 孔瑞阳 计科91 2019010175

第五章上机题1：

用幂法求下列矩阵按模最大的特征值 λ_1 及其对应的特征向量 x_1，使 $|(\lambda_1)_{k+1} - (\lambda_1)_k| < 10^{-5}$.

$$(1)\ A = \begin{bmatrix} 5 & -4 & 1 \\ -4 & 6 & -4 \\ 1 & -4 & 7 \end{bmatrix}$$

$$(2)\ B = \begin{bmatrix} 25 & -41 & 10 & -6 \\ -41 & 68 & -17 & 10 \\ 10 & -17 & 5 & -3 \\ -6 & 10 & -3 & 2 \end{bmatrix}$$

思路:

使用C++编程实现书中 P155 的伪代码。

规定初始向量为 $(1, 0, 0, \cdots)$，每次迭代后按照书中方法规格化向量。

实验结果:

```
lambda1 = 12.254311
x1 = (-0.674021, 1.000000, -0.889557)

lambda1 = 98.521698
x1 = (-0.603972, 1.000000, -0.251135, 0.148953)
```

以上结果在 MATLAB 中使用以下语句验证成功：

```
a = [5 -4 1; -4 6 -4; 1 -4 7]
b = [25 -41 10 -6; -41 68 -17 10; 10 -17 5 -3; -6 10 -3 2]
[x, y] = eig(a)
[x, y] = eig(b)
```

结果分析:

结果的精确度较高，和 MATLAB 中直接得到的结果相差无几。

并且由于使用了向量规格化的方法，算法执行过程中没有出现溢出，成功完成。

第五章上机题3：

设 $A = \begin{bmatrix} 0.5 & 0.5 & 0.5 & 0.5 \\ 0.5 & 0.5 & -0.5 & -0.5 \\ 0.5 & -0.5 & 0.5 & -0.5 \\ 0.5 & -0.5 & -0.5 & 0.5 \end{bmatrix}$, 实现基本的 QR 算法, 观察矩阵序列收敛的情况, 然后解释观察到的现象.

思路:

QR 分解使用 householder 变换法，用 C++ 实现 P171 的伪代码。

基本的 QR 迭代算法用 C++ 实现 P174 的伪代码。

若 A 是拟上三角矩阵，则计算 A 的所有特征值。

由于算法可能进入死循环，所以设置最大迭代次数 1000。

实验结果:

```
Iteration does not end!
```

结果分析:

由于 A 本身是一个正交矩阵，当对 A 执行 QR 分解得到的结果是 $A = AI$ 或者 $A = (-A)(-I)$。

即 A 会在 A 和 $-A$ 之间来回震荡，导致不能收敛。

第五章上机题4：

采用带原点位移的 QR 算法计算 $A = \begin{bmatrix} 0.5 & 0.5 & 0.5 & 0.5 \\ 0.5 & 0.5 & -0.5 & -0.5 \\ 0.5 & -0.5 & 0.5 & -0.5 \\ 0.5 & -0.5 & -0.5 & 0.5 \end{bmatrix}$, 的特征值, 观察迭代过程的收敛情况, 与上机题 3 的实验结果做比较.

思路:

用 C++ 实现 P178 的伪代码。

先用 householder 变换将 A 转化为一个三对角矩阵，再进行迭代。

将矩阵 A 转化为三对角矩阵的过程如下：

共进行 $n-2$ 次变换，考虑第 k 次变换的 householder 矩阵构造：

$\sigma = \sqrt{\sum_{i=k+1}^{n} A_{i,k}^2}$

$$w_i = \begin{cases} 0, & i < k+1 \\ \sqrt{0.5 * (1 + \frac{|a_{i,i+1}|}{\sigma})}, & i = k+1 \\ \frac{a_{k,i}}{2\sigma w_{k+1}}, & i > k+1 \end{cases}$$

实验结果:

```
eigenvalue:
1.0000, 1.0000, -1.0000, 1.0000
```

结果分析:

使用带原点位移的 QR 算法能成功解决上机题 3 中用基本 QR 迭代算法不能求出的矩阵的特征值。

说明原点位移技术能让 QR 算法对更一般的矩阵都收敛。