homework 20241029

陈皓阳 $23307130004@m.fudan.edu.cn$

第一题

1、$(A+E)\hat{x} = \hat{x}\hat{\lambda}$

$\Leftrightarrow \quad E\hat{x} = \hat{x}\hat{\lambda} - A\hat{x} = -(A\hat{x} - \hat{x}\lambda) = -r$

$\Rightarrow \quad \|E\hat{x}\|_2 = \|-r\|_2 = \|r\|_2$ 无数、

构造 $E = -r\hat{x}^*$，满足 $E\hat{x} = -r\hat{x}^*\hat{x} = -r\|\hat{x}\|_2 = -r$

$\Rightarrow \|E\|_2 = \|-r\hat{x}^*\|_2 \le \|r\|_2 \cdot \|\hat{x}^*\|_2 = \|r\|_2$、

于是 $\exists E \in \mathbb{C}^{n\times n}$, s.t. $(A+E)\hat{x} = \hat{x}\lambda$、$\|E\|_2 \le \|r\|_2$.

第二题

2、用数学归纳法。当 $n=0$ 时，$A_0 - \mu_0 I = Q_0 R_0$ 满足。

当 $n=k$ 时，假设该成立 $(A_0-\mu_0 I)(A_0-\mu_1 I)\cdots(A_0-\mu_k I) = (Q_0\cdots Q_k)(R_k\cdots R_0)$、

将 A_1 视为迭代初值，则有 $(A_1-\mu_1 I)(A_1-\mu_2 I)\cdots(A_1-\mu_{k+1} I) = (Q_1\cdots Q_{k+1})(R_{k+1}\cdots R_1)$ ①

由迭代关系：$A_0 - \mu_0 I \overset{QR分解}{=} Q_0 R_0$，即 $A_0 = Q_0 R_0 + \mu_0 I$

(令) $A_1 = R_0 Q_0 + \mu_0 I$

发现 $(A_1 - \mu_0 I) = Q_0^* Q_0 R_0 Q_0 = Q_0^*(A_0 - \mu_0 I) Q_0$、

实际上，对 $\forall i = 0\ldots, k+1$，$(A_1 - \mu_i I) = Q_0^*(A_0 - \mu_i I) Q_0$.

故①：$Q_0^*(A_0-\mu_1 I)Q_0 Q_0^*(A_0-\mu_2 I)Q_0 \cdots Q_0^*(A_0-\mu_{k+1} I)Q_0 = (Q_1\cdots Q_{k+1})(R_{k+1}\cdots R_1)$、

即 $(A_0-\mu_1 I)\cdots(A_0-\mu_{k+1} I) = (Q_0\cdots Q_{k+1})(R_{k+1}\cdots R_1)Q_0^*$、 ②

又发现交换 A_0 与 $\mu_i I$ 的次序不影响矩阵运算，即 $(A_0-\mu_0 I)(②.\text{left}) = (②.\text{left})(A_0-\mu_0 I)$

故 $(A_0-\mu_0 I)(A_0-\mu_1 I)\cdots(A_0-\mu_{k+1} I) = (Q_0\cdots Q_{k+1})(R_{k+1}\cdots R_1)Q_0^*(A_0-\mu_0 I)$

当 $n=k+1$ 时.

$$= (Q_0\cdots Q_{k+1})(R_{k+1}\cdots R_1)Q_0^*(Q_0 R_0)$$

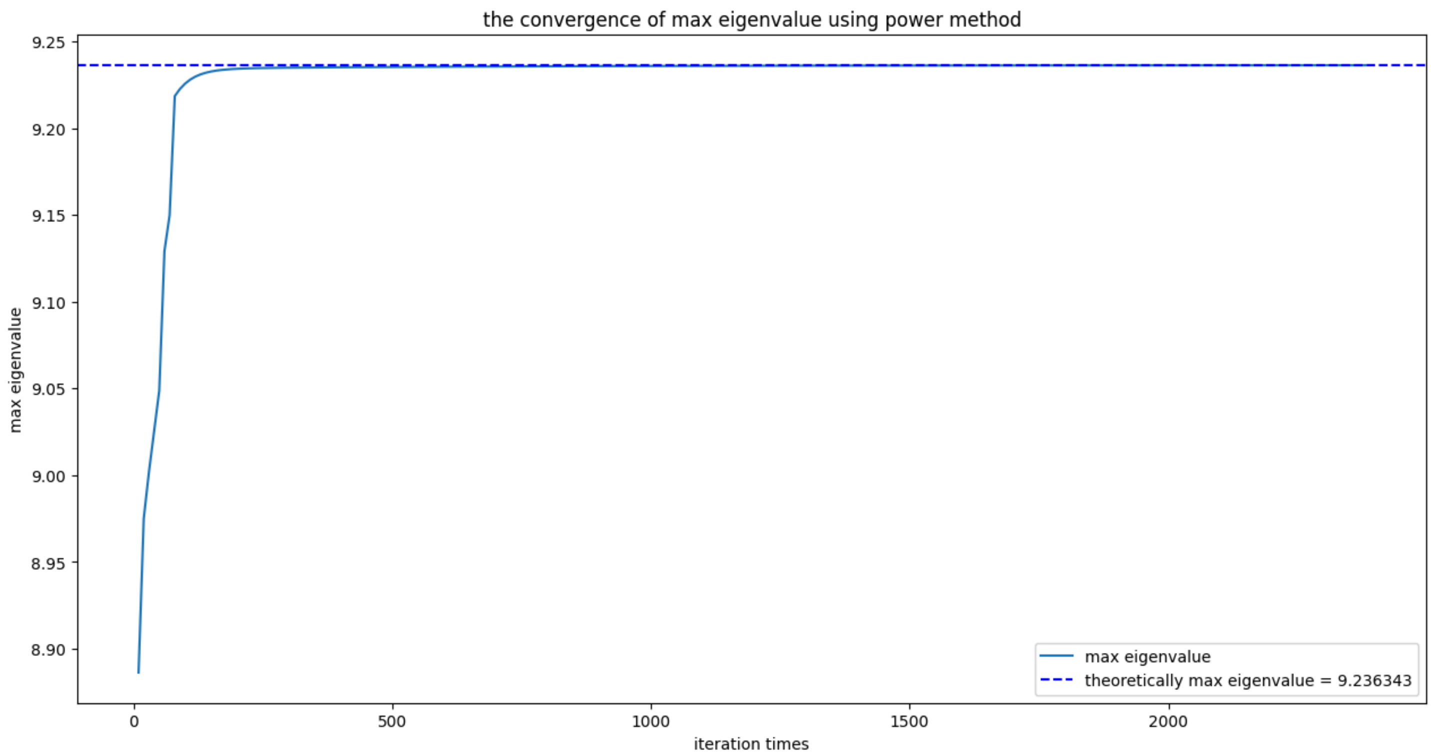
$$= (Q_0\cdots Q_{k+1})(R_{k+1}\cdots R_0)$$ 成立。

第三题

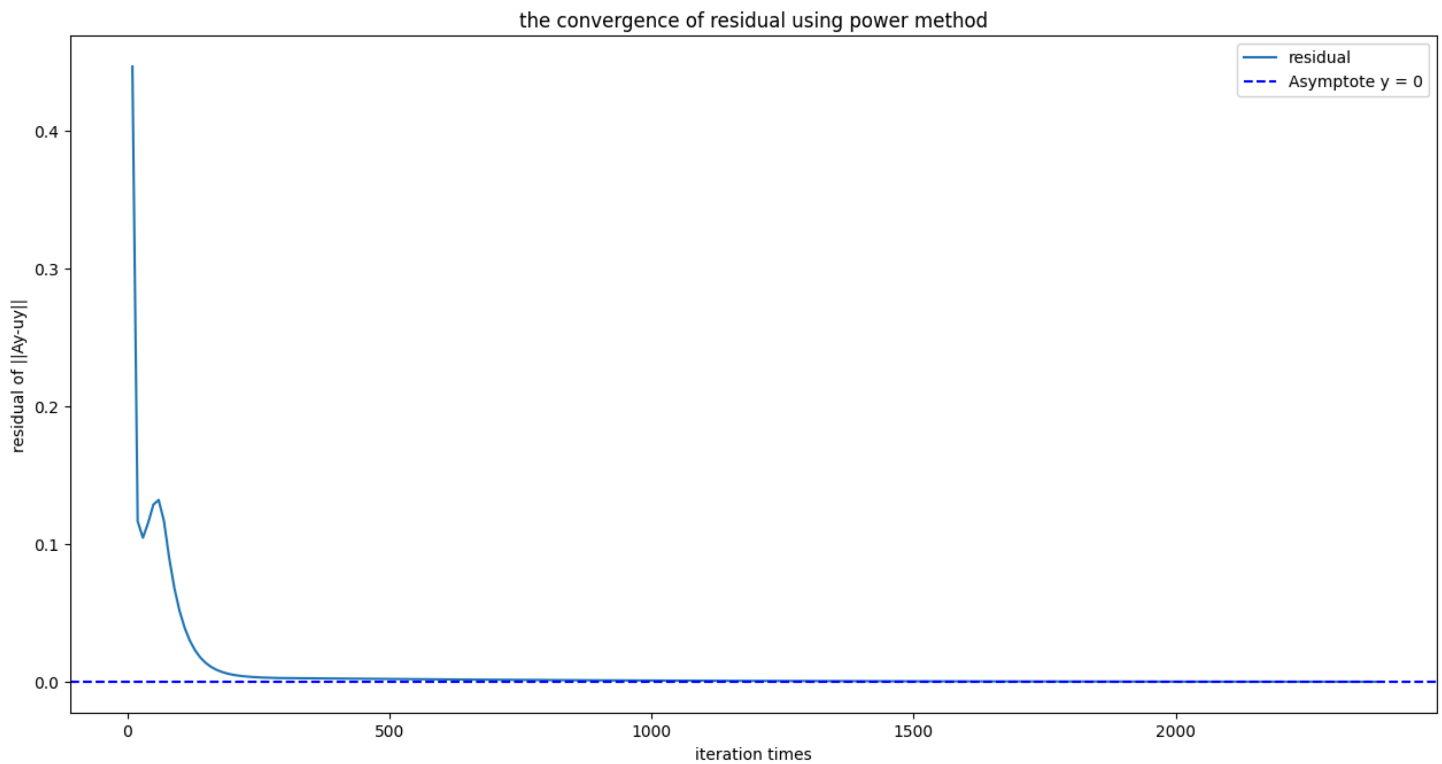
代码文件 **power_method.py**
迭代历史 **T3_iterhistory.txt**

每次运行代码随机生成 1000×1000 的矩阵（知道其最大特征值），用幂法计算其最大特征值，代码中幂法的停机条件是残差小于 10^{-16}；展示迭代过程时，记录了残差大于 10^{-4} 的情况，因为当残差过小时，在图像上就不会有明显变化

以下是某一次（选取了一次迭代过程较长的）运行产生的结果：



最大特征值



残差

```
iter:  1000 residual:  0.0009730924359242499 max eigenvalue:  9.235871937399645
iter:  2000 residual:  0.00018980821220171418 max eigenvalue:  9.236253158482034
iter:  3000 residual:  3.5553488817119394e-05 max eigenvalue:  9.236326028965271
iter:  4000 residual:  6.608862116674885e-06 max eigenvalue:  9.236339626479708
iter:  5000 residual:  1.2267400514298643e-06 max eigenvalue:  9.23634215225708
iter:  6000 residual:  2.276478420215610 8e-07 max eigenvalue:  9.236342621031127
iter:  7000 residual:  4.224285055087762e-08 max eigenvalue:  9.236342708020043
iter:  8000 residual:  7.838607651855511e-09 max eigenvalue:  9.236342724161833
iter:  9000 residual:  1.454534270806107e-09 max eigenvalue:  9.236342727157108
iter:  10000 residual:  2.699036549813627e-10 max eigenvalue:  9.236342727712913
iter:  11000 residual:  5.008349290847036e-11 max eigenvalue:  9.236342727816048
iter:  12000 residual:  9.293232849927335e-12 max eigenvalue:  9.236342727835186
iter:  13000 residual:  1.7248424910576432e-12 max eigenvalue:  9.236342727838737
iter:  14000 residual:  3.198552533945076e-13 max eigenvalue:  9.236342727839396
iter:  15000 residual:  5.906386491005833e-14 max eigenvalue:  9.236342727839519
iter:  16000 residual:  1.1213252548714081e-14 max eigenvalue:  9.23634272783954
iter:  17000 residual:  2.220446049250313e-15 max eigenvalue:  9.236342727839546
theoretically max eigenvalue:  9.236342727839544
max eigenvalue:  9.236342727839546
```

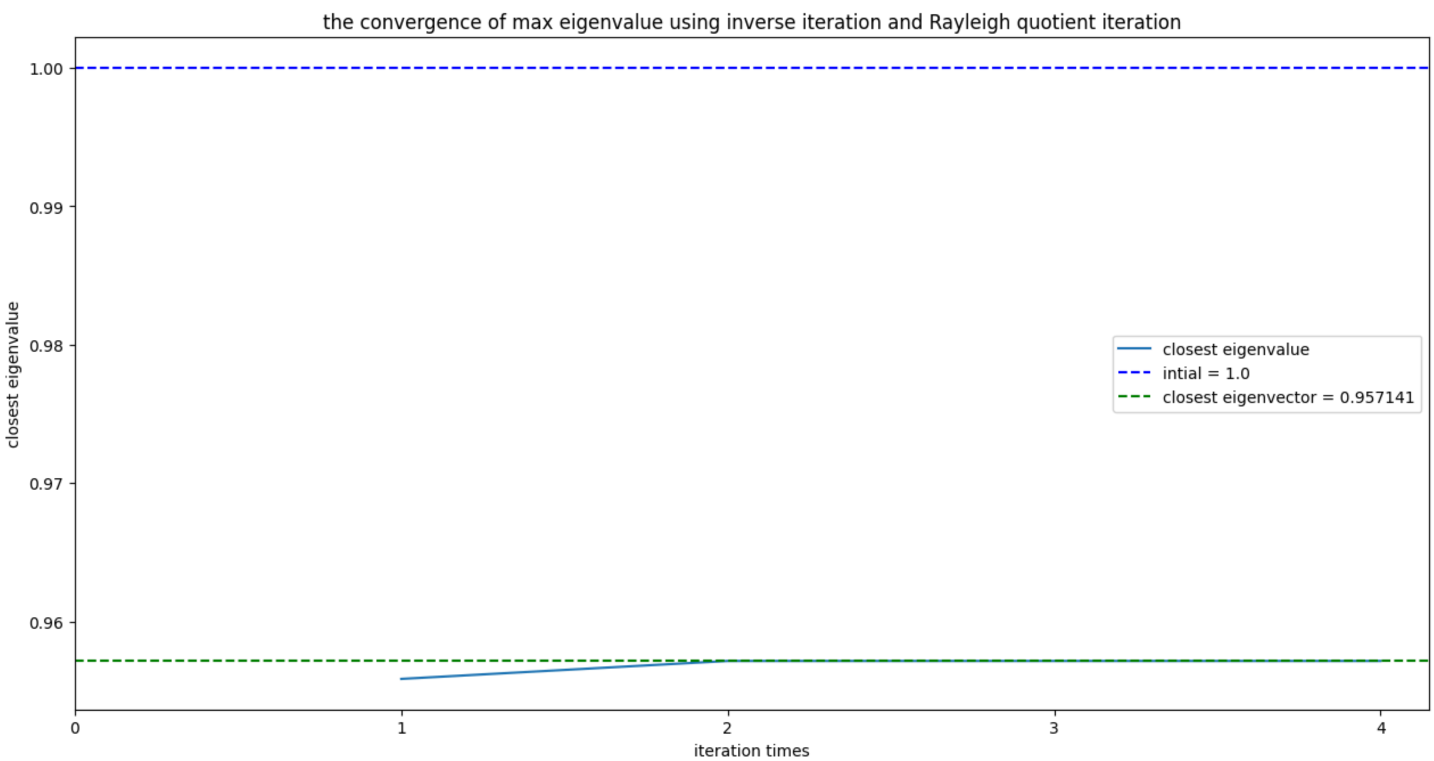
迭代历史（每一千次迭代记录一次）

第四题

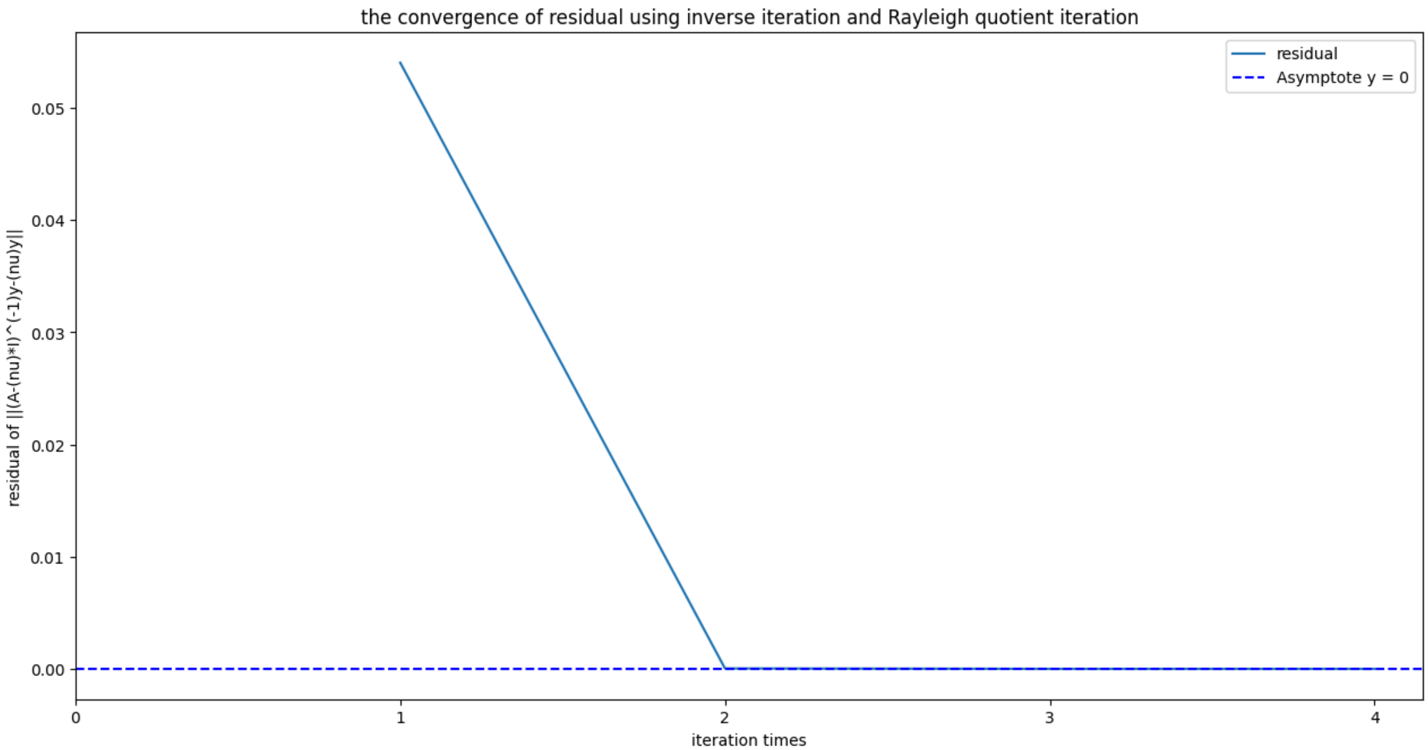
代码文件 **hilbert_inverse.py**
迭代历史 **T4_iterhistory.txt**

每次运行，虽然矩阵是固定不变的，但是迭代的向量是随机生成的，故每次迭代过程有所不同，但总体而言，结果显示 $200 \times 200 \ Hilbert\ matrix$ 最接近于 1 的特征值大约是 0.9571409212158626，运行时间大多数情况下可以达到 $0.01s$ 左右

虽然 $Hilbert\ matrix$ 是病态的，但是获得其最接近于 1 的特征值却较少受到其病态特征的影响，这是因为，虽然该矩阵的最小特征值在矩阵维数较大时无比接近于 0，这也是该矩阵病态的原因，但是由于该矩阵越靠近右下角，矩阵元素越小，这使得第一行第一列的元素 1 在矩阵中占据了“主导”地位。该病态矩阵特征值的分布在 $0+\epsilon$ 附近比较稠密，在 1 附近比较稀疏，于是迭代过程较快，用时较短



最接近于 1 的特征值



残差

```
iter:  1 residual:  0.05669425517040261 closest eigenvalue:  0.9553204275311244
iter:  2 residual:  4.817233871096116e-05 closest eigenvalue:  0.9571409069833385
iter:  3 residual:  1.7141843500212417e-12 closest eigenvalue:  0.9571409212158632
iter:  4 residual:  2.220446049250313e-16 closest eigenvalue:  0.9571409212158627
total time:  0.011999607086181164
target eigenvalue:  1.0
closest eigenvalue:  0.9571409212158627
```

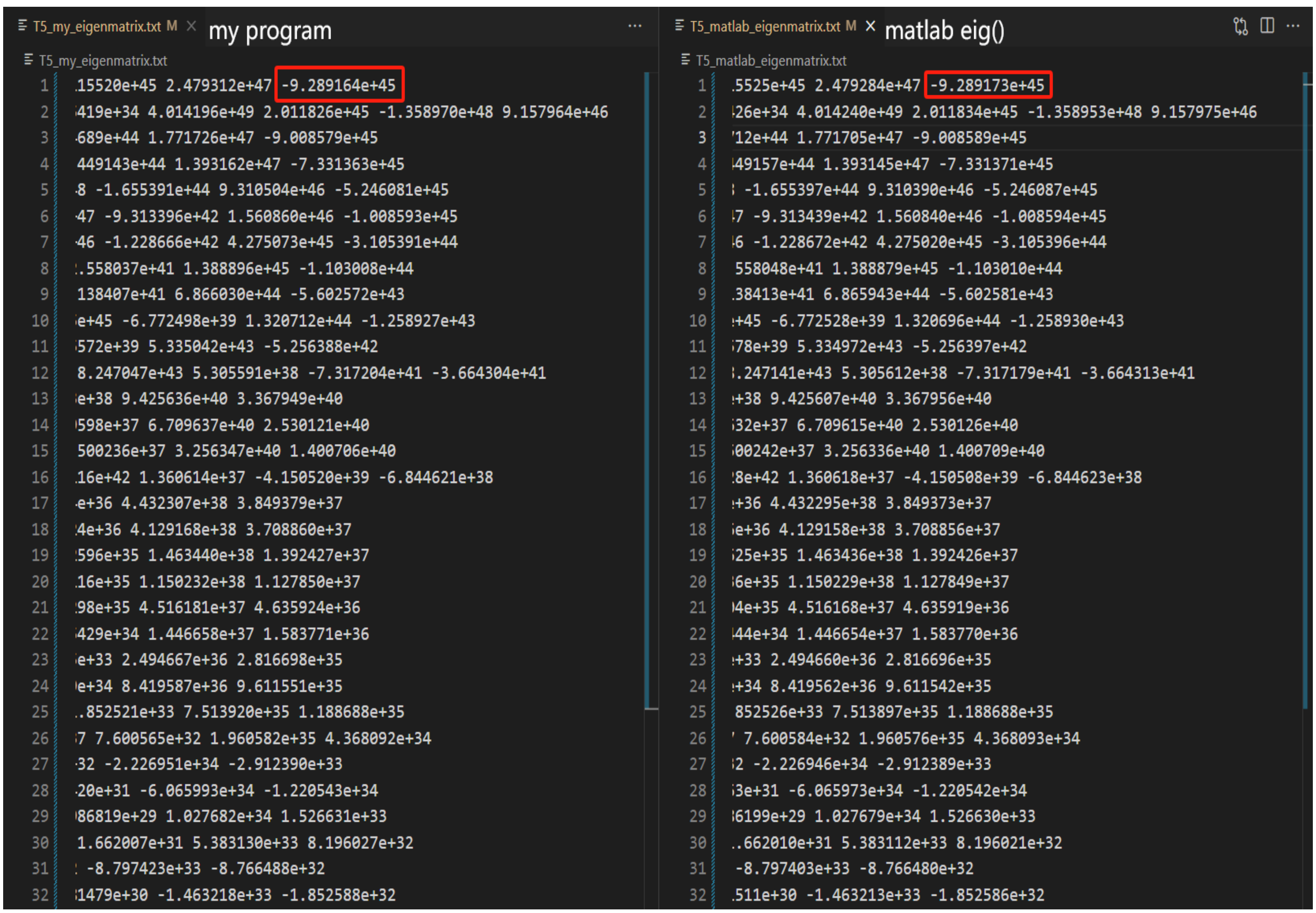
迭代历史（每一千次迭代记录一次）

第五题

$python$代码文件 **uptri_eigenmatrix.py**
输出文件 **T5_A.txt** 记录随机生成的矩阵
输出文件 **T5_my_eigenmatrix.txt** 记录用$python$求解的特征矩阵
$matlab$代码文件 **uptri_eigenmatrix.m** 读取 **T5_A.txt** 后使用自带函数 $eig()$
输出文件 **T5_matlab_eigenmatrix.txt** 记录了用$matlab$求解的特征矩阵

程序 **uptri_eigenmatrix.py** 计算特征向量的大致思路是，由于随机生成的上三角阵特征值就是对角元，于是分别用该矩阵减去各个对角元乘以单位阵，接着用回代法计算特征向量；在此过程中，若系数矩阵第 i 个对角元为 0 ，置特征向量第 i 个元素为 1，该元素之后的置 0，该元素之前的元素用回代法解线性方程组得出

随机生成一个 100×100 的矩阵。以下比较特征矩阵，左边是用上述算法得出的特征矩阵，右边是用 $matlab$ 求解的特征矩阵（已将对角元进行归一化）：



```
T5_my_eigenmatrix.txt M × my program
T5_my_eigenmatrix.txt
.15520e+45 2.479312e+47 -9.289164e+45
419e+34 4.014196e+49 2.011826e+45 -1.358970e+48 9.157964e+46
689e+44 1.771726e+47 -9.008579e+45
449143e+44 1.393162e+47 -7.331363e+45
8 -1.655391e+44 9.310504e+46 -5.246081e+45
47 -9.313396e+42 1.560860e+46 -1.008593e+45
46 -1.228666e+42 4.275073e+45 -3.105391e+44
.558037e+41 1.388896e+45 -1.103008e+44
138407e+41 6.866030e+44 -5.602572e+43
e+45 -6.772498e+39 1.320712e+44 -1.258927e+43
572e+39 5.335042e+43 -5.256388e+42
8.247047e+43 5.305591e+38 -7.317204e+41 -3.664304e+41
e+38 9.425636e+40 3.367949e+40
598e+37 6.709637e+40 2.530121e+40
500236e+37 3.256347e+40 1.400706e+40
.16e+42 1.360614e+37 -4.150520e+39 -6.844621e+38
e+36 4.432307e+38 3.849379e+37
4e+36 4.129168e+38 3.708860e+37
596e+35 1.463440e+38 1.392427e+37
.16e+35 1.150232e+38 1.127850e+37
98e+35 4.516181e+37 4.635924e+36
429e+34 1.446658e+37 1.583771e+36
e+33 2.494667e+36 2.816698e+35
e+34 8.419587e+36 9.611551e+35
.852521e+33 7.513920e+35 1.188688e+35
7 7.600565e+32 1.960582e+35 4.368092e+34
32 -2.226951e+34 -2.912390e+33
20e+31 -6.065993e+34 -1.220543e+34
86819e+29 1.027682e+34 1.526631e+33
1.662007e+31 5.383130e+33 8.196027e+32
-8.797423e+33 -8.766488e+32
1479e+30 -1.463218e+33 -1.852588e+32
```

```
T5_matlab_eigenmatrix.txt M × matlab eig()
T5_matlab_eigenmatrix.txt
.5525e+45 2.479284e+47 -9.289173e+45
26e+34 4.014240e+49 2.011834e+45 -1.358953e+48 9.157975e+46
12e+44 1.771705e+47 -9.008589e+45
49157e+44 1.393145e+47 -7.331371e+45
-1.655397e+44 9.310390e+46 -5.246087e+45
7 -9.313439e+42 1.560840e+46 -1.008594e+45
6 -1.228672e+42 4.275020e+45 -3.105396e+44
558048e+41 1.388879e+45 -1.103010e+44
.38413e+41 6.865943e+44 -5.602581e+43
+45 -6.772528e+39 1.320696e+44 -1.258930e+43
78e+39 5.334972e+43 -5.256397e+42
.247141e+43 5.305612e+38 -7.317179e+41 -3.664313e+41
+38 9.425607e+40 3.367956e+40
32e+37 6.709615e+40 2.530126e+40
00242e+37 3.256336e+40 1.400709e+40
8e+42 1.360618e+37 -4.150508e+39 -6.844623e+38
+36 4.432295e+38 3.849373e+37
e+36 4.129158e+38 3.708856e+37
25e+35 1.463436e+38 1.392426e+37
6e+35 1.150229e+38 1.127849e+37
4e+35 4.516168e+37 4.635919e+36
44e+34 1.446654e+37 1.583770e+36
+33 2.494660e+36 2.816696e+35
+34 8.419562e+36 9.611542e+35
852526e+33 7.513897e+35 1.188688e+35
7.600584e+32 1.960576e+35 4.368093e+34
2 -2.226946e+34 -2.912389e+33
3e+31 -6.065973e+34 -1.220542e+34
6199e+29 1.027679e+34 1.526630e+33
.662010e+31 5.383112e+33 8.196021e+32
-8.797403e+33 -8.766480e+32
.511e+30 -1.463213e+33 -1.852586e+32
```

红框圈出的元素是最后一个特征值对应的特征向量的第一个元素，选择该元素比较的原因是通过回代法得到它经历的步骤最多，可能是与理论结果偏差最大的元素。实际观察发现，该元素相对误差大概是 10^{-5}；绝对误差很大，因为该元素理论值已经达到 10^{45} 级别