LU Decomposition Example

Chuck Allison - CNS 3320

April 2006

The goal of LU decomposition is to factor a matrix of coefficients in a way that allows for efficient, accurate solutions of systems of equations with different right-hand sides. When using partial pivoting, a reordering of rows occurs, so we obtain a factorization that looks like

$$LU = PA$$

where L is lower triangular (with 1's on the diagonal), U is upper triangular (and is, in fact, the result of the Gaussian elimination with partial pivoting), and P is a permutation matrix that represents the row swaps that occurred.

To illustrate, consider the following system of equations:

$$\begin{aligned} 2x + y - 2z &= 10 \\ 3x + 2y + 2z &= 1 \\ 5x + 4y + 3z &= 4 \end{aligned}$$

To reduce the matrix of coefficients, $A = \begin{pmatrix} 2 & 1 & -2 \\ 3 & 2 & 2 \\ 5 & 4 & 3 \end{pmatrix}$ using partial pivoting, we begin by swapping rows 1 and 3:

$$P_{1,3}A = \begin{pmatrix} 5 & 4 & 3 \\ 3 & 2 & 2 \\ 2 & 1 & -2 \end{pmatrix}$$

Next we zero-out the numbers below the pivot 5 with the operations $E_{1,2}(-\frac{3}{5})$ and $E_{1,3}(-\frac{2}{5})$:

$$E_{1,3}(-\frac{2}{5})E_{1,2}(-\frac{3}{5})P_{1,3}A = \begin{pmatrix} 5 & 4 & 3 \\ (\frac{3}{5}) & -\frac{2}{5} & \frac{1}{5} \\ (\frac{2}{5}) & -\frac{3}{5} & -\frac{16}{5} \end{pmatrix}$$

The parenthesized entities represent the zeroes obtained through Gaussian elimination, but we will store the *opposite* of the multipliers in their place, because of the special property of elementary matrices that $E_{i,j}^{-1}(m) = E_{i,j}(-m)$.

To continue with the elimination, we notice that we must make $-\frac{3}{5}$ the new pivot so we swap rows 2 and 3 (including the multipliers–they must accompany the rows they operated on originally–you'll see why shortly):

$$P_{2,3}E_{1,3}(-\frac{2}{5})E_{1,2}(-\frac{3}{5})P_{1,3}A = \begin{pmatrix} 5 & 4 & 3 \\ (\frac{2}{5}) & -\frac{3}{5} & -\frac{16}{5} \\ (\frac{3}{5}) & -\frac{2}{5} & \frac{1}{5} \end{pmatrix}$$

and then eliminate the $-\frac{2}{5}$ in position [3,2], and store the opposite of the multiplier there:

$$E_{2,3}(-\frac{2}{3})P_{2,3}E_{1,3}(-\frac{2}{5})E_{1,2}(-\frac{3}{5})P_{1,3}A = \begin{pmatrix} 5 & 4 & 3 \\ (\frac{2}{5}) & -\frac{3}{5} & -\frac{16}{5} \\ (\frac{3}{5}) & (\frac{2}{3}) & \frac{7}{3} \end{pmatrix} = U \tag{1}$$

We now have

$$U = \begin{pmatrix} 5 & 4 & 3 \\ 0 & -\frac{3}{5} & -\frac{16}{5} \\ 0 & 0 & \frac{7}{3} \end{pmatrix}, L = \begin{pmatrix} 1 & 0 & 0 \\ \frac{2}{5} & 1 & 0 \\ \frac{3}{5} & \frac{2}{3} & 1 \end{pmatrix}$$

stored together in the original storage for A, and the reader can verify that $LU = PA$, where

$$P = P_{2,3}P_{1,3} = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

If you need further convincing that this is not just bunch of black magic (although it is sort of like magic :-), observe the following, starting with (1) (and remember that $P_{i,j}^{-1} = P_{i,j}$, and $E_{i,j}^{-1}(m) = E_{i,j}(-m)$):

$$\begin{aligned} E_{2,3}(-\frac{2}{3})P_{2,3}E_{1,3}(-\frac{2}{5})E_{1,2}(-\frac{3}{5})P_{1,3}A = U \\ \Rightarrow A = P_{1,3}E_{1,2}(\frac{3}{5})E_{1,3}(\frac{2}{5})P_{2,3}E_{2,3}(\frac{2}{3})U \\ \Rightarrow A = MU, \\ M = \begin{pmatrix} \frac{2}{5} & 1 & 0 \\ \frac{3}{5} & \frac{2}{3} & 1 \\ 1 & 0 & 0 \end{pmatrix} \end{aligned}$$

(You can verify M by computing the product of the elementary matrices in front of U above.) Now, notice that M is a permutation of L. In fact, if you apply $P = P_{2,3}P_{1,3}$ to M, you get the L we had above (that's the magic). So multiplying both sides by P gives:

$$A = MU \Rightarrow PA = PMU \Rightarrow PA = LU$$

This shows why it made sense to move the stored multipliers in the lower triangle together with the rest of their row during partial pivoting. We would have obtained an erroneous L otherwise.

We can now solve the original system as follows:

$$\begin{aligned} A\mathbf{x} &= \mathbf{b} \\ \Rightarrow PA\mathbf{x} &= P\mathbf{b} \\ \Rightarrow LU\mathbf{x} &= P\mathbf{b} \end{aligned}$$

By making the substitution $U\mathbf{x} = \mathbf{y}$, we first solve the triangular system $L\mathbf{y} = P\mathbf{b}$ with forward substitution, and then the triangular system $U\mathbf{x} = \mathbf{y}$ with back substitution. For the first system, $L\mathbf{y} = P\mathbf{b}$, we get:

$$\begin{aligned} y_1 &= 4 \\ \frac{2}{5}y_1 + y_2 &= 10 \\ \frac{3}{5}y_1 + \frac{2}{3}y_2 + y_3 &= 1 \end{aligned}$$

which has solution $(y_1, y_2, y_3) = (4, \frac{42}{5}, -7)$. We then solve $U\mathbf{x} = \mathbf{y}$:

$$\begin{aligned} 5x + 4y + 2z &= 4 \\ -\frac{3}{5}y - \frac{16}{5}z &= \frac{42}{5} \\ \frac{7}{3}z &= -7 \end{aligned}$$

which yields the final solution of $(x, y, z) = (1, 2, -3)$. This process of solving two triangular systems in succession can repeated for multiple right-hand sides, while L, U, and P need only be calculated once.

For efficiency in programming, we can store the swaps in a vector of length 2 ($= n-1$), since we don't swap the last row with anything at the end of the Gaussian elimination. So in the example above, the vector $(3, 3)$ records a $P_{1,3}$ followed by a $P_{2,3}$ (i.e., position 1 in the vector stores what row 1 was swapped with during elimination, and position 2 has what row 2 was subsequently swapped with). (Of course, in most of today's programming languages, arrays are zero based, so the first entry is really row 0.) In the case no swapping taking place, the ith position in this vector will be i itself.