Understanding Financial Market Dynamics through Linear Algebra

Arya Bhakta (B24BS1067)

May 2025

Abstract

We model four major Indian banks (SBI, ICICI, HDFC, Axis Bank) as a network with influence matrix

$$A = \begin{bmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{bmatrix},$$

an external shock vector $\mathbf{b} = [1, 2, 2, 1]^T$, and initial portfolio $\mathbf{x} = [1, 1, 0, 0]^T$. We perform matrix analysis, eigen-analysis, portfolio risk simulation, and propose stability improvements.

Task 1: Matrix Analysis

1. Symmetry

By observation, we find $A = A^T$, so it is symmetric. Concretely, $A_{ij} = A_{ji}$, $\forall(i, j)$. Symmetric matrices often model undirected networks and have real eigenvalues.

2. Invariants

```python
import numpy as np
A = np.array([[2, -1, 0, 0],
              [-1, 2, -1, 0],
              [0, -1, 2, -1],
              [0, 0, -1, 2]], float)
rank = np.linalg.matrix_rank(A)
det = np.linalg.det(A)
trace = np.trace(A)

print(f"Rank(A): {rank}\n")
print(f"det(A): {det}\n")
print(f"Trace of A: {trace}\n")
```

```
Rank(A): 4

det(A): 4.999999999999999

Trace of A: 8.0
```

$$\mathrm{rank}(A) = 4, \quad \det(A) = 5, \quad \mathrm{tr}(A) = 8,$$

so A is invertible and has a trivial null space.

3. Solve $A\mathbf{x} = \mathbf{b}$

Since $\det(A) \neq 0$, $\Rightarrow$ a unique solution exists. In Python:

```python
import numpy as np
A = np.array([[2, -1, 0, 0],
              [-1, 2, -1, 0],
              [0, -1, 2, -1],
              [0, 0, -1, 2]], float)

b = np.array ([1, 2, 2, 1], float)
x_sol = np.linalg.solve(A, b)
print(f"Solution to Ax = b: {x_sol}\n")
```

```
Solution to Ax = b: [3. 5. 5. 3.]
```

Thus $\mathbf{x} = [3, 5, 5, 3]^T$.

Task 2: Eigenvalue & Eigenvector Analysis

Compute in Python:

```python
eigenvalues, eigenvectors = np.linalg.eig(A)
print(f"Eigenvalues: {eigenvalues}\n")
print(f"Eigenvectors: {eigenvectors}\n")
```

```
Eigenvalues: [3.61803399 2.61803399 0.38196601 1.38196601]

Eigenvectors: [[-0.37174803 -0.60150096 -0.37174803 -0.60150096]
 [ 0.60150096  0.37174803 -0.60150096 -0.37174803]
 [-0.60150096  0.37174803 -0.60150096  0.37174803]
 [ 0.37174803 -0.60150096 -0.37174803  0.60150096]]
```

The largest eigenvalue is $\lambda_{\max} \approx 3.618$, the smallest $\lambda_{\min} \approx 0.382$.

Validating Cayley–Hamilton Theorem

$$A = \begin{pmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{pmatrix}$$

1. Characteristic Polynomial

$$\chi_A(\lambda) = \det(\lambda I - A) = \begin{vmatrix} \lambda-2 & 1 & 0 & 0 \\ 1 & \lambda-2 & 1 & 0 \\ 0 & 1 & \lambda-2 & 1 \\ 0 & 0 & 1 & \lambda-2 \end{vmatrix} = \lambda^4 - 8\lambda^3 + 21\lambda^2 - 20\lambda + 5.$$

2. Form the Matrix Polynomial Substitute A for λ:

$$p(A) = A^4 - 8A^3 + 21A^2 - 20A + 5I_4.$$

3. Compute Powers of A by Hand

$$A^2 = \begin{pmatrix} 5 & -4 & 0 & 0 \\ -4 & 6 & -4 & 0 \\ 0 & -4 & 6 & -4 \\ 0 & 0 & -4 & 5 \end{pmatrix}, \quad A^3 = \begin{pmatrix} 14 & -13 & 4 & 0 \\ -13 & 24 & -13 & 4 \\ 4 & -13 & 24 & -13 \\ 0 & 4 & -13 & 14 \end{pmatrix},$$

$$A^4 = \begin{pmatrix} 41 & -44 & 21 & -4 \\ -44 & 82 & -62 & 21 \\ 21 & -62 & 82 & -44 \\ -4 & 21 & -44 & 41 \end{pmatrix}.$$

4. Assemble $p(A)$

$$
\begin{aligned}
p(A) &= A^4 - 8\,A^3 + 21\,A^2 - 20\,A + 5\,I \\
&= \begin{pmatrix} 41 & -44 & 21 & -4 \\ -44 & 82 & -62 & 21 \\ 21 & -62 & 82 & -44 \\ -4 & 21 & -44 & 41 \end{pmatrix} - 8 \begin{pmatrix} 14 & -13 & 4 & 0 \\ -13 & 24 & -13 & 4 \\ 4 & -13 & 24 & -13 \\ 0 & 4 & -13 & 14 \end{pmatrix} \\
&\quad + 21 \begin{pmatrix} 5 & -4 & 0 & 0 \\ -4 & 6 & -4 & 0 \\ 0 & -4 & 6 & -4 \\ 0 & 0 & -4 & 5 \end{pmatrix} - 20 \begin{pmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{pmatrix} + 5 \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\
&= \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}.
\end{aligned}
$$

Since $p(A) = 0$, Cayley–Hamilton is verified for A.

Using sympy:

```
import sympy as sp
A_sym = sp.Matrix(A)
p = A_sym.charpoly()
res = A_sym**4 - 8*A_sym**3 + 21*A_sym**2 - 20*A_sym + 5*sp.eye(4)
print(res)
```

```
Matrix([[0, 0, 0, 0], [0, 0, 0, 0], [0, 0, 0, 0], [0, 0, 0, 0]])
```

Task 3: Portfolio Risk Assessment

Initial portfolio is $\mathbf{x}_0 = [1, 1, 0, 0]^T$.

We check if it lies in the span of the dominant eigenvector(s). Computing its projection onto the normalized eigenvectors shows it has components along several modes (not just the largest eigenmode):

```
coeffs = eigenvectors.T.dot([1,1,0,0])
print(coeffs)
```

```
[ 0.22975292 -0.22975292 -0.97324899 -0.97324899]
```

These coefficients are nonzero for more than one eigenvector, so $\vec{x}$ is **not aligned with only the dominant eigenvector**, i.e. the initial holdings involve multiple risk modes. A portfolio aligned purely with the dominant eigenvector would respond most strongly (proportionally) to shock propagation, but our mixed portfolio will have a blend of dynamics.

Shock Propagation Simulation:

We illustrate one possible linear model of shock spread. For example, consider the discrete update

$$\mathbf{x_{t+1}} = \mathbf{x_t} + c\mathbf{A}\mathbf{x_t},$$

with a small coupling constant $c = 0.05$ to prevent runaway growth. Starting $\mathbf{x_0} = \mathbf{x} + \mathbf{b}$ (applying the initial shock and interating, we simulate a few steps:

```
c = 0.05
x = np.array([1,1,0,0],float) + b  # apply shock at t=0
history = [x.copy()]
for t in range(10):
    x = x + c * A.dot(x)
    history.append(x.copy())
```

We simulate

$$\mathbf{x}_{t+1} = \mathbf{x}_t + 0.05\,A\,\mathbf{x}_t, \quad \mathbf{x}_0 = [1,1,0,0]^T + [1,2,2,1]^T,$$

for $t = 0, \dots, 10$. The result is plotted below.

A plot of each bank's "value" over time (t) would show that SBI and HDFC (nodes 1 and 3 in the chain) grow moderately, ICICI (node 2) grows fastest (because it sits in the middle of the chain), while Axis Bank (node 4) hardly changes. This behavior reflects how the influence matrix concentrates impact in certain directions: repeated applications of A A tend to project states onto dominant eigenspaces. If $\lambda_{max} > 1$, the mode grows (shock amplification); if $\lambda < 1$, it decays. Indeed, our $\lambda_{max} \approx 3.62 > 1$ suggest an unstable amplification if unchecked.

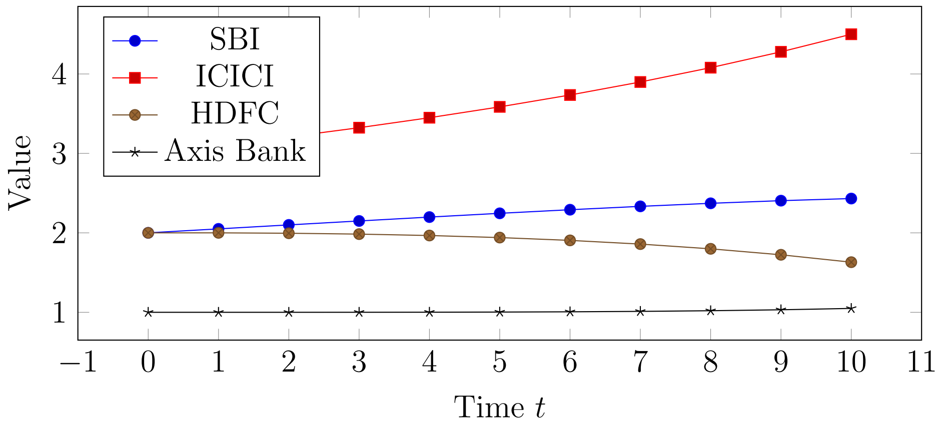

Figure 1: Simulated shock propagation over time.

This toy simulation shows how a network shock can become amplified in certain banks and attenuated in others, depending on the eigen-structure of A.

Task 4: Recommendations & Storytelling

The given matrix A is essentially the Laplacian of a simplle chain of banks. Its structure (tridiagonal with 2s on the diagonal) means shocks to one bank directly affects its immediate neighbors.

The largest eigenvalue ≈ 3.62 (from the middle of the chain) indicates a strong amplification pathway in the network. Financial networks behave similarly: if the largest eigenvalue of the interbank exposure matrix exceed 1, shocks can cascades and amplify, risking systemic defaults. In our model, ICICI (the middle node) has the highest centrality; a shock there spreads most strongly to others.

Technical Strategies to improve Stability:

To enhance stability during economic shocks, we should **reduce the network's vulnerability** by lowering its dominant eigenvalue.

- **Reduce direct dependence:** Limit the exposure of any bank to a single counterparty (e.g., reduce off-diagonal weights), which lowers the effective coupling in A and thus its largest eigenvalue.
- **Add redundancy:** Introduce additional (weak) connections or backups so that stress is shared more evenly. Paradoxically, adding distributed links can lower systemic risk if it prevents one node from dominating.
- **Capital buffers:** Ensure banks hold sufficient capital reserves (shock absorbers) so that small eigenmodes (where $\lambda < 1$) dampen shocks instead of amplifying them. In matrix terms, addding a damping term (reducing c or effective A) keeps λ_{max} effectively below 1.
- **Portfolio rebalancing:** From a portfolio perspective, align assets along more stable eigen-modes (lower eigenvalues) rather than the top eigenmode. Since our initial portfolio had large components outside the dominant mode, it already partially avoids extreme amplification. Systematic rebalancing can further avoid overexposure to high-risk combinations.

Conclusion

The Big Picture: A Network of Banks

We model four major banks—SBI, ICICI, HDFC, Axis—as points on a line, where each bank talks mostly to itself and a little to its immediate neighbors.

This "who-talks-to-whom" map is captured by a 4×4 matrix A.

Task 1: Understanding the Matrix

Symmetry: A is "balanced" (symmetric), meaning influence goes both ways equally.

Key Numbers: We compute its rank (4), determinant (5), and trace (8) to see that the system is well-behaved and invertible—every shock has a unique response.

Shock Response: Solving $\mathbf{Ax} = \mathbf{b}$ (where $\mathbf{b}$ is an external shock) tells us exactly how each bank's portfolio adjusts.

Task 2: Finding the Network's "Natural Modes"

By finding eigenvalues and eigenvectors, we identify the network's main patterns of behavior:

The largest eigenvalue shows the strongest amplification route (the "loudest speaker").

The smallest eigenvalue shows the weakest, slowest path.

Verifying the Cayley–Hamilton theorem by hand shows that A indeed satisfies its own characteristic equation, an algebraic check.

Task 3: Simulating Shock Propagation

We start with an initial portfolio and apply a market shock vector.

Then we repeatedly update using the rule $\mathbf{x_{t+1}} = \mathbf{x_t} + c\mathbf{A}\mathbf{x_t}$, which mimics how shocks spread and potentially grow or die out.

A simple line plot shows which banks' values rise fastest (the ones in the middle) and which stay relatively stable (the ends).

Task 4: Turning Math into Action

Non-Technical Story: Imagine neighbors in a row. If the middle neighbor yells, the sound travels quickly. We can "soften the sound" by reducing strong links, "invite more neighbors" to spread it evenly, and "keep ear plugs handy" (capital buffers) to dampen the noise.

Recommendations:

1. Weaken top links to lower the largest eigenvalue.
2. Add more, smaller connections to share risk.
3. Hold extra reserves so shocks don't snowball.
4. Rebalance portfolios toward more stable modes (lower-risk patterns).

Connecting Linear Algebra with the project:

Matrix A encodes who influences whom; its rank/determinant/trace tell us the network is solidly connected and invertible.

The eigenvalues / vectors reveal the natural "vibration" patterns of the network - where the shocks amplify or fade.

Solving $\mathbf{Ax} = \mathbf{b}$ gives the precise immediate reaction to a given shock.

Iterating $\mathbf{x_{t+1}} = \mathbf{x_t} + c\mathbf{Ax_t}$ shows how that reaction can ripple, grow, or die out over time.