Gaussian Elimination and LU Decomposition

Mark Brautigam

Math 370 – Numerical Analysis – 11 April 2025

Gaussian Elimination

Gaussian Elimination is a method for solving a system of linear equations. It starts with an **augmented matrix** that has the system coefficients on the left and the result vector on the right. It proceeds by a sequence of row operations to create an upper triangular matrix, which has zeros below the diagonal. The resulting matrix has what is called **row echelon form.**

These row operations have the form of multiplying a row by a multiplier, then subtracting the result from another row in order to eliminate one of its coefficients. For example, consider the following matrix:

$$\left|\begin{array}{ccc|c} 1 & 2 & -1 & 3 \\ 2 & 1 & -2 & 3 \\ -3 & 1 & 1 & -6 \end{array}\right|$$

We could multiply row 1 by 2, then subtract the result from row 2. This results in the following matrix:

$$\left|\begin{array}{ccc|c} 1 & 2 & -1 & 3 \\ 0 & -3 & 0 & -3 \\ -3 & 1 & 1 & -6 \end{array}\right|$$

You can see that we have eliminated the coefficient in the first column of row 2. We can do similar operations to eliminate all the entries below the diagonal.

Then by another sequence of row operations called **back substitution,** we can transform the coefficient matrix to the identity matrix, which reveals the solution in the result vector. The row operations for back substitution use the same multiplication and subtraction operations as before. Back substitution starts with the upper triangular matrix that results from eliminating all the entries below the diagonal.

$$\left|\begin{array}{ccc|c} 1 & 2 & -1 & 3 \\ 0 & -3 & 0 & -3 \\ 0 & 0 & -2 & -4 \end{array}\right|$$

Because the last row of the matrix has the solution for the last variable, we can use it to eliminate that variable from the other rows above. Then we can do the same with the next-to-last row, working upward. This is why it is called back substitution, because we start at the bottom of the matrix and work backwards toward the top.

Partial Pivoting

Sometimes the system of equations has zero coefficients. This can be problematic if those zeros occur on the diagonal, because it prevents the row operations for succeeding rows in the matrix. This can be fixed by swapping two rows, because in a linear system, it really doesn't matter what order we look at the individual equations. This is called **partial pivoting.** We can just swap rows until the zeros get moved off the diagonal. When coding, we swap the row that has a zero on the diagonal with a row that has the highest coefficient (absolute value) in that column. This gives us consistency, efficiency, and stability, because higher coefficients keep the multipliers as small as possible, producing more accurate results in the row operations. Smaller numbers on the pivot points would require larger multipliers, which can swamp the other rows, so we try to avoid that. Choosing to put the higher numbers on the pivot points gives us smaller multipliers, which will have less probability of swamping the other rows.

LU Decomposition

LU Decomposition is a method for solving a system of linear equations by resolving the coefficient matrix A into a product of a lower triangular matrix (L) and an upper triangular matrix (U). Then

$$A = LU$$

We don't actually invert the original matrix A as a part of this process, but it is a requirement that the original matrix A be invertible. To perform LU Decomposition, we start by creating the U matrix from the A matrix using Gaussian Elimination as described above. In addition, we'll create the L matrix by starting with the identity matrix. The identity matrix has ones on the diagonal and zeros everywhere else.

$$I = \begin{vmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{vmatrix}$$

At each step in the LU Decomposition process, we store the multipliers in the corresponding entries of the L matrix. In the previous example, where we multiplied row 1 by 2 then subtracted the result from row 2, we would store the multiplier 2 into the first column of row 2 of the L matrix.

$$L = \begin{vmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 0 & 0 & 1 \end{vmatrix}$$

After we've completed these operations, we'll have an L and a U matrix, but we still won't have the solution. We derive the solution as follows. These matrix operations can represent the original system of equations:

$$Ax = b$$

Since $A = LU$, we now have $$LUx = b$$

We now define $c = Ux$, so we can solve the simpler equation

$$Lc = b$$

Since we know L and b, we can solve for c using back substitution. For example, we might have matrices that look like this:

$$\begin{vmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -3 & -2 & 1 \end{vmatrix} \text{ x } \begin{vmatrix} c_1 \\ c_2 \\ c_3 \end{vmatrix} = \begin{vmatrix} 2 \\ 3 \\ -6 \end{vmatrix}$$

Which we can easily solve:

$$c_1 = 2.$$
$$2c_1 + c_2 = 3 \text{ so } c_2 = -1$$
$$-3c_1 - 2c_2 + c_3 = -6 \text{ so } -6 + 2 + c_3 = -6 \text{ so } c_3 = -2$$

We still have to solve for x. We do so by solving the matrix equation

$$Ux = c$$

We already know U and c, so it just remains to calculate x by a similar process to how we calculated c.

LU Decomposition with Partial Pivoting

Just as with Gaussian Elimination, sometimes with LU Decomposition, we may find zeros on the diagonal that make further progress impossible. In those cases, we can again use partial pivoting to move the zero diagonal entries to another row where they won't be on the diagonal. When doing **LU Decomposition with Partial Pivoting**, we will add a new matrix called the Permutation Matrix (P). The Permutation Matrix just keeps track of which rows we have switched. This is important because at the end of the process, we'll need to switch the corresponding entries in the results matrix (b) also. The Permutation Matrix starts as an Identity Matrix, with ones one the diagonal and zeros off diagonal. Switching the rows just moves the ones to another row, so we still have a matrix with just ones and zeros, and these tell us which rows we have switch. For example, we might start with an identity matrix:

$$P = \begin{vmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{vmatrix}$$

If the Partial Pivoting process requires that we exchange the first and second rows, the resulting permutation matrix will look like this:

$$P = \begin{vmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{vmatrix}$$

To perform LU Decomposition with Partial Pivoting, we will be using the following equation:

$$PA = LU$$

Where

- P is the permutation matrix, which starts as the identity matrix,
- A is the coefficient matrix of the system we are solving,
- L is the lower triangular matrix, which starts as the identity matrix, and
- U is the upper triangular matrix, which starts the same as A.

We can see that, at the beginning, $P = L = I$, and $A = U$; therefore the equation is satisfied at the start. When we switch two rows in the permutation matrix, this is a reminder to switch those terms in the b matrix later. That is the purpose of the permutation matrix: it reminds us which rows we've switched. This is because the LU decomposition and the back substitution may occur at different times.

The sequence of operations looks like the following:

Start by checking the need for partial pivoting. We start with row 1, column 1. If that entry is not the largest (absolute value) in column 1, in matrix U, we exchange the first row with whichever row has the highest value in column 1. Then we perform the exact same row exchange operation on matrix P.

After this operation, we will perform a multiply-and-subtract operation on the U matrix to eliminate one of the entries below the diagonal. The first time, we will aim to achieve a zero in row 2, column 1. Future operations will aim to put a zero in other spots below the diagonal. This step is the same as what we did in Gaussian Elimination. But in LU Decomposition, we have the additional step of storing the multiplier in its appropriate spot in the L matrix.

Once we have our P, L, and U matrices, and the original A coefficient matrix and b vector, we can proceed to find x through a sequence of back substitutions similar to latter parts of both Gaussian Elimination and LU Decomposition, with an additional step. We want to solve the system of equations $Ax = b$, but what we have is $PA = LU$. Multiply both sides of $Ax = b$ by P to obtain:

$$PAx = Pb$$

Since LU = PA, we have $$LUx = Pb$$

Define vectors $d = Pb$ and $c = Ux$, to find $$Lc = d$$

Since we know L and b, we solve for c using back substitution.

Then we solve the equation $$Ux = c$$

Since we know U and c, this lets us solve for x, again by back substitution. These operations are made very easy because L and U are lower and upper triangular matrices that have enough zeros off diagonal to make easy solutions.

Gaussian Elimination vs LU Decomposition

LU Decomposition can be more efficient than Gaussian Elimination in the case where we have several similar systems of equations such that the coefficients are the same but the output vectors are different. That is, we may be solving several different systems of the form $Ax = b$ with the same A but different b.

This article explains the rationale:
https://www.cl.cam.ac.uk/teaching/1314/NumMethods/supporting/mcmaster-kiruba-ludecomp.pdf
Numerical Methods course at University of Cambridge, SY 2013–2014, Dr. David Greaves.

If we were to attempt to solve these several systems by Gaussian Elimination, we'd have to start from scratch each time. Part of the reason is because both the A and b vectors are destroyed in the process of Gaussian Elimination. (Although I think this is a bit of a red herring because it is certainly possible to store away the matrix and vector immutably for safekeeping before doing the matrix operations on mutable copies.)

More pertinent is that the row operations would be the same on matrix A no matter what vector b is, so we'd certainly be duplicating a lot of effort. In a sense, LU Decomposition is an attempt to separate the operations on matrix A from the corresponding operations on vector b. We do this by saving a record of the operations we performed on matrix A, so that we can perform the same operations on vector b later. This lets us resolve matrix A just one time, after which we can perform the recorded operations on any number of vectors b we like, even at a later time.

LU Decomposition appears to have a few more steps, as the algorithm is longer. But, according to the textbook, even if we're solving only one equation, the number of computations necessary for LU Decomposition is no more than for Gaussian Elimination. So there seems to be no reason to favor Gaussian Elimination when LU Decomposition is more efficient when solving multiple equations and no worse when solving only one.

(Textbook, pages 86–87, Section 2.2.3, Complexity of the LU Factorization.)

Partial Pivoting or not?

Partial pivoting may be required to solve certain systems of equations. If there are zeros on any of the pivot points (the diagonal), partial pivoting is absolutely necessary to find a solution to the system. It might be possible to look at the matrix, and if no zeros are found on the diagonal, just solve without partial pivoting. But it might be possible that the multiply-and-subtract operations could create unintended zeros later that weren't apparent at the start. This would cause the process to fail. So it's better just to use partial pivoting in all cases.

The following page describes the problems with zeros on the diagonal, and very small numbers that might approximate division by zero when performing the matrix operations.

https://lemesurierb.people.charleston.edu/introduction-to-numerical-methods-and-analysis-julia/docs/linear-equations-2-pivoting.html
(College of Charleston, Introduction to Numerical Methods, Brenton LeMesurier, 2020–2021.)

This page from MIT describes the partial pivoting process and rationale:
https://web.mit.edu/10.001/Web/Course_Notes/GaussElimPivoting.html
(MIT, Terry D. Johnson, Fall 2000)

Part of the rationale is that by putting the largest absolute value at the pivot point, which determines the multiplier, we avoid round-off errors that might accrue from math operations on smaller numbers. When multiplying a very small number by a very large number to eliminate a number to get a subtraction term, the large multiplier may **swamp** the other numbers in that row and exceed the precision of the machine. This is described in the textbook. (Page 95, section 2.3.2, Swamping.) It is also described in other textbooks, such as

https://ece.uwaterloo.ca/~dwharder/nm/Lecture_materials/pdfs/3.3%20Linear%20algebra.pdf
(University of Waterloo, ECE 204, Numerical Methods, Douglas Wilhelm Harder, Winter 2025.)

Partial pivoting does require more operations, and in the case of LU Decomposition, additional memory to store the P matrix. But since the operation may fail without it, we consider it not optional.