ME5470-HW1_Report

CO21BTECH11005

QUESTION-1

a) Once the code execution is done we get two files as outputs:
   One is the ASCII file, where the data is stored in ASCII format.
   The other is the Binary file, where data is stored in binary format.

   The size of binary file = 125MB
   The size of the ASCII file = 360MB

b) Size of array in the memory by multiplying = 8 * 4000 * 4000 = 128MB
   So what we understand is binary file format is better for storage on clause if human readability is not required.

   However, if the data needs to be shared or reviewed manually, ASCII format should be considered despite the trade-off in size.

QUESTION-2

Mathematical Explanation of the Algorithm

The algorithm provided for checking whether a vector is an eigenvector of a given matrix involves several mathematical operations. Below, the steps are explained mathematically:

1. Matrix-Vector Multiplication:
   - Given an n x n matrix A and a vector x of size n, compute the matrix-vector product:
     y = A * x
     where y_i = sum(A[i][j] * x[j] for j in range(n)) for i = 1, 2, ..., n.

2. Scalar Multiplication Check:

- The algorithm checks if y is a scalar multiple of x. Mathematically, this means verifying if there exists a scalar lambda (the eigenvalue) such that:
  y = lambda * x
- Equivalently, this requires verifying the proportionality condition for every pair of components i and i+1 of y and x:
  y[i] * x[i+1] = y[i+1] * x[i] for all i = 1, 2, ..., n-1.
- If any pair fails this condition, the vector x is not an eigenvector.

3. Determining the Eigenvalue:
- If x is confirmed to be an eigenvector, the eigenvalue lambda is calculated as the ratio:
  lambda = y[k] / x[k]
  where x[k] != 0. This ratio should remain consistent for all k where x[k] != 0.

4. Handling Numerical Precision:
- To account for computational inaccuracies due to floating-point arithmetic, a small tolerance epsilon is used:
  abs(y[i] * x[i+1] - y[i+1] * x[i]) < epsilon
  where epsilon is set to 1e-10. This ensures that small numerical errors do not incorrectly disqualify eigenvectors.

5. Iterative Vector Validation:
- The algorithm processes multiple vectors sequentially. For each vector x_i:
  - Compute y_i = A * x_i.
  - Check the proportionality condition to determine if x_i is an eigenvector.
  - If true, compute and output the eigenvalue lambda. Otherwise, indicate that x_i is not an eigenvector.

This method effectively combines matrix-vector multiplication, scalar comparison, and tolerance-based validation to identify eigenvectors and their corresponding eigenvalues.

```
vec_000003_000001.in : Yes : -6.000000
vec_000003_000002.in : Yes : -6.000000
vec_000003_000003.in : Yes : -1.000000
vec_000003_000004.in : Not an eigenvector
```

The above are the results for n = 3

```
vec_000005_000001.in : Yes : 0.268098
vec_000005_000002.in : Not an eigenvector
vec_000005_000003.in : Yes : 0.986875
vec_000005_000004.in : Yes : 1.399039
```

The above are the results for n = 5

```
vec_000050_000001.in : Not an eigenvector
vec_000050_000002.in : Yes : 0.479628
vec_000050_000003.in : Yes : 1.337887
vec_000050_000004.in : Not an eigenvector
```

The above are the results for n = 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 : Not an eigenvector
```

The above are the results for n = 80