Keneskhanov Madiyar, MSc, Teacher, madiyar.kengeskanov@astanait.edu.kz

Astana IT University, Department of Computational and Data Science

Assignment 3 - Computational Mathematics
Week 4

Instructions

For each problem, please include:
- Source code with comments explaining all important steps.
- Provide all calculation results, including intermediate steps.
- A screenshot of the program output and explanation.
- Provide mathematical justifications for methods where appropriate.
- For eigenvalue methods, provide accuracy and convergence.
- Test results using Python library functions, if appropriate.
- Include graphs/charts or tables if appropriate to illustrate the result.

Save your answers as a PDF report and submit it to the Moodle.

Task 1: Iterative method for matrix inversion.

Problem:

1. Implement an iterative method to compute the inverse of matrix A^{-1}. Use an initial guess B=1/tr(A) · I, where tr(A) is the trace of the matrix. Set the accuracy to 10^{-6}.
2. Matrix: $$A = \begin{bmatrix} 5 & 2 & 1 \\ 2 & 6 & 3 \\ 1 & 3 & 7 \end{bmatrix}$$

Required:

1. Print the resulting inverse matrix.
2. Compare the result with the built-in function numpy.linalg.inv.

Task 2: LU factorization and solution of a system of linear equations.

Problem:

1. Perform LU factorization of the matrix:

$$A = \begin{bmatrix} 10 & -1 & 2 & 0 \\ -1 & 11 & -1 & 3 \\ 2 & -1 & 10 & -1 \\ 0 & 3 & -1 & 8 \end{bmatrix}$$

2. Using the result of the expansion, solve the system Ax=b, where:

$$b = \begin{bmatrix} 5 \\ 20 \\ -10 \\ 15 \end{bmatrix}$$

Required:

1. Print matrices L and U.
2. Solve the system and print the x values.
3. Compare the result with the solution via numpy.linalg.solve.

Task 3: Finding the Largest Eigenvalue and Vector Using Power Method.

Problem:

1. Implement the power iteration method to find the largest eigenvalue and the corresponding eigenvector.
2. Matrix:
$$\begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix}$$
3. Initial vector v_0=[1,0,0].

Required:

1. Find the largest eigenvalue and vector.
2. Compare the result with the numpy.linalg.eig function.

Task 4: Comparison of Givens and Householder methods.

Problem:

1. Reduce the following matrix to upper triangular form using:
   - Givens' method.
   - Householder's method.

$$A = \begin{bmatrix} 4 & 1 & 2 & 0 \\ 1 & 3 & 1 & 2 \\ 2 & 1 & 5 & 1 \\ 0 & 2 & 1 & 4 \end{bmatrix}$$

Required:

1. Derive the Q and R matrices for each method.
2. Compare the efficiency and numerical stability of the two methods.

Task 5: Finding all eigenvalues using Jacobi's method.

Problem:

1. Using Jacobi's method, find all eigenvalues for the following matrix: Set the accuracy to 10^{-6}.

$$\begin{bmatrix} 1 & 1 & 0.5 \\ 1 & 1 & 0.25 \\ 0.5 & 0.25 & 2 \end{bmatrix}$$

Required:

1. Print the eigenvalues.
2. Compare the result with the numpy.linalg.eigvals function.

P.S. Please be prepared to explain your code/solution/answers.