

Name: Suhas Kamath
SR No: 06-18-01-10-51-25-1-25945
Email ID: suhaskamath@iisc.ac.in
Date: October 26, 2025

Assignment No: Assignment 5 - Question 1
Course Code: DS284
Course Name: Numerical Linear Algebra
Term: AUG 2025

Preparation and Algorithm

The first step in analysing the graph is to obtain the Markov transition graph. The Markov transition matrix for the graph is as follows:

$$\mathbf{M} = \begin{pmatrix} 0 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.5 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0.5 & 0 \\ 0.5 & 0.5 & 0 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0.5 & 0 & 0 \\ 0 & 0.5 & 0 & 0 & 0.5 & 0 & 0 & 0.5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.5 & 0 & 0.5 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0 \end{pmatrix}$$

We need to run the following power iteration algorithm on the above Markov transition matrix:

```
Initialise a random vector  $\tilde{\mathbf{v}}^{(0)}$  such that  $\|\tilde{\mathbf{v}}^{(0)}\|_2 = 1$ 
for  $k = 1, 2, \dots$  do
     $\tilde{\mathbf{w}} = \mathbf{A}\tilde{\mathbf{v}}^{(k-1)}$ 
     $\tilde{\mathbf{v}}^{(k)} = \frac{\tilde{\mathbf{w}}}{\|\tilde{\mathbf{w}}\|_2}$ 
     $\lambda^{(k)} = (\tilde{\mathbf{v}}^{(k)})^T \mathbf{A}\tilde{\mathbf{v}}^{(k)}$ 
end for
```

The series of steps followed by the Python program is:

1. Build the Markov transition graph \mathbf{M} , using the nodes in the graph.
2. Run the power iteration for a maximum of 100 iterations. Stop the loop if the iteration converges before this. Convergence is measured by taking the 2-norm of the difference of the eigenvectors of successive iterates, i.e. $\|\mathbf{v}^{(k+1)} - \mathbf{v}^{(k)}\|_2$.¹
3. The highest and lowest PageRank and their respective nodes are identified and output.
4. The required plots are saved.

Results

Performing power iterations on this matrix, we obtain the following results:

Number of iterations required: 38
Highest PageRank: Node 5 (PageRank: 0.4685)
Lowest PageRank: Node 3 (PageRank: 0.0000)

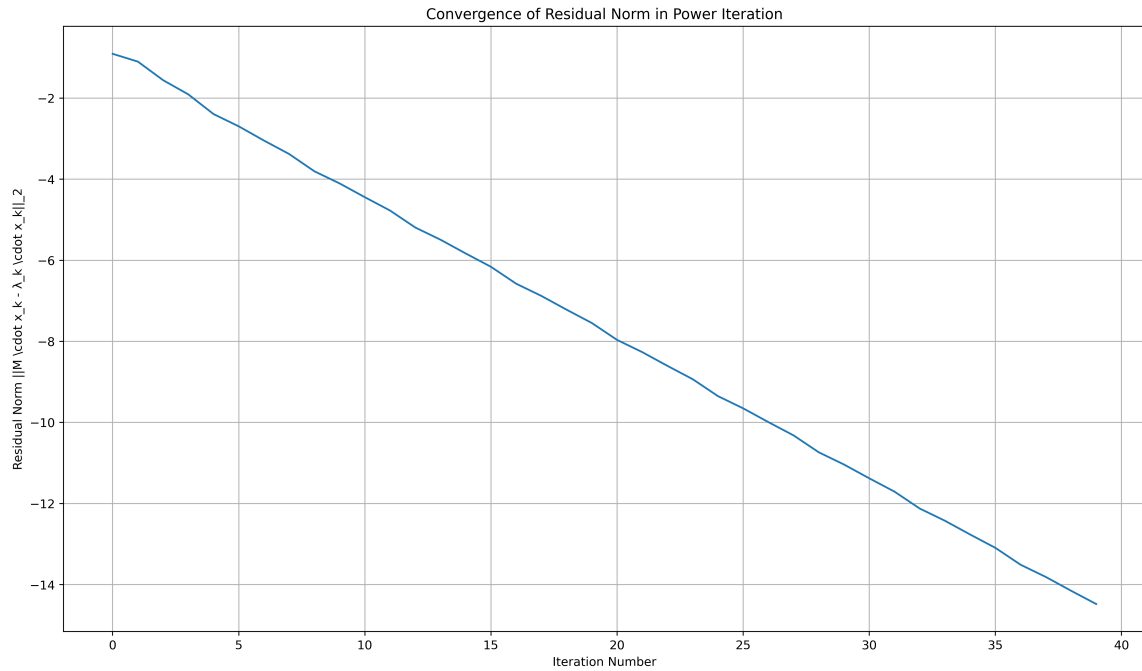
¹For this question, as no tolerance was specified, I decided to use 10^{-6} .

Name: Suhas Kamath
SR No: 06-18-01-10-51-25-1-25945
Email ID: suhaskamath@iisc.ac.in
Date: October 26, 2025

Assignment No: Assignment 5 - Question 1
Course Code: DS284
Course Name: Numerical Linear Algebra
Term: AUG 2025

Plots

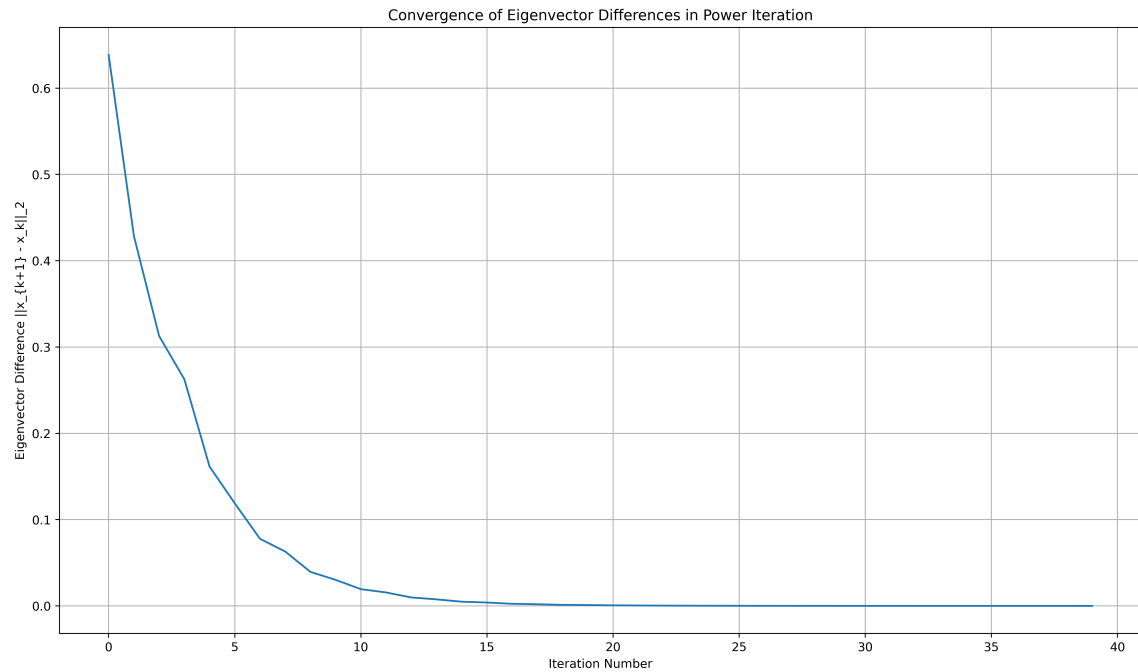
The log plot of the 2-norm of the residual of the eigenvalue problem, i.e. $\log \|\mathbf{M}\mathbf{x}^{(k+1)} - \lambda^{(k+1)}\mathbf{x}^{(k+1)}\|_2$, is as follows:



Name: Suhas Kamath
SR No: 06-18-01-10-51-25-1-25945
Email ID: suhaskamath@iisc.ac.in
Date: October 26, 2025

Assignment No: Assignment 5 - Question 1
Course Code: DS284
Course Name: Numerical Linear Algebra
Term: AUG 2025

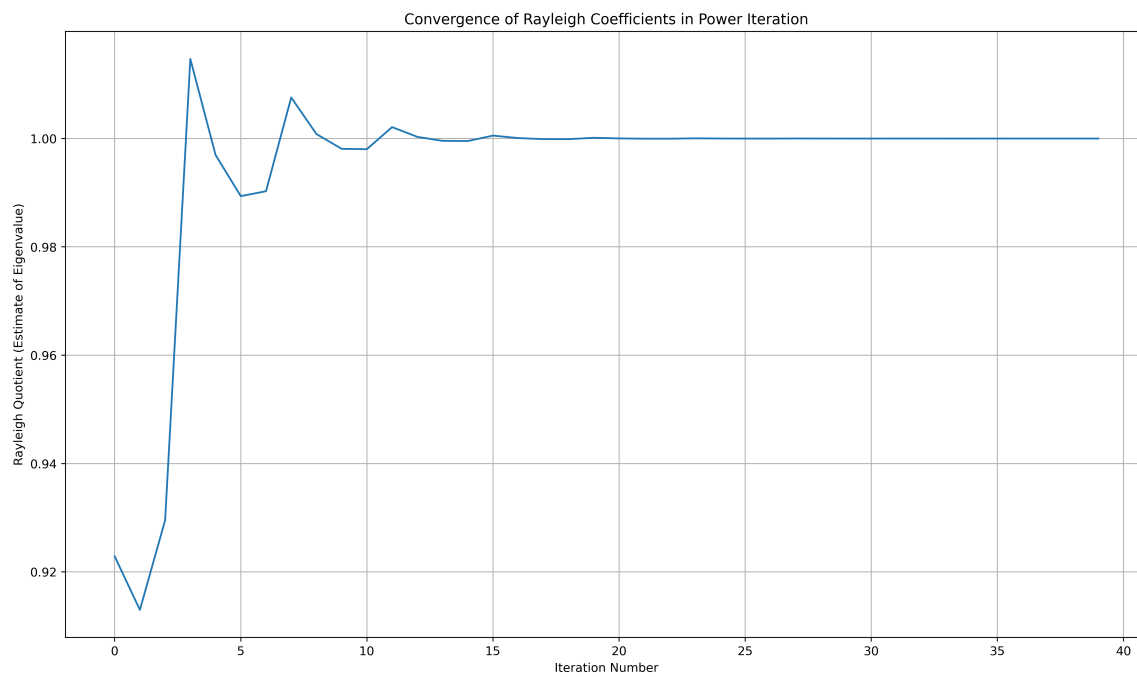
The 2-norm plot of the difference between the vectors of successive iterates, i.e. $\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\|_2$, is as follows:



Name: Suhas Kamath
SR No: 06-18-01-10-51-25-1-25945
Email ID: suhaskamath@iisc.ac.in
Date: October 26, 2025

Assignment No: Assignment 5 - Question 1
Course Code: DS284
Course Name: Numerical Linear Algebra
Term: AUG 2025

The Rayleigh quotient calculated at each stage of the iteration is as follows:



Name: Suhas Kamath
SR No: 06-18-01-10-51-25-1-25945
Email ID: suhaskamath@iisc.ac.in
Date: October 26, 2025

Assignment No: Assignment 5 - Question 1
Course Code: DS284
Course Name: Numerical Linear Algebra
Term: AUG 2025

Code

The Python code is as follows:

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 import os
4
5 # Multiple indexes may have the same max value
6 # In other words, a PageRank tie is possible
7 # Returns tuple: (array of max indexes, max value)
8 def find_max_indexes(vector):
9     max_value = np.max(vector)
10    max_indexes = np.where(vector == max_value)[0]
11    return (max_indexes, max_value)
12
13 # Multiple indexes may have the same min value
14 # In other words, a PageRank tie is possible
15 # Returns tuple: (array of min indexes, min value)
16 def find_min_indexes(vector):
17     min_value = np.min(vector)
18     min_indexes = np.where(vector == min_value)[0]
19     return (min_indexes, min_value)
20
21 # --- Build the Markov transition matrix M ---
22 M = np.zeros((10, 10))
23
24 # Outgoing links dictionary: for each node j, list nodes i that j links to.
25 # Interpretation: j -> i for each i in links[j].
26 links = {
27     0: [1, 2], 1: [2, 5], 2: [0, 1], 3: [4], 4: [5, 7],
28     5: [6, 9], 6: [2, 7], 7: [4, 5], 8: [1, 7], 9: [8]
29 }
30
31 # Fill M so each nonzero column j distributes mass uniformly over the
32 # outgoing links of j. For example, if node j points to k_j targets,
33 # each target i receives 1/k_j probability from column j.
34 for j, i_list in links.items():
35     k_j = len(i_list)
36     for i in i_list:
37         M[i, j] = 1.0 / k_j
38
39 # --- Power iteration parameters and initialization ---
40 max_iterations = 100          # hard cap to avoid infinite loops
41 tolerance = 1e-6             # stopping criterion on eigenvector change
42
43 # Random initial vector (nonzero), normalized to unit 2-norm.
44 x_k = np.random.rand(M.shape[0])
45 x_k /= np.linalg.norm(x_k)
46
47 # Containers to store convergence diagnostics for plotting/analysis
48 residuals = []               # ||M x_k - lambda_k x_k||_2
49 eigenvector_differences = []  # ||x_{k+1} - x_k||_2
50 eigenvalues = []             # Rayleigh quotient estimates
51
52 print("Starting power iteration...")
53
```

Name: Suhas Kamath
SR No: 06-18-01-10-51-25-1-25945
Email ID: suhaskamath@iisc.ac.in
Date: October 26, 2025

Assignment No: Assignment 5 - Question 1
Course Code: DS284
Course Name: Numerical Linear Algebra
Term: AUG 2025

```
54 # --- Power iteration loop ---
55 for k in range(max_iterations):
56     # Apply the matrix and renormalize to keep vector magnitude bounded
57     x_k_plus_1 = M @ x_k
58     x_k_plus_1 /= np.linalg.norm(x_k_plus_1)
59
60     # Rayleigh quotient provides an estimate of the eigenvalue for x_{k+1}
61     eigenval = x_k_plus_1.T @ M @ x_k_plus_1
62     eigenvalues.append(eigenval)
63
64     # Residual norm: measures how well (lambda, x) satisfies the eigen eqn
65     residuals.append(np.log(np.linalg.norm(M @ x_k_plus_1 - eigenval * x_k_plus_1)))
66
67     # Change in eigenvector between iterations (useful stopping metric)
68     eigenvector_differences.append(np.linalg.norm(x_k_plus_1 - x_k))
69
70     x_k = x_k_plus_1
71
72     # Check convergence
73     if eigenvector_differences[-1] < tolerance:
74         print(f"\nConverged after {k+1} iterations.")
75         break
76
77 # PageRank interpretation: normalize so components sum to 1 (probability)
78 pagerank_vector = x_k / np.linalg.norm(x_k)
79
80 for i in range(len(pagerank_vector)):
81     print(f"Node {i}: PageRank = {pagerank_vector[i]}")
82
83 print(f"Highest PageRank: Node(s) {find_max_indexes(pagerank_vector)[0]} (Rank:
84     ↳ {find_max_indexes(pagerank_vector)[1]:.4f})")
85
86 print(f"Lowest PageRank: Node(s) {find_min_indexes(pagerank_vector)[0]} (Rank:
87     ↳ {find_min_indexes(pagerank_vector)[1]:.4f})")
88
89 save_dir = r"H:\My Drive\Numerical Linear Algebra\Assignments\Assignment 5"
90 print(f"\nAttempting to save plots to: {save_dir}")
91
92 #Plot 1
93 plt.figure(figsize=(16,9))
94 plt.plot(residuals)
95 plt.title('Convergence of Residual Norm in Power Iteration')
96 plt.xlabel('Iteration Number')
97 plt.ylabel('Residual Norm ||M x_k - lambda_k x_k||_2')
98 plt.grid(True)
99 plt.savefig(os.path.join(save_dir, 'residual_norm.png'), dpi=400)
100
101 #Plot 2
102 plt.figure(figsize=(16,9))
103 plt.plot(eigenvector_differences)
104 plt.title('Convergence of Eigenvector Differences in Power Iteration')
105 plt.xlabel('Iteration Number')
106 plt.ylabel('Eigenvector Difference ||x_{k+1} - x_k||_2')
107 plt.grid(True)
108 plt.savefig(os.path.join(save_dir, 'eigenvector_differences.png'), dpi=400)
```

Name: Suhas Kamath
SR No: 06-18-01-10-51-25-1-25945
Email ID: suhaskamath@iisc.ac.in
Date: October 26, 2025

Assignment No: Assignment 5 - Question 1
Course Code: DS284
Course Name: Numerical Linear Algebra
Term: AUG 2025

```
108
109 #Plot 3
110 plt.figure(figsize=(16,9))
111 plt.plot(eigenvalues)
112 plt.title('Convergence of Rayleigh Coefficients in Power Iteration')
113 plt.xlabel('Iteration Number')
114 plt.ylabel('Rayleigh Quotient (Estimate of Eigenvalue)')
115 plt.grid(True)
116 plt.savefig(os.path.join(save_dir, 'rayleigh_coefficients.png'), dpi=400)
117
118 print("\nAll plots saved.")
```

Output

The output of the code is as follows:

```
1 Starting power iteration...
2
3 Converged after 38 iterations.
4 Node 0: PageRank = 0.23426049470201327
5 Node 1: PageRank = 0.4685210514425084
6 Node 2: PageRank = 0.46852134899268866
7 Node 3: PageRank = 0.0
8 Node 4: PageRank = 0.15617386821293897
9 Node 5: PageRank = 0.46852161969991823
10 Node 6: PageRank = 0.23426071515052027
11 Node 7: PageRank = 0.3123473886175061
12 Node 8: PageRank = 0.23426042938272038
13 Node 9: PageRank = 0.23426071515052027
14 Highest PageRank: Node(s) [5] (Rank: 0.4685)
15 Lowest PageRank: Node(s) [3] (Rank: 0.0000)
16
17 Attempting to save plots to: H:\My Drive\Numerical Linear
   ↳ Algebra\Assignments\Assignment 5
18
19 All plots saved.
```