Project 3

An algorithm for Partial Pivoting in Gaussian Elimination for Numerical Stability

Govher Sapardurdyyeva
South Dakota State University
Math 374 - Scientific Computation
Jung-Han Kimn
May 9, 2025

Abstract

This project uses Gaussian Elimination and Partial Pivoting to solve linear equations represented by n×n matrices. To improve numerical stability and eliminate rounding errors, the main algorithm converts the coefficient matrix to an upper triangular form and uses partial pivoting (selecting the biggest absolute value in each column as the pivot element). The final solution is obtained by back substitution. The algorithm was evaluated on a 4x4 matrix provided by the professor, as well as a 3x3 matrix, to ensure its accuracy and generality.

Introduction

Steps for an Algorithm to Solve n×n Matrices Using Gaussian Elimination with Partial Pivoting

This project aims to solve a system of equations using Gaussian elimination with partial pivoting. We begin by applying the algorithm, implemented in Python, to solve a 4x4 matrix provided by the professor.

$$\begin{pmatrix} 3 & -13 & 9 & 3 \\ -6 & 4 & 1 & -18 \\ 6 & -2 & 2 & 4 \\ 12 & -8 & 6 & 10 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} -19 \\ -34 \\ 16 \\ 26 \end{pmatrix}$$

Finding the pivot is the first step in the algorithm. This entails choosing the entry with the highest absolute value, which will serve as our pivot, by scanning the first column (column 0) from the current row downward. Then the code checks if the pivot element in column `k` is zero. If the entire column below the current row is zero, the matrix is singular and has no unique solution. Otherwise, the algorithm proceeds by swapping the current row `k` with the row `pk` that contains the largest absolute value in column `k`—this is the partial pivoting step. The corresponding entries in the right-hand side vector `b` are also swapped to maintain consistency. After that, the algorithm eliminates entries below the pivot by dividing each entry in column `k` by the pivot element `A[k][k]`. Then, it updates the rest of the row using row operations to create zeros below the pivot, gradually transforming the matrix into an upper triangular form.

Step 0 (*Find the pivot):*

$$Find\ the\ row\ index\ P_k \geq k\ for\ which\ \left|a_{P_k},\ k\right| = max_{k \leq i \leq n}\left|a_i, k\right|$$

Step 1:

```
if A[pk][k] == 0: #if the entire column is 0
    print("The matrix is singular and has no unique solution")
    break
else:
    # Swap row k with row pk
    A[k], A[pk] = A[pk], A[k]
    b[k], b[pk] = b[pk], b[k]
```

Step 2

```
for i in range(k + 1, n):
    A[i][k] = A[i][k] / A[k][k]  #eliminate the entry
    for j in range(k + 1, n + 1):
    A[i][j] = A[i][j] - A[i][k] * A[k][j] #update the entry
```

Figure 1 illustrates all the steps of the algorithm for solving the 4x4 matrix, with the upper triangle of the matrix highlighted in yellow. The row operations performed are shown on the y-coordinate of the plane.

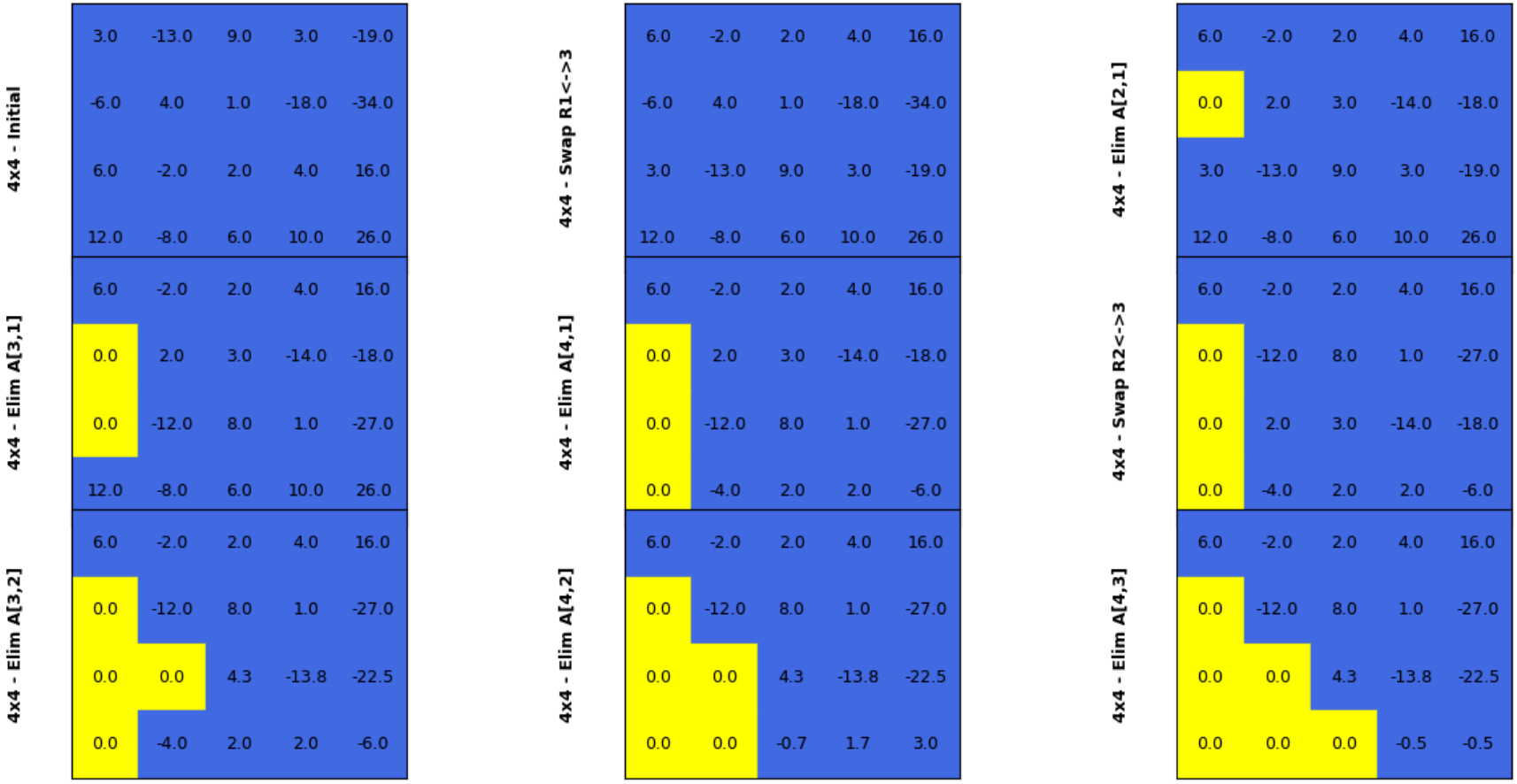

Figure 1 (4x4 Matrix Row Operations)

Figure 2 displays the back substitution process after the row operations for partial pivoting have been completed.

x1 = (16.0 - (-2.0 * x2) - (2.0 * x3) - (4.0 * x4)) / 6.0 = 3.00

x2 = (-27.0 - (8.0 * x3) - (1.0 * x4)) / -12.0 = 1.00

x3 = (-22.5 - (-13.8 * x4)) / 4.3 = -2.00

x4 = (-0.5) / -0.5 = 1.00

(Figure 2)

Now we apply the same algorithm to solve a 3x3 matrix. I selected a 3x3 matrix from *Elementary Linear Algebra* by Ron Larson, 8th edition, to demonstrate the algorithm.

$$\begin{bmatrix} 1 & -2 & 3 \\ -1 & 3 & 0 \\ 2 & -5 & 5 \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \begin{bmatrix} 9 \\ -4 \\ 17 \end{bmatrix}$$

Below is the solution produced by the algorithm, showing each intermediate step and the updated matrix, with the upper triangular portion highlighted in yellow in Figure 3.

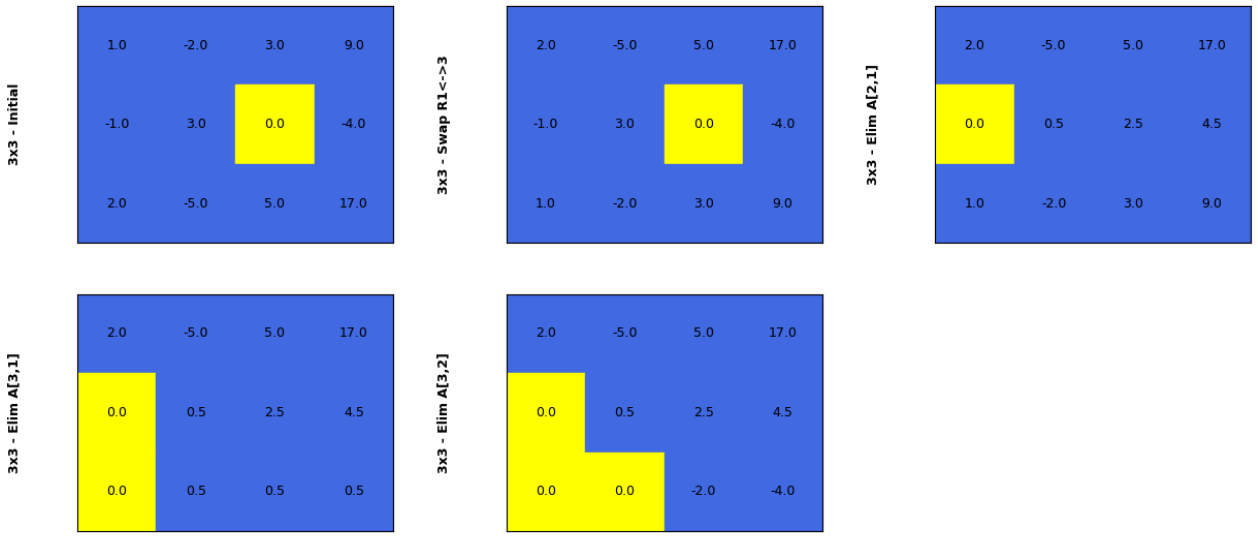

Figure 3 (3x3 Matrix Row Operations)

Next, Figure 4 presents the back substitution process used to determine the values of x.

```
x1 = (17.0 - (-5.0 * x2) - (5.0 * x3)) / 2.0 = 1.00

x2 = (4.5 - (2.5 * x3)) / 0.5 = -1.00

x3 = (-4.0) / -2.0 = 2.00
```

(Figure 4)

Conclusion

In conclusion, the algorithm successfully solves systems of linear equations for n×n matrices using Gaussian elimination with partial pivoting. Below are the results obtained from applying the method to the given matrices.

4x4 Matrix solutions to the system:

$x_1 = 3,\ x_2 = 1,\ x_3 = -2,\ x_4 = 1$

3x3 Matrix solutions to the system:

$x_1 = 1,\ x_2 = -1,\ x_3 = 2$

References

Cheney, E. W., & Kincaid, D. (2020). *Numerical Mathematics and Computing*. Brooks/Cole.

Higham, N. J. (2011). Gaussian elimination. *WIREs Computational Statistics*, *3*(3), 230–238. https://doi.org/10.1002/wics.164

Johnson, T. D. (n.d.). Gauss elimination with partial pivoting. https://web.mit.edu/10.001/Web/Course_Notes/GaussElimPivoting.html

Larson, R. (2017). *Elementary Linear Algebra*. Cengage Learning.