ASSIGNMENT 1 REPORT

Varun Gopal

CO22BTECH11015

Answer 1

(a) File Sizes

After running the code for n=4000, two files were generated:

- **ASCII file**: Stores data in ASCII format with a size of **327 MB**.
- **Binary file**: Stores data in binary format with a size of **125 MB**.

(b) Memory and Disk Usage for double

- **Size in memory** : Calculated as $8 \times 4000 \times 4000 = 128{,}000{,}0008 \times 4000 \times 4000 = 128{,}000{,}000$ bytes $\approx$ **128 MB**.
- **Disk usage** :
  - ASCII file: **327 MB** (due to text representation, decimal precision, and line formatting).
  - Binary file: **125 MB** (direct memory dump, hence more compact).

Comments on Format

- Binary format is better suited for storing large datasets because:
  - The ASCII file used up a lot of data in comparision to the binary file due to storing the numbers as ASCII characters. Whereas binary file stores raw binary information in disk.
  - It is **space-efficient** and **faster** for reading/writing.
  - Maintains **full numerical precision** without errors caused by floating-point representation in ASCII.

Answer 2

Eigenvector Check

The results demonstrate whether each vector is an eigenvector of the given matrix. The algorithm used is as follows:

1. Input Details:
    - Given matrix An×n and vector xn×1.
2. Steps:
    - Compute y=Ax.
    - If x is an eigenvector, it satisfies: Ax= λ x
    - Find eigenvalue λ using: λ using the first row elements
    - Check if the difference y− λ x is close to zero within a tolerance.
    - We are using INT_MAX as a flag value to check whether an eigen value exists or not. If we want to allow even this value to be permitted then use an extra Boolean function to check if the eigenvalue exists or not.

Testing

The code was tested with inputs for n=3,5,50,80, producing accurate results for eigenvector validation.

Sample Outputs for n=80,50:-

```
vec_000080_000001.in : Yes : 0.333018
vec_000080_000002.in : Yes : 0.493142
vec_000080_000003.in : Yes : 0.939275
vec_000080_000004.in : No
```

```
vec_000050_000001.in : No
vec_000050_000002.in : Yes : 0.479628
vec_000050_000003.in : Yes : 1.337887
vec_000050_000004.in : No
```