Solving Systems of Equations, Errors and Explorations

David Tran and Spencer Kelly

March 10, 2024

Abstract

1 Introduction

2 The PA = LU factorization method for linear systems

2.1 Why is PA = LU needed for solving linear systems approximately?

When solving linear systems of the form $Ax = b$, we begin by gaussian elimination of the matrix A, followed by back substitution, and ultimately arrive at our solution. However, when a particular matrix A is being used for multiple iterations, the overhead involved can become quite an obstacle. This is because the process of Gaussian elimination is a computationally expensive process, with complexity on the order $O(n^3)$. But with $PA = LU$ factorization, we essentially remove the overhead involved with Gaussian elimination, for all but the first iteration, by rewriting the matrix A in terms of the upper and lower matrices L, and U, respectively. Thus, for every subsequent iteration involving the same matrix, we need not perform gaussian elimination, since L and U allow us to immediately begin performing the second step of solving; back-substitution, which only has complexity $O(n^2)$.

However, when performing naive Gaussian elimination to form the matrices L and U, we are at risk of swamping, or the existence of a zero-pivot. With the help of a permutation matrix P, we can now swap rows and columns, to mitigate the propogation of errors due to multiplying rows by large values, and avoid zero-pivots. The permutation matrix P keeps track of the swapping of rows and columns, so that the linear system itself remains unperturbed (however it is now written $PAx = Pb$).

A great example would be the system $Ax = b$ with $A = \begin{pmatrix} 1 & 2 & 4 \\ 3 & 8 & 14 \\ 2 & 6 & 13 \end{pmatrix}$. To begin, we swap rows 1 and 2, since we want our multiplication to be by the smallest values possible during Gaussian elimination. When doing this, we update our permutation matrix: $P = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$, and we can subsequently perform gaussian elimination to yield the matrix $\begin{pmatrix} 3 & 8 & 14 \\ (\frac{1}{3}) & \frac{-2}{3} & \frac{-2}{3} \\ (\frac{2}{3}) & \frac{2}{3} & \frac{11}{3} \end{pmatrix}$, where the brackets around the values in the place of what should be 0 represents the multiplier for the elimination of the row, (important for bookkeeping when doing permutations).

Once again performing Gaussian elimination we obtain the matrix $\begin{pmatrix} 3 & 8 & 14 \\ (\frac{1}{3}) & \frac{-2}{3} & \frac{-2}{3} \\ (\frac{2}{3}) & (-1) & 3 \end{pmatrix}$. From this matrix, we can easily find that $L = \begin{pmatrix} 1 & 0 & 0 \\ \frac{1}{3} & 1 & 0 \\ \frac{2}{3} & -1 & 1 \end{pmatrix}$ and $U = \begin{pmatrix} 3 & 8 & 14 \\ 0 & \frac{-2}{3} & \frac{-2}{3} \\ 0 & 0 & 3 \end{pmatrix}$. We can then use these to solve linear systems for b_k, with the increased efficiency of LU factorization, and the error mitigation of the pivots enabled by the inclusion of the permutation matrix.

2.2 How to identify systems Ax = b for which PA = LU is not suited

Although it is a very effective direct method of giving (theoretically) the exact solution of the linear system, PA = LU factorization is not always the tool you want to employ for solving said linear system. For example, if the matrix involved is positive definite, and symmetric, we can employ methods such as Cholensky Factorization, which is more efficient. If the matrix is not only positive definite and symmetric, but also sparse, we can employ the Conjugate Gradient Method, which is even more efficient than the aforementioned method, and has even lower memory demand.

2.3 Larger applications of PA = LU factorization

The applications of $PA = LU$ factorization extend beyong just solving linear systems,and can be used to solve very common problems. One such problem is solving for the inverse of a matrix, which is made significantly easier when said matrix is decomposed into upper and lower matrices via $PA = LU$ factorization. On top of this, we can easily compute the determinant of a matrix given its $PA = LU$ factorization, when we recall that the determinant of a triangular matrix is just its diagonal entries, and that the determinant of a permutation matrix is just $(-1)^n$, where n is the number of rows swapped by the permutation.

3 Iterative solution of systems of linear equations

3.1 Solving an equation for n = 100,000

3.2 Comparison of PA = LU and Jacobi Iteration

3.3 Why is solving such large systems important in applications?

4 Implement Newton's method for multiple variables

4.1 Implement Newton's method for systems using vectorization

4.2 Testing

4.3 Challenging Example

5 Summary

6 Appendices

6.1 Code

6.2 Plots

7 Code

8 Summary

8.1 Results

8.2 Team Description

8.3 Future Explorations

8.4 References

Appendix