**ESO-208A**

**Computational Methods in Engineering**

**Assignment 2**

**2022-23 Semester-1**

**Name-Navneet Singh**

**ROLL NO. 200626**

**Programming Assignment 2:** Linear Simultaneous Equations and Eigenvalues

1. **AX = B**

Test Matrix:

Text file(.txt):

3

4.0 2.0 0.0 10.0

2.0 4.0 1.0 11.5

0.0 1.0 5.0 4.5

Result:

1. **LU Decomposition by Crout (without Pivoting):**

**X**

1.500000

2.000000

0.500000

**L**

4.000000 0.000000 0.000000

2.000000 3.000000 0.000000

0.000000 1.000000 4.666667

**U**

1.000000 0.500000 0.000000

0.000000 1.000000 0.333333

0.000000 0.000000 4.666667

**Q2. EIGENVALUES:**

Test matrix:

Maximum iterations: 50

Maximum relative approximate error: 0.001%

Find Eigenvalue closest to: 8

**Text file:**

3

8.0 -1.0 -1.0

-1.0 4.0 -2.0

-1.0 -2.0 10.0

100

0.001

8.0

**Results:**

1. **Power method**

**Eigenvalue: 10.778917**

**Eigenvector:**

**-0.249717**

**-0.240120**

**0.938075**

**\*The method is slow to convergence**

**Iterations: 25**

**Eigenvalue Estimates:**

|  |  |
| --- | --- |
| **Iteration Number** | **Estimate** |
| **1** | **5.354126** |
| **2** | **8.141196** |
| **3** | **9.380869** |
| **4** | **9.862671** |
| **5** | **10.174000** |
| **6** | **10.396165** |
| **7** | **10.545651** |
| **8** | **10.640354** |
| **9** | **10.697925** |
| **10** | **10.732034** |
| **11** | **10.751935** |
| **12** | **10.763442** |
| **13** | **10.770060** |
| **14** | **10.773856** |
| **15** | **10.776029** |
| **16** | **10.777271** |
| **17** | **10.777982** |
| **18** | **10.778387** |
| **19** | **10.778619** |
| **20** | **10.778752** |
| **21** | **10.778827** |
| **22** | **10.778871** |
| **23** | **10.778895** |
| **24** | **10.778909** |
| **25** | **10.778917** |

1. **Inverse Power method:**

**Eigenvalue: 3.074945**

**Eigenvector:**

**0.248710**

**0.920346**

**0.301839**

**\*The method is faster because the eigenvector corresponds close to , the starting vector.**

**Iterations: 7**

**Eigenvalue Estimates:**

|  |  |
| --- | --- |
| **Iteration number** | **Eigenvalue Estimates** |
| **1** | **3.535534** |
| **2** | **3.139889** |
| **3** | **3.083686** |
| **4** | **3.076137** |
| **5** | **3.075107** |
| **6** | **3.074965** |
| **7** | **3.074945** |

1. **Inverse Power method with shift**

**Eigenvalue: 8.289168**

**Eigenvector:**

**-0.947277**

**0.261158**

**-0.185640**

**Iterations: 8**

**Eigenvalue Estimates:**

|  |  |
| --- | --- |
| **Iteration number** | **Eigenvalue Estimates** |
| **1** | **8.316228** |
| **2** | **8.246682** |
| **3** | **8.284137** |
| **4** | **8.288587** |
| **5** | **8.289101** |
| **6** | **8.289161** |
| **7** | **8.289168** |
| **8** | **8.289168** |

**This method is again efficient since it targets the eigenvector to be found and is also fast because the eigenvector happens to be close to the starting vector.**

1. **QR Method**

**Eigenvalues:**

**10.778923**

**8.146136**

**3.074941**

**Iterations: 28**

**Eigenvalue Estimates:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Iteration** |  |  |  |
| **1** | **8.393939** | **7.097507** | **6.508554** |
| **2** | **8.616299** | **9.817073** | **3.566628** |
| **3** | **8.879953** | **10.000502** | **3.119545** |
| **4** | **9.209505** | **9.711613** | **3.078882** |
| **5** | **9.574800** | **9.349896** | **3.075303** |
| **6** | **9.923394** | **9.001629** | **3.074977** |
| **7** | **10.211169** | **8.713886** | **3.074945** |
| **8** | **10.421606** | **8.503452** | **3.074942** |
| **9** | **10.562231** | **8.362828** | **3.074941** |
| **10** | **10.650632** | **8.274426** | **3.074941** |
| **11** | **10.704088** | **8.220971** | **3.074941** |
| **12** | **10.735655** | **8.189403** | **3.074941** |
| **13** | **10.754037** | **8.171021** | **3.074941** |
| **14** | **10.764654** | **8.160405** | **3.074941** |
| **15** | **10.770756** | **8.154302** | **3.074941** |
| **16** | **10.774255** | **8.150804** | **3.074941** |
| **17** | **10.776257** | **8.148802** | **3.074941** |
| **18** | **10.777402** | **8.147657** | **3.074941** |
| **19** | **10.778056** | **8.147003** | **3.074941** |
| **20** | **10.778430** | **8.146629** | **3.074941** |
| **21** | **10.778644** | **8.146415** | **3.074941** |
| **22** | **10.778766** | **8.146293** | **3.074941** |
| **23** | **10.778835** | **8.146223** | **3.074941** |
| **24** | **10.778875** | **8.146184** | **3.074941** |
| **25** | **10.778898** | **8.146161** | **3.074941** |
| **26** | **10.778911** | **8.146148** | **3.074941** |
| **27** | **10.778918** | **8.146141** | **3.074941** |
| **28** | **10.778923** | **8.146136** | **3.074941** |

**As can be seen, the method is extremely tedious and computationally heavy. But it gives the eigenvalues and eigenvectors of a matrix directly. So due to less assumptions in its methodology, it becomes complex but much more useful for high dimensional vector spaces where taking assumptions is not humanely possible.**