

矩阵特征值计算

上机题1

题目要求矩阵的最大模长的特征值以及对应的特征向量，则使用课上所讲的幂法求解即可。即先初始化一个向量 x ，然后迭代计算 $x = Ax$ ，直到 x 收敛。最后计算 Ax 的模长即为特征值， x 即为特征向量。

运行代码 `python P1.py` 后，可以得到如下结果：

```
Eigenvalue of A: 12.254327364587589
Eigenvector of A: [ 0.67401588 -1.          0.88956596]
Correct Eigenvalue of A: 12.254315860794474
Correct Eigenvector of A: [-0.6740206   1.          -0.88955836]
Eigenvalue of B: 98.52160650737164
Eigenvector of B: [-0.60396543  1.          -0.25114387  0.1489578 ]
Correct Eigenvalue of B: 98.52169771010125
Correct Eigenvector of B: [ 0.60397234 -1.          0.25113513 -0.14895345]
```

其中 `Correct` 开头的结果为 `numpy` 库计算得到的正确结果。可以发现使用幂法计算得到的结果与正确结果非常接近。

上机题3

使用QR算法来求矩阵 A 的特征值，在每一轮迭代中，使用 `numpy` 中的QR分解将矩阵 A 分解为 $A = QR$ ，然后计算 RQ ，直到 RQ 收敛，在这里我们直接将迭代次数设为 100 次。最后 RQ 的对角线元素即为矩阵 A 的特征值。

运行代码 `python P3.py` 后，可以得到如下结果：

```
Correct Eigenvalues of A: [ 1. -1.  1.  1.]
Iteration 0 A:
[[ 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]]
Iteration 10 A:
[[ 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]]
Iteration 20 A:
[[ 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]]
Iteration 30 A:
[[ 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]]
Iteration 40 A:
[[ 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]]
Iteration 50 A:
[[ 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]]
Iteration 60 A:
[[ 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]]
Iteration 70 A:
[[ 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]]
Iteration 80 A:
[[ 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]]
Iteration 90 A:
[[ 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]]
Eigenvalues of A: [0.5 0.5 0.5 0.5]

```

其中第一行为使用 `numpy` 库计算得到的正确结果。之后每十轮迭代的结果都会输出，可以发现矩阵 A 不变，这是因为矩阵 A 是正交矩阵，所以QR分解后 $RQ = A$ ，迭代结果保持不变。不能收敛到上三角矩阵，故无法得到正确的特征值。

上机题4

类似于上机题3，不过在3中QR算法无法收敛，故使用原点位移技术。在每轮迭时，依次计算 $Q_k R_k = A_k - s_k I$, $A_{k+1} = R_k Q_k + s_k I$ ，直到 A_k 收敛，这里迭代 100 次。其中取 $s_k = A_k(n, n)$ 。最后 A_k 的对角线元素即为矩阵 A 的特征值。

运行代码 `python P4.py` 后，可以得到如下结果：

```

Correct Eigenvalues of A: [ 1. -1.  1.  1.]
Iteration 0 A:

```

```
[[-0.5          0.67082039 -0.43915503 -0.32732684]
[ 0.67082039  0.7          0.1963961   0.14638501]
[-0.43915503  0.1963961   0.87142857 -0.09583148]
[-0.32732684  0.14638501 -0.09583148  0.92857143]]
```

Iteration 10 A:

```
[[-1.00000000e+000 -2.25937782e-016 -3.31839599e-016  3.56079025e-016]
[-2.67022966e-135  1.00000000e+000 -3.07779994e-017  4.71171178e-018]
[-1.26289159e-135  1.03681031e-016  1.00000000e+000 -3.01579692e-017]
[ 1.65154094e-160  1.95331210e-042  3.45880034e-042  1.00000000e+000]]
```

Iteration 20 A:

```
[[-1.00000000e+000 -2.25937782e-016 -3.31839599e-016  3.56079025e-016]
[ 8.62873490e-301  1.00000000e+000 -3.07779994e-017  4.71171178e-018]
[ 4.08098110e-301  1.03681031e-016  1.00000000e+000 -3.01579692e-017]
[ 0.00000000e+000  0.00000000e+000  0.00000000e+000  1.00000000e+000]]
```

Iteration 30 A:

```
[[-1.00000000e+000 -2.25937782e-16 -3.31839599e-16  3.56079025e-16]
[ 0.00000000e+00  1.00000000e+00 -3.07779994e-17  4.71171178e-18]
[ 0.00000000e+00  1.03681031e-16  1.00000000e+00 -3.01579692e-17]
[ 0.00000000e+00  0.00000000e+00  0.00000000e+00  1.00000000e+00]]
```

Iteration 40 A:

```
[[-1.00000000e+000  2.25937782e-16  3.31839599e-16  3.56079025e-16]
[ 0.00000000e+00  1.00000000e+00 -3.07779994e-17 -4.71171178e-18]
[ 0.00000000e+00  1.03681031e-16  1.00000000e+00  3.01579692e-17]
[ 0.00000000e+00  0.00000000e+00  0.00000000e+00  1.00000000e+00]]
```

Iteration 50 A:

```
[[-1.00000000e+000 -2.25937782e-16 -3.31839599e-16  3.56079025e-16]
[ 0.00000000e+00  1.00000000e+00 -3.07779994e-17  4.71171178e-18]
[ 0.00000000e+00  1.03681031e-16  1.00000000e+00 -3.01579692e-17]
[ 0.00000000e+00  0.00000000e+00  0.00000000e+00  1.00000000e+00]]
```

Iteration 60 A:

```
[[-1.00000000e+000  2.25937782e-16  3.31839599e-16  3.56079025e-16]
[ 0.00000000e+00  1.00000000e+00 -3.07779994e-17 -4.71171178e-18]
[ 0.00000000e+00  1.03681031e-16  1.00000000e+00  3.01579692e-17]
[ 0.00000000e+00  0.00000000e+00  0.00000000e+00  1.00000000e+00]]
```

Iteration 70 A:

```
[[-1.00000000e+000 -2.25937782e-16 -3.31839599e-16  3.56079025e-16]
[ 0.00000000e+00  1.00000000e+00 -3.07779994e-17  4.71171178e-18]
[ 0.00000000e+00  1.03681031e-16  1.00000000e+00 -3.01579692e-17]
[ 0.00000000e+00  0.00000000e+00  0.00000000e+00  1.00000000e+00]]
```

Iteration 80 A:

```
[[-1.00000000e+000  2.25937782e-16  3.31839599e-16  3.56079025e-16]
[ 0.00000000e+00  1.00000000e+00 -3.07779994e-17 -4.71171178e-18]
[ 0.00000000e+00  1.03681031e-16  1.00000000e+00  3.01579692e-17]
[ 0.00000000e+00  0.00000000e+00  0.00000000e+00  1.00000000e+00]]
```

Iteration 90 A:

```
[[-1.00000000e+000 -2.25937782e-16 -3.31839599e-16  3.56079025e-16]
[ 0.00000000e+00  1.00000000e+00 -3.07779994e-17  4.71171178e-18]
[ 0.00000000e+00  1.03681031e-16  1.00000000e+00 -3.01579692e-17]
[ 0.00000000e+00  0.00000000e+00  0.00000000e+00  1.00000000e+00]]
```

```
Eigenvalues of A: [-1.  1.  1.  1.]
```

其中第一行为使用 `numpy` 库计算得到的正确结果。之后每十轮迭代的结果都会输出，可以发现矩阵 A 在迭代过程中逐渐收敛到上三角矩阵，最后取对角线元素得到正确的特征值。