Computional Homework15 Report

Student: 纪浩正, jihz2023@mail.sustech.edu.cn

1 Solve the Eigenvector and Eigenvalue Near $\lambda \approx 6.0$

The objective is to determine the eigenvalue of matrix A that is closest to the value 6.0. Two iterative schemes are employed: inverse iteration with a fixed shift, and an accelerated variant using the Rayleigh quotient. For a chosen shift σ, inverse iteration computes x_{k+1} from

$$(A - \sigma I)x_{k+1} = x_k,$$

leading to convergence toward the eigenvector associated with the eigenvalue nearest to σ. The corresponding eigenvalue estimate is obtained through

$$\lambda_{k+1} \approx \sigma + \frac{1}{\mu_k}, \qquad \mu_k = \frac{\|x_{k+1}\|}{\|x_k\|}.$$

When Rayleigh quotient acceleration is used, the shift is updated dynamically via

$$\rho(x) = \frac{x^\top Ax}{x^\top x},$$

which can significantly increase the convergence speed, particularly for symmetric matrices. However, an early introduction of the Rayleigh quotient may steer the iteration toward an unintended eigenvalue, while omitting it entirely may lead to stagnation. Both methods must therefore be combined to ensure robust and accurate computation.

1.1 Inverse Iteration with Shift

```
def _InverseShift(self,x0):
    A=self.A.copy()
    for ii in range(A.shape[0]):
        A[ii,ii]-=self.shift
    L,U=self._LUDecomposition(A)
    x=x0/np.linalg.norm(x0)
    start=time.time()
    for ii in range(self.N):
        x_new=self._solvex(x,L,U)
        miu=np.linalg.norm(x_new)/np.linalg.norm(x)
        x_new=x_new/np.linalg.norm(x_new)
```

```
            x=x_new
        eigenvalue=self.shift+1/miu
        return eigenvalue,x
```

1.2 Rayleigh Quotient Update

```
            if self.mode==1:
                y_new=self.A.dot(x_new)
                rayleigh=np.dot(x_new,y_new)/np.dot(x_new,x_new)
                x_new=y_new/miu
```

2 Results

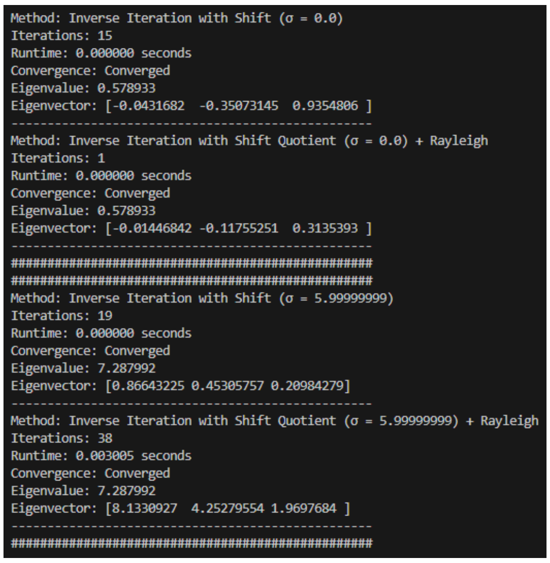
Two strategies are compared for combining inverse iteration and Rayleigh quotient:

- **Strategy 1:** Perform inverse iteration without the Rayleigh quotient to obtain a stable initial approximation, then refine the result using Rayleigh quotient iteration.
- **Strategy 2:** Use Rayleigh quotient iteration from the beginning to obtain a rapid approximation, followed by inverse iteration without Rayleigh to stabilize convergence.

- **Strategy 1 (Inverse iteration first, then Rayleigh):**
  - Eigenvalue near 0.578933 Eigenvector: $[-0.01446842, -0.11755251, 0.3135393]$
  - Eigenvalue near 7.287992 Eigenvector: $[8.1330927, 4.25279554, 1.9697684]$
  - The eigenvalue near 2.133047 is **not found**.
- **Strategy 2 (Rayleigh first, then inverse iteration):**
  - Eigenvalue near 0.578933 Eigenvector: $[-0.0431682, -0.35073145, 0.9354806]$
  - Eigenvalue near 2.133074 Eigenvector: $[-0.49742503, 0.8195891, 0.28432736]$
  - Eigenvalue near 7.287992 Eigenvector: $[0.86643225, 0.45305757, 0.20984279]$

2.1 Strategy 1: No Rayleigh First, Then Apply Rayleigh

Figure 1 shows that this approach recovers only **two eigenvalues**, near 0.58 and 7.29. For the shift $\sigma = 1$, inverse iteration alone fails to converge to the eigenvalue near 2.13, demonstrating that the fixed-shift method is sensitive to the choice of shift.



```
Method: Inverse Iteration with Shift (σ = 0.0)
Iterations: 15
Runtime: 0.000000 seconds
Convergence: Converged
Eigenvalue: 0.578933
Eigenvector: [-0.0431682  -0.35073145  0.9354806 ]
--------------------------------------------------
Method: Inverse Iteration with Shift Quotient (σ = 0.0) + Rayleigh
Iterations: 1
Runtime: 0.000000 seconds
Convergence: Converged
Eigenvalue: 0.578933
Eigenvector: [-0.01446842 -0.11755251  0.3135393 ]
--------------------------------------------------
##################################################
##################################################
Method: Inverse Iteration with Shift (σ = 5.99999999)
Iterations: 19
Runtime: 0.000000 seconds
Convergence: Converged
Eigenvalue: 7.287992
Eigenvector: [0.86643225 0.45305757 0.20984279]
--------------------------------------------------
Method: Inverse Iteration with Shift Quotient (σ = 5.99999999) + Rayleigh
Iterations: 38
Runtime: 0.003005 seconds
Convergence: Converged
Eigenvalue: 7.287992
Eigenvector: [8.1330927  4.25279554 1.9697684 ]
--------------------------------------------------
##################################################
```

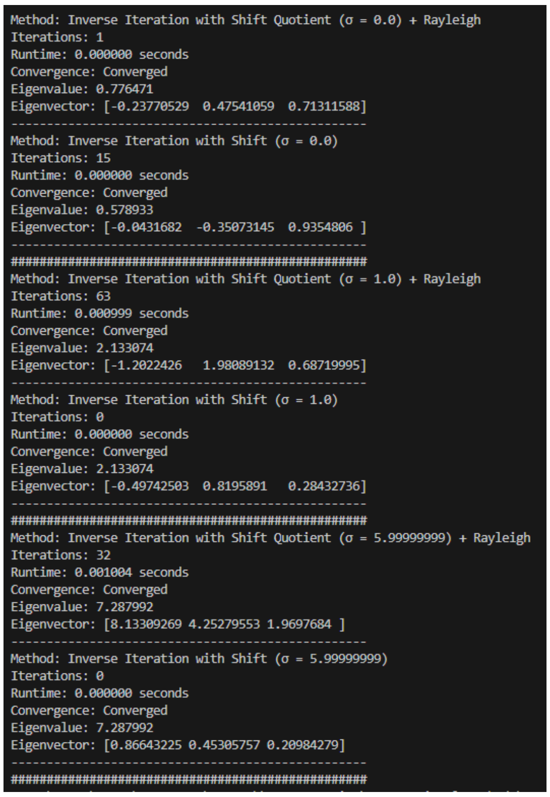
Figure 1 Results of Strategy 1: Inverse iteration without Rayleigh refinement, followed by Rayleigh quotient correction.

```python
A=np.array([[6.0,2,1],
           [2,3,1],
           [1,1,1]])
x0=np.array([-1,2,3])
EA=EigenvalueAlgorithms(A,tol=1e-8,N=1000)
shifts=np.array([0,1,6-1e-8])

for shift in shifts:
    eval,evec=EA.SloveEigenvalue(x0,shift=shift,mode=0)
    EA.SloveEigenvalue(evec,shift=shift,mode=1)
    print('#'*50)
print('\n\n')
```

2.2 Strategy 2: Rayleigh First, Then No Rayleigh

Figure 2 shows that this strategy successfully identifies **all three eigenvalues**: near 0.58, 2.13, and 7.29. The Rayleigh quotient provides a suitable shift update at each iteration, enabling convergence even when the initial shift is not close to the target eigenvalue. Inverse iteration without Rayleigh then stabilizes the final refinement.



```
Method: Inverse Iteration with Shift Quotient (σ = 0.0) + Rayleigh
Iterations: 1
Runtime: 0.000000 seconds
Convergence: Converged
Eigenvalue: 0.776471
Eigenvector: [-0.23770529  0.47541059  0.71311588]
--------------------------------------------------
Method: Inverse Iteration with Shift (σ = 0.0)
Iterations: 15
Runtime: 0.000000 seconds
Convergence: Converged
Eigenvalue: 0.578933
Eigenvector: [-0.0431682  -0.35073145  0.9354806 ]
--------------------------------------------------
##################################################
Method: Inverse Iteration with Shift Quotient (σ = 1.0) + Rayleigh
Iterations: 63
Runtime: 0.000999 seconds
Convergence: Converged
Eigenvalue: 2.133074
Eigenvector: [-1.2022426   1.98089132  0.68719995]
--------------------------------------------------
Method: Inverse Iteration with Shift (σ = 1.0)
Iterations: 0
Runtime: 0.000000 seconds
Convergence: Converged
Eigenvalue: 2.133074
Eigenvector: [-0.49742503  0.8195891   0.28432736]
--------------------------------------------------
##################################################
Method: Inverse Iteration with Shift Quotient (σ = 5.99999999) + Rayleigh
Iterations: 32
Runtime: 0.001004 seconds
Convergence: Converged
Eigenvalue: 7.287992
Eigenvector: [8.13309269 4.25279553 1.9697684 ]
--------------------------------------------------
Method: Inverse Iteration with Shift (σ = 5.99999999)
Iterations: 0
Runtime: 0.000000 seconds
Convergence: Converged
Eigenvalue: 7.287992
Eigenvector: [0.86643225 0.45305757 0.20984279]
--------------------------------------------------
##################################################
```

Figure 2 Results of Strategy 2: Rayleigh iteration for initialization, followed by inverse iteration without Rayleigh.

```python
for shift in shifts:
    eval,evec=EA.SloveEigenvalue(x0,shift=shift,mode=1)
    EA.SloveEigenvalue(evec,shift=shift,mode=0)
    print('#'*50)
```

2.3 Discussion

The behavior of the two strategies can be summarized as follows:

- Strategy 1 may fail when the shift is not close to the true eigenvalue, since fixed-shift inverse iteration requires accurate pre-positioning.
- Strategy 2 reliably converges for all shifts tested, although the Rayleigh quotient may temporarily steer the iteration toward an unintended eigenvalue when used alone.
- A hybrid approach is the most effective: Rayleigh quotient iteration provides rapid attraction toward a nearby eigenvalue, while inverse iteration without Rayleigh ensures a stable refinement.

The overall conclusion is that neither method alone is sufficient for consistent convergence. A combined approach is required to balance convergence speed and stability.