Scientific Computing: Homework 4

Harsh Bandhey 2017234

Problem 1: Power iteration

```
C:\Users\harsh\Documents\SCMP\HW4>python problem1.py
Regual Power Iteration
Eigenvalue= [[11.]]
 Eigenvector=
 [[0.37139068]
 [0.74278135]
 [0.55708601]]
Inverse Power Iteration
Eigenvalue= [[-2.]]
 Eigenvector=
 [[-0.18257419]
 [-0.36514837]
 [ 0.91287093]]
True Eigenvalues and Eigenvectors
Eigenvalues= [11. -2. -3.]
 Eigenvectors=
 [[ 3.71390676e-01  1.82574186e-01  2.17732649e-17]
 [ 7.42781353e-01  3.65148372e-01 -5.54700196e-01]
 [ 5.57086015e-01 -9.12870929e-01  8.32050294e-01]]
```

Regular Power Iteration gave the largest eigenvalue, Inverse Power Iteration gave the eigenvalue close to 0. Both methods were almost equal to that of the inbuilt method, including eigenvector with a sign inversion, which is non-consequential.

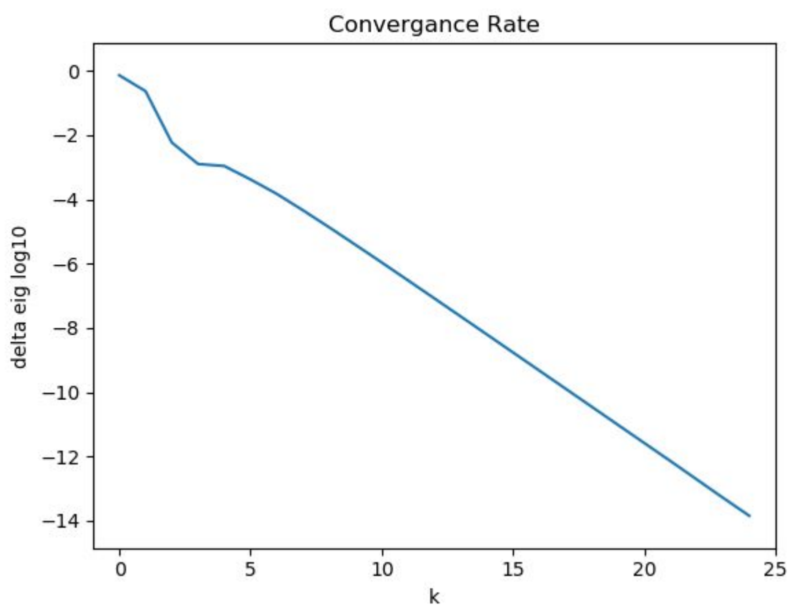
Problem 2: Shifted inverse iteration

```
C:\Users\harsh\Documents\SCMP\HW4>python problem2.py
Shifted Inverse Power Iteration
Eigenvalue= [[2.13307448]]
 Eigenvector=
 [[-0.49742502]
 [ 0.8195892 ]
 [ 0.28432708]]
True Eigenvalues and Eigenvectors
Eigenvalues= [7.28799214 2.13307448 0.57893339]
 Eigenvectors=
 [[ 0.86643225  0.45305757  0.20984279]
 [ 0.49742503 -0.8195891  -0.28432735]
 [-0.0431682  -0.35073145  0.9354806 ]]
```

Shifted Inverse Power Iteration gave the eigenvalue close to the supplied estimate of the eigenvalue. The eigenvalue is the same as that of the inbuilt method, the eigenvector is a little different but approximate of the inbuilt result with a sign inversion.

Problem 3: Rayleigh Quotient Iteration

```
C:\Users\harsh\Documents\SCMP\HW4>python problem3.py
True Eigenvalues and Eigenvectors
Eigenvalues= [11. -2. -3.]
 Eigenvectors=
 [[ 3.71390676e-01  1.82574186e-01  2.17732649e-17]
 [ 7.42781353e-01  3.65148372e-01 -5.54700196e-01]
 [ 5.57086015e-01 -9.12870929e-01  8.32050294e-01]]
Largest Eigenvalues= 10.999999999999993
 Largest Eigenvector=
 [0.37139068 0.74278135 0.55708601]
Rayleigh Quotient Iteration
Eigenvalue= 11.0
 Eigenvector=
 [[0.37139068]
 [0.74278135]
 [0.55708601]]
Rate of Convergence
Rate of convergence 0.0
```



Rayleigh quotient iteration gave the largest eigenvalue of the matrix. The eigenvalue almost approximate of that of the inbuilt method, the eigenvector is the same as that of the inbuilt result with a sign inversion.
The Rate of Convergence is 0 assuming $\lim_{inf}$ is the point of convergence.
Also given is a graph of log10(delta eigenvalue) vs iteration, the graph is mostly a straight line, thus inferring the convergence is of a logarithmic scale.

Problem 4: Modified QR Iteration

```
C:\Users\harsh\Documents\SCMP\HW4>python problem4.py


Applying on Matrix 1
[[ 2  3  2]
 [10  3  4]
 [ 3  6  1]]
Modified QR Iteration
Eigenvalues= [11.00000001 -3.00000001 -2.        ]
 Eigenvectors=
 [[-0.37139068 -0.01915301 -0.92827912]
 [-0.74278136 -0.59374335  0.30942637]
 [-0.557086    0.8044265   0.20628425]]
True Eigenvalues and Eigenvectors
Eigenvalues= [11. -2. -3.]
 Eigenvectors=
 [[ 3.71390676e-01  1.82574186e-01  2.17732649e-17]
 [ 7.42781353e-01  3.65148372e-01 -5.54700196e-01]
 [ 5.57086015e-01 -9.12870929e-01  8.32050294e-01]]


Applying on Matrix 2
[[6 2 1]
 [2 3 1]
 [1 1 1]]
Modified QR Iteration
Eigenvalues= [7.28798379 2.13308282 0.57893339]
 Eigenvectors=
 [[ 0.86706453 -0.49632208 -0.0431682 ]
 [ 0.45201426  0.82016496 -0.35073145]
 [ 0.20948081  0.28459415  0.9354806 ]]
True Eigenvalues and Eigenvectors
Eigenvalues= [7.28799214 2.13307448 0.57893339]
 Eigenvectors=
 [[ 0.86643225  0.49742503 -0.0431682 ]
 [ 0.45305757 -0.8195891  -0.35073145]
 [ 0.20984279 -0.28432735  0.9354806 ]]
```

For Both Matrices of Problem 1 and Problem 2, Modified QR Iteration gave close eigenvalue and eigenvector approximations to the ones calculated by the inbuilt method with minor deviation and sign inversion in a few cases in eigenvectors and almost approximate eigenvalues.