+ Code + Text

NAME:- KETAN DILIP ATTARDE

REG NO:- 24-27-06

MTECH DATA SCIENCE

Algorithm: Gaussian Elimination with Partial Pivoting and Back Substitution

Input:

- Matrix A of size $n * n$
- Vector b of size n

Output:

- Solution vector x

Steps:

1. **Augment Matrix:**
    - Create an augmented matrix by combining matrix A and vector b.
2. **Gaussian Elimination with Partial Pivoting:**
    - For each iteration k from 0 to $n-1$:
        - Find the pivot index by selecting the row with the maximum absolute value in the current column (from k to $n-1$).
        - Swap rows to move the pivot element to the current row.
        - Eliminate entries below the pivot by subtracting a multiple of the pivot row from each subsequent row.
3. **Back Substitution:**
    - Initialize a solution vector x of size n with zeros.
    - For each row i from $n-1$ to 0 (backward):

- Calculate $x[i]$ using the formula: $x[i] = \frac{\{\text{augmented\_matrix}\}[i,-1] - \sum_{j=i+1}^{n} \{\text{augmented\_matrix}\}[i,j] \cdot x[j]}{\{\text{augmented\_matrix}\}[i,i]}$

4. **Output:**
    - The solution vector x contains the values of the unknowns.

```
import numpy as np


def gaussian_elimination_pivot(A, b):
    # Combine the matrix A and vector b into one augmented matrix
    augmented_matrix = np.column_stack((A, b))

    n = len(b)

    for k in range(n):
        # Partial pivoting: find the pivot (maximum element in the current column)
        pivot_index = np.argmax(np.abs(augmented_matrix[k:, k])) + k

        # Swap rows to move the pivot element to the current row
        augmented_matrix[[k, pivot_index]] = augmented_matrix[[pivot_index, k]]

        # Eliminate entries below the pivot
        for i in range(k + 1, n):
            factor = augmented_matrix[i, k] / augmented_matrix[k, k]
            augmented_matrix[i, k:] -= factor * augmented_matrix[k, k:]

    # Back substitution
    x = np.zeros(n)
    for i in range(n - 1, -1, -1):
        x[i] = (augmented_matrix[i, -1] - np.dot(augmented_matrix[i,
                i+1:n], x[i+1:])) / augmented_matrix[i, i]

    return x
```

```
# Example usage
A = np.array([[2, 1, -1],
              [-3, -1, 2],
              [-2, 1, 2]]).astype(np.float64)

b = np.array([8, -11, -3]).astype(np.float64)

print("The augmented matrix is:- ")
print(np.column_stack((A, b)))
print("-"*50)

solution = gaussian_elimination_pivot(A, b)
print("Solution:", solution)
```

```
The augmented matrix is:-
[[  2.   1.  -1.   8.]
 [ -3.  -1.   2. -11.]
 [ -2.   1.   2.  -3.]]
--------------------------------------------------
Solution: [ 2.  3. -1.]
```