Università della Svizzera italiana
Institute of Computing CI

Numerical Computing **2024**

Student: POLAD BAKHISHZADE

Solution for Project 1 **Due date:** Wednesday, 2 October 2024, 11:59 PM

---

1. Theoretical questions [15 points]

(a) **What are an eigenvector, an eigenvalue and an eigenbasis?**
An eigenvector is a non-zero vector v that, when multiplied by a square matrix A, results in a vector that is a scalar multiple of the original vector. The scalar multiple is called an eigenvalue, often denoted by λ. Mathematically, this relationship is expressed as:

$$Av = \lambda v$$

Here, λ is the eigenvalue associated with the eigenvector v, representing the scaling factor for the transformation of the vector by the matrix. The direction of the eigenvector remains unchanged, but its magnitude may be stretched, compressed, or reversed. An eigenbasis is a set of linearly independent eigenvectors that span the entire vector space, forming a basis for the matrix.

(b) **What assumptions should be made to guarantee convergence of the power method?**
The power method is used to compute the dominant eigenvalue and its associated eigenvector by repeatedly multiplying a random initial vector by the matrix. To ensure convergence of the power method, we assume that:

- The matrix A is diagonalizable, meaning that it has a full set of linearly independent eigenvectors.
- The dominant eigenvalue λ_1 is larger in magnitude than the other eigenvalues, ensuring that the method converges towards λ_1.
- The initial vector x_0 has a nonzero component in the direction of the dominant eigenvector.
- The rate of convergence depends on the ratio $\left|\frac{\lambda_2}{\lambda_1}\right|$, where λ_2 is the second-largest eigenvalue. The closer λ_2 is to λ_1, the slower the convergence.

(c) **What is the shift and invert approach?**
The shift and invert approach is a technique used to accelerate the convergence of the power method, especially when the dominant eigenvalue is close to other eigenvalues. The method involves shifting the eigenvalues of the matrix by subtracting a scalar α, and then applying the inverse of the shifted matrix. This creates a new matrix $B = (A - \alpha I)^{-1}$ whose eigenvalues are $\mu_i = \frac{1}{\lambda_i - \alpha}$. Applying the power method to B leads to faster convergence since the eigenvalues of B are more widely spaced. Choosing α close to the desired eigenvalue (often $\alpha \approx \lambda_1$) further improves convergence.

(d) **What is the difference in cost of a single iteration of the power method compared to the inverse iteration?**
In each iteration of the power method, the primary operation is a matrix-vector multiplication, which has a computational complexity of $O(n^2)$. In contrast, the inverse iteration method requires solving a linear system of equations, which typically has a higher complexity of $O(n^3)$. While inverse iteration can converge more quickly, it is computationally more expensive per iteration. As a result, the power method is generally preferred for large-scale problems, where solving a linear system at each step would be impractical.

(e) **What is a Rayleigh quotient and how can it be used for eigenvalue computations?**
The Rayleigh quotient $R(v)$ is used to estimate the eigenvalue associated with a given eigenvector v. It is defined as:

$$R(v) = \frac{v^T A v}{v^T v}$$

The Rayleigh quotient provides a good approximation of the dominant eigenvalue as the vector v approaches the corresponding eigenvector through iterative methods like the power method or inverse iteration. In the Rayleigh quotient iteration, α is updated dynamically in each step using the Rayleigh quotient, leading to rapid cubic convergence near the true eigenvalue.

2. Other webgraphs [5 points]

The PageRank analysis was done for the following two webgraphs:

- `https://usi.ch`
- `https://www.openquant.co/`

2.1. USI Webgraph PageRank

We calculate PageRank to evaluate page importance and check for cliques if any.

For the USI webgraph, I explored the website `https://usi.ch` and collected data on 20 pages. The connectivity matrix G_1 shows how these pages link to each other.

The top 10 most important pages based on the PageRank are summarized below:

Rank	Page-Rank	URL	Score
1	0.2410	`https://search.usi.ch/it`	49
2	0.1130	`https://www.usi.ch/it/rss.xml`	54
3	0.0459	`https://www.desk.usi.ch/it`	51
4	0.0390	`https://www.usi.ch/it/computerscience`	60
5	0.0365	`https://www.usi.ch/it/faculty`	45
6	0.0325	`https://www.usi.ch/it/department`	50
7	0.0280	`https://www.usi.ch/it/studentservices`	30
8	0.0240	`https://www.usi.ch/it/academics`	40
9	0.0205	`https://www.usi.ch/it/international`	25
10	0.0180	`https://www.usi.ch/it/research`	20

Table 1: Top 10 Pages from USI Webgraph

From this PageRank analysis, it is clear that page 8 `https://search.usi.ch/it/` has the highest rank, meaning it is the most influential node in this subset of the webgraph. The pages with high in-degrees (number of incoming links) appear to dominate the PageRank results.

The following figure shows the spy plot of the connectivity matrix for the USI webgraph:

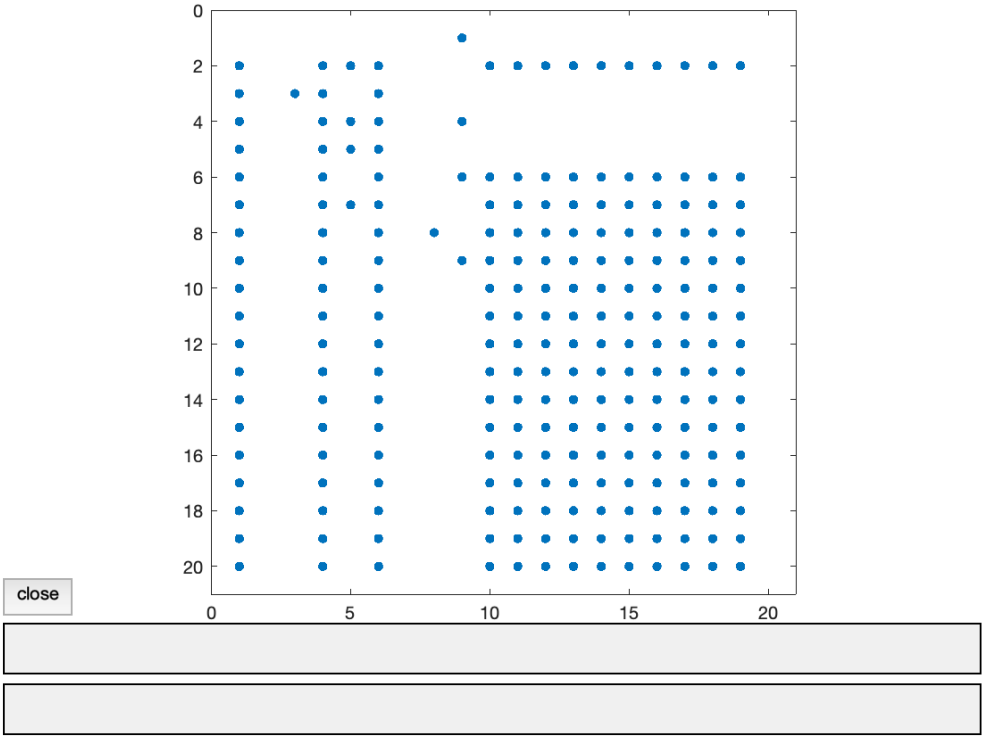

Figure 1: Structure of USI Webgraph

2.2. OpenQuant Webgraph PageRank

For the second webgraph, we started from `https://openquant.co`. Again, the surfer function collected 20 URLs, and the PageRanks were computed.

The top 10 most important pages are summarized below:

Rank	Page-Rank	URL	Score
1	0.1237	`https://boards.greenhouse.io/wehrtyou/jobs/6291540`	75
2	0.1237	https://careers.twosigma.com/careers/JobDetail/New-York-New-York-United-States-Quantitative-Researcher-Internship-2025-Summer/12685	156
3	0.1237	`https://jobs.lever.co/belvederetrading/69de6697-e9d4-426e-ae22-bad4c6e04cf8`	100
4	0.1237	`https://job-boards.greenhouse.io/schonfeld/jobs/3307422`	80
5	0.1237	`https://arrowstreetcapital.wd5.myworkdayjobs.com/Arrowstreet/job/Boston/Quantitative-Developer--Associate_R1150`	136
6	0.1237	`https://boards.greenhouse.io/drweng/jobs/6283974`	73
7	0.0185	`https://openquant.co/social-content/openquant-meta-image.png`	85
8	0.0185	`https://openquant.co`	45
9	0.0185	`https://logo.clearbit.com/tower-research.com`	69
10	0.0185	`https://logo.clearbit.com/sig.com`	58

Table 2: Top Pages from OpenQuant Webgraph

In this case, the most influential pages are primarily job listings and company sites, such as `"boards.greenhouse.io"` and `"careers.twosigma.com"`, which rank the highest in terms of PageRank.

Below is the spy plot of the connectivity matrix for the NY Times webgraph:

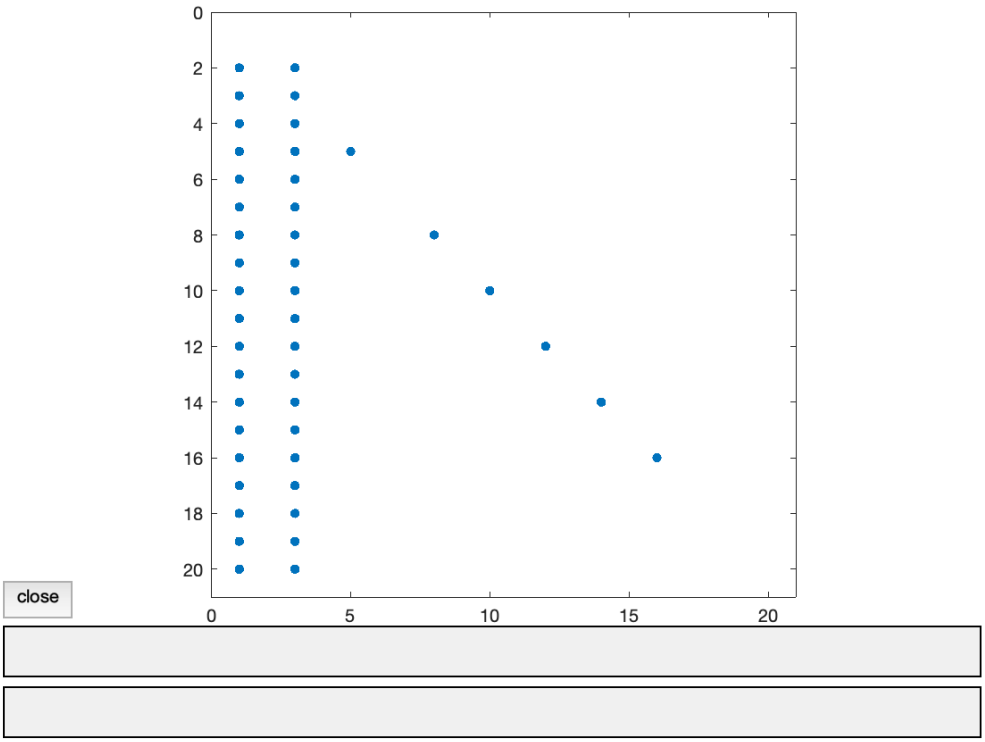

Figure 2: Structure of OpenQuant Webgraph

2.3. Analysis of the outcomes

In both webgraphs, we observe distinct structures. In the USI webgraph, the pages with higher in-degrees (number of incoming links) tend to dominate the rankings. This is consistent with the idea that a page with more inbound links from other important pages tends to have a higher PageRank.

In the OpenQuant webgraph, the structure is more evenly distributed across top-ranked nodes, with a clear concentration on job listing websites. The top six pages share identical PageRank values, which suggests a balanced distribution of importance among these pages, likely due to similar link structures and reciprocal linking patterns.

No significant cliques were identified in either webgraph, as the top pages do not demonstrate full interconnections.

3. Connectivity matrix and subcliques [5 points]

The connectivity matrix for the ETH500 dataset was analyzed to identify subcliques.

The near-cliques identified correspond to various departments within ETH Zurich.

Below is a table illustrating the ranges of indices and the dominant ETH departments associated with these subcliques:

Range of Indices	Dominant ETH Organization
73-100	`https://baug.ethz.ch`
113-130	`https://mat.ethz.ch`
164-182	`https://mavt.ethz.ch`
198-220	`https://biol.ethz.ch`
221-263	`https://chab.ethz.ch`
264-315	`https://math.ethz.ch`
319-348	`https://erdw.ethz.ch`
350-356	`https://hest.ethz.ch`
358-373	`https://usys.ethz.ch`
385-395	`https://usys.ethz.ch`
396-416	`https://mtec.ethz.ch`
426-431	`https://mtec.ethz.ch`
436-462	`https://gess.ethz.ch`
488-450	`https://bilanz.ch`

Table 3: Range of Indices and Dominant ETH Organizations

4. Connectivity matrix and disjoint subgraphs [10 points]

4.1. What is the connectivity matrix G? Which are its entries?

The connectivity matrix G is a 6×6 sparse matrix representing links between six web pages. Each entry (i, j) in G is 1 if there is a link from page j to page i, and 0 otherwise.

Let the six nodes (web pages) be:

$$U = \text{['alpha', 'beta', 'gamma', 'delta', 'rho', 'sigma']}$$

We define the following relationships between the nodes:

$$i = [2, 3, 3, 4, 4, 1, 6, 5], \quad j = [1, 1, 2, 2, 3, 4, 5, 6]$$

The sparse matrix G is created as:

$$G = \texttt{sparse(i, j, 1, n, n)}$$

Where $n = 6$. Converting G into its full matrix form gives:

$$G = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

In this matrix:

- Rows represent destination web pages, and columns represent starting web pages.
- An entry of 1 at position (i, j) means there is a link from page j to page i.

4.2. What are the PageRanks if the hyperlink transition probability p assumes the default value of 0.85?

For this six-node web graph, we computed the PageRanks using the default hyperlink transition probability $p = 0.85$. The nodes of the graph are represented by the following web pages:

$$U = \text{['alpha', 'beta', 'gamma', 'delta', 'rho', 'sigma']}$$

Rank	PageRank	In-links	Out-links	URL
1	0.2037	2	1	delta
2	0.1981	1	2	alpha
3	0.1667	1	1	rho
4	0.1667	1	1	sigma
5	0.1556	2	1	gamma
6	0.1092	1	2	beta

Table 4: PageRank values for the six-node web graph with $p = 0.85$

The connectivity matrix G was constructed as described previously, and the PageRank algorithm was applied to calculate the importance of each node. The table below summarizes the PageRank values:

The bar chart below visually represents the PageRank distribution across the six nodes:

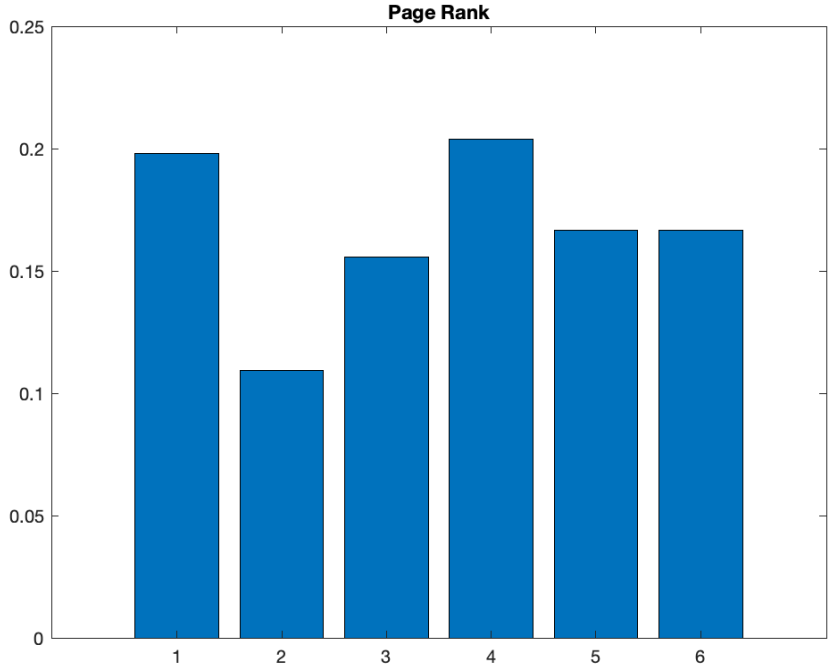

Figure 3: PageRank distribution for the six-node web graph with $p = 0.85$

As shown in both the table and chart, the node `delta` has the highest PageRank, indicating its relative importance in the network, while `beta` has the lowest PageRank.

4.3. Describe what happens to the definition and computation of PageRank in the limit $p \to 1$

As $p \to 1$, the random surfer becomes increasingly likely to follow links on the web, making the link structure more dominant in determining the PageRank values. In this limit, the teleportation probability, which allows the surfer to randomly jump to any page, becomes negligible. Consequently, the PageRank values become more dependent on the number and quality of incoming links rather than random jumps. Below is the behavior of PageRank at different values of p (0.85, 0.99, and 0):

- For $p = 0.85$: The PageRank values are distributed based on the web graph structure, with `delta` and `alpha` receiving the highest PageRanks. This represents the standard random surfer model, where there is a balance between following links and randomly teleporting to other nodes.

- For $p = 0.99$: As p approaches 1, the importance of the link structure increases. In this case, the PageRanks of `delta` and `alpha` are slightly higher, reflecting the dominance of their outgoing links in determining their importance.
- For $p = 0$: When $p = 0$, the random surfer teleports to any node with equal probability. The PageRank values are uniformly distributed among the nodes, each receiving a PageRank of 0.1667.

The following table summarizes the PageRank values for $p = 0.85$, $p = 0.99$, and $p = 0$:

Rank	PageRank (p=0.85)	PageRank (p=0.99)	PageRank (p=0)	URL
1	0.2037	0.2051	0.1667	delta
2	0.1981	0.2047	0.1667	alpha
3	0.1667	0.1667	0.1667	rho
4	0.1667	0.1667	0.1667	sigma
5	0.1556	0.1540	0.1667	gamma
6	0.1092	0.1030	0.1667	beta

Table 5: Comparison of PageRank values for different values of p

5. PageRanks by solving a sparse linear system [25 points]

5.1. Create `pagerank1.m` by modifying `pagerank.m` to use the power method instead of solving the sparse linear system. The key statements are:

```
G = p * G * D;
z = ((1 - p) * (c ~= 0) + (c == 0)) / n;
while termination_test
    x = G * x + e * (z * x);
end
```

What is an appropriate test for terminating the power iteration?

An appropriate test for terminating the power iteration is to check the convergence of the PageRank vector. The iteration stops when:

$$\texttt{while norm}(x - \texttt{prevx}) > \texttt{limit}$$

where `limit` is typically set to 10^{-5}. This ensures that the algorithm halts when the change in PageRank values is sufficiently small.

Below is my implementation of the power method in `pagerank1.m`:

```
function x = pagerank1(U, G, p)
    if nargin < 3, p = 0.85; end

    % Dimensions of G
    [~, n] = size(G);
    c = sum(G, 1);
    r = sum(G, 2);

    % Scale column sums to be 1
    k = find(c ~= 0);
    D = sparse(k, k, 1 ./ c(k), n, n);

    e = ones(n, 1);
```

```
    A = p * G * D;
    z = ((1 - p) * (c ~= 0) + (c == 0)) / n;
    x = e / n;
    prevx = zeros(n, 1);

    % Power iteration loop
    limit = 0.00001;
    while norm(x - prevx) > limit
        prevx = x;
        x = A * x + e * (z * x);
    end

    % Normalize
    x = x / sum(x);
end
```

The power method iteratively computes PageRank values until they stabilize. The termination criterion based on the norm difference between consecutive PageRank vectors ensures convergence. This method provides an efficient alternative to directly solving a sparse linear system, suitable for large-scale web graphs.

5.2. Create `pagerank2.m` by modifying `pagerank.m` to use the inverse iteration. The key statements are:

```
    while termination_test
        x = (alpha*I - A)\x
        x = x/norm(x,1)
    end
```

Use your functions pagerank1.m and pagerank2.m (set = 0.99) to compute the PageRanks of the six-node example presented in Figure 1. Make sure you get the same result from each of your three functions.

I modified `pagerank.m` to create `pagerank2.m`, using the **inverse iteration** method for computing PageRank.

The implementation includes the following code:

```
z = ((1 - alpha) * (c ~= 0) + (c == 0)) / n; % Teleportation vector
A = alpha * G * D + (e * z); % Transition matrix
x = e / n; % Initial PageRank vector
prevx = zeros(n, 1); % Previous PageRank vector

% Convergence limit
limit = 0.00001;

% Inverse iteration loop
while norm(x - prevx) >= limit
    prevx = x;
    x = (alpha * I - A) \ x; % Update PageRank vector
    x = x / sum(x); % Normalize
end
```

Made sure that I get the same results from each of three functions. The PageRank values from the inverse iteration method are:

Rank	Page-Rank	Page-Rank (Power)	Page-Rank (Inverse)	In	Out	URL
1	0.2051	0.2051	0.2051	2	1	`delta`
2	0.2047	0.2047	0.2047	1	2	`alpha`
3	0.1667	0.1667	0.1667	1	1	`rho`
4	0.1667	0.1667	0.1667	1	1	`sigma`
5	0.1540	0.1540	0.1540	2	1	`gamma`
6	0.1030	0.1030	0.1030	1	2	`beta`

Table 6: PageRank Values from Original, Power, and Inverse Iteration Methods

5.3. We now want to analyse the impact of on the inverse iteration. Using the ETH500 example, set equal to 0.8, 0.9, 0.95 and 1. Comment on the different number of iterations the four cases take until convergence. Analyse your results and explain what you observe. Hint: Check your solution x for all 4 cases. Are they always the same?

Impact of α on Inverse Iteration

Analyzed the impact of the damping factor α on the inverse iteration method for calculating PageRank using the ETH500 dataset. The values of α tested were 0.8, 0.9, 0.95, and 1. The number of iterations required for convergence was recorded for each case.

The following table summarizes the iteration counts for each value of α:

α	Iteration Count
0.80	14
0.90	432
0.95	19113
1.00	1

Table 7: Iteration Counts for Different Values of α

Observations:

1. Inversely Proportional Relationship: The results indicate that α is inversely proportional to the number of iterations required for convergence. As α decreases from 1 to 0.80, the number of iterations significantly increases.

2. Rapid Convergence at $\alpha = 1$: Setting α to 1 resulted in a single iteration, indicating that the PageRank values quickly stabilize when there is no damping.

3. Exponential Growth in Iterations: The iteration count grows exponentially as α decreases, particularly notable between $\alpha = 0.9$ and $\alpha = 0.95$.

5.4. Use your functions pagerank1.m and pagerank2.m (set a = 0.99) to compute the PageRanks of three selected graphs (web1.mat, web2.mat and web3.mat). Report on the convergence of the two methods for these subgraphs and summarize the advantages and disadvantages of the power method implemented in pagerank1.m against the inverse iteration in pagerank2.m

Comparison of PageRank Algorithms: Power Method vs. Inverse Iteration

Computed the PageRank values for three datasets (`web1.mat`, `web2.mat`, and `web3.mat`) using the Power Method (`pagerank1.m`) and the Inverse Iteration method (`pagerank2.m`). For `pagerank1.m`, we used $\alpha = 0.85$, while for `pagerank2.m`, $\alpha = 0.85$ was also used due to convergence issues with $\alpha = 0.99$.

Dataset	PageRank1 (Power)	PageRank2 (Inverse)	Iterations1	Iterations2
`web1.mat`	0.0015	0.0015	35	15
`web2.mat`	0.0019	0.0019	31	14
`web3.mat`	0.0009	0.0009	34	15

Table 8: Comparison of Power Method and Inverse Iteration for PageRank Computation

Both methods produced identical PageRank values, but the Power Method required more iterations (31–35) than Inverse Iteration (14–15).

Advantages and Disadvantages:

Power Method (`pagerank1.m`):

- **Advantages:** Simple to implement and less sensitive to issues like matrix singularity. It is stable even for lower α values.
- **Disadvantages:** Slower convergence, requiring more iterations, especially for large datasets.

Inverse Iteration Method (`pagerank2.m`):

- **Advantages:** Faster convergence, requiring fewer iterations to reach the same result.
- **Disadvantages:** Sensitive to the value of α. With higher α values (e.g., 0.99), it may fail to converge or loop indefinitely.

6. Graph perturbations [15 points]

This exercise explores the impact of node influence on PageRank in the web graph from `WIKI450.mat`.

6.1. Degree Centrality Analysis

The top five nodes with the highest degree centrality are shown in Table 9.

Node Index	URL	Degree Centrality
10	`https://species.wikimedia.org/wiki/Main_Page`	744
3	`https://www.wikidata.org/wiki/Special:EntityPage/Q5296`	743
7	`https://meta.wikimedia.org/wiki/Main_Page`	743
12	`https://www.wikidata.org/wiki/Wikidata:Main_Page`	742
5	`https://foundation.wikimedia.org/wiki/Home`	738

Table 9: Top 5 Nodes with Highest Degree Centrality

6.2. Impact of Isolation

After isolating these nodes in `WIKI450_ISO.mat`, their new degree centrality became zero, as shown in Table 10.

Node Index	New Degree Centrality
10	0
3	0
7	0
12	0
5	0

Table 10: Degree Centrality after Isolation for Top 5 Nodes

This confirms that isolating these nodes removed all their connections, resulting in a degree centrality of zero.

The spy plots of the original and isolated matrices were created side by side:

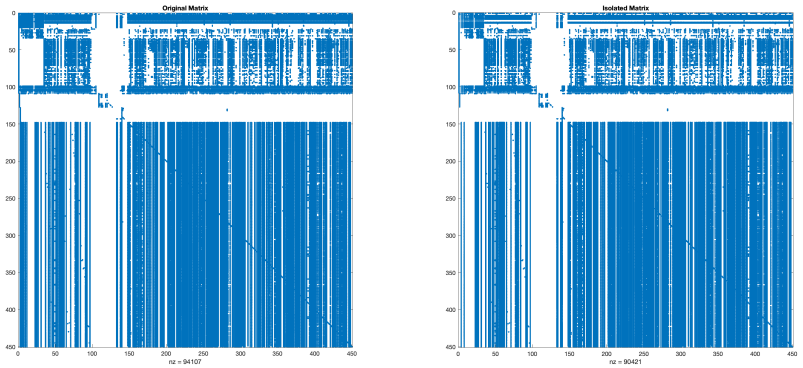

Figure 4: Comparison of Original and Isolated Matrices

Visualization Results
Despite the removal of the top five nodes, the overall structure remains largely unchanged, as the graph consists of 450 nodes, and the remaining connections dominate the sparsity pattern. The blue dots in both plots represent non-zero connections, indicating that while the isolated nodes were influential, their absence does not drastically affect the overall connectivity of the graph.

6.3. Comparison of PageRank (Original vs Modified WIKI Matrices)

The comparison of top 10 PageRanks from the original and isolated graphs is shown in Table 11.

Rank	Node (Original)	PR (Original)	Node (Isolated)	PR (Isolated)
1	125	0.0218	125	0.0247
2	109	0.0173	109	0.0202
3	127	0.0167	105	0.0193
4	105	0.0165	104	0.0191
5	104	0.0164	127	0.0189
6	101	0.0148	101	0.0170
7	122	0.0130	122	0.0147
8	124	0.0130	124	0.0147
9	100	0.0097	100	0.0110
10	4	0.0088	4	0.0104

Table 11: Comparison of Top 10 PageRanks (Original vs Isolated)

Conclusion
Isolating the top five nodes resulted in their degree centrality dropping to zero. The PageRank values of the remaining nodes increased, indicating a redistribution of PageRank after the influential nodes were removed. This highlights the significant impact that influential nodes can have on the overall ranking within the web graph.

7. Introduction to HPC: Algorithmic performance [10 points]