Department of Mathematics, University of Utah

Analysis of Numerical Methods I
MTH6610 – Section 001 – Fall 2017

Lecture notes – Pivoting in LU factorizations
Wednesday October 16, 2019

These notes are not a substitute for class attendance. Their main purpose is to provide a lecture overview summarizing the topics covered.

Reading: Trefethen & Bau III, Lecture 21

We have seen that given an invertible $A \in C^{n\times n}$, then it may be possible to form the LU factorization of A:

$$A = LU,$$

where L is lower triangular and U is upper triangular, and both are square and invertible. It only "may" be possible since the procedure to form this factorization is Gaussian elimination, which can fail for computational reasons that are directly related to invertibility or conditioning of A.

Operation count

The LU factorization of A requires $n-1$ steps, and at step k, rows $k+1,\ldots,n$ have vector operations (about n arithmetic operations) performed on them to achieve elimination. Thus, the computational complexity of LU factorization is on the order of

$$\sum_{k=1}^{n-1}(n-k)n \sim \mathcal{O}(n^3),$$

revealing the large-n complexity of performing LU factorizations. This complexity is similar to the cost of performing SVD or QR decompositions, but the constant in front of n^3 is smaller.

Uses of LU factorizations

Note that, if this factorization succeeds, we can immediately perform a couple of useful operations:

- Given $b \in \mathbb{C}^n$, the solution to $Ax = b$ is $x = U^{-1}L^{-1}b$. Since L and U are triangular, then application of each inverse requires only $\mathcal{O}(n^2)$ work, which is less than the straightforward n^3 work usually required. The catch, of course, is that one must have the LU factorization on hand. Note that, taking $b = e_j$, this allows us to compute column j of A^{-1}.
- The factorization shows that $\det A = (\det L)(\det U)$. The determinants on the right-hand side are simply the product of the diagonal entries. Recall from the LU factorization procedure that the diagonal of L has entries 1. Therefore, we have $\det A = \det U = \prod_{j=1}^{n} u_{j,j}$. This is a computationally efficient method to compute determinants of square matrices.

Pivoting

Since matrices may not have LU factorizations, this seems to limit the applicability of such factorizations. However, we can ensure that matrices have LU decompositions by performing *pivoting*. First we need a definition.

Definition 1
Let $[n] = \{1, 2, \ldots, n\}$. A permutation σ of $[n]$ is a bijective map $\sigma : [n] \to [n]$.

Informally, a permutation σ is a reordering of the elements in $[n]$. The bijectivity of this map ensures that this is an actual reordering and not simply picking n values from $[n]$ with replacement, which allows for repeated values. This definition induces a definition of permutation matrices.

Definition 2
A matrix $P \in \mathbb{C}^{n\times n}$ is a permutation matrix if it has the form

$$P = \begin{pmatrix} e_{\sigma(1)} & e_{\sigma(2)} & \cdots & e_{\sigma(n)} \end{pmatrix},$$

where e_j, $j = 1, \ldots, n$, is the canonical basis in $\mathbb{C}^n$, and σ is a permutation of $[n]$.

Permutation matrices perform exactly what they sound like: the operation Px produces a vector whose entries are a reshuffling of the entries of x under the permutation map σ.
Permutations allow us to amend the deficiencies in the LU procedure via a process called *pivoting*. If $A \in \mathbb{C}^{n\times n}$, recall that step k of the algorithm views the $(n-k+1)\times(n-k+1)$ submatrix of the current reduced A. Suppose it has the form

$$\widetilde{A}_k = \begin{pmatrix} y_{1,1} & y_{1,2} & \cdots & y_{1,n-k+1} \\ y_{2,1} & y_{2,2} & \cdots & y_{2,n-k+1} \\ \vdots & \vdots & \ddots & \vdots \\ y_{n-k+1,1} & y_{n-k,2} & \cdots & y_{n-k+1,n-k+1} \end{pmatrix}$$

The standard LU process assumes $y_{1,1}$ is non-zero and uses it as the denominator in an arithmetic division. When $y_{1,1}$ is zero, the operation cannot proceed, and more generally when $|y_{1,1}|$ is very small, the LU procedure can suffer numerical instabilities. Pivoting is a strategy to ameliorate such instabilities.
For LU factorizations, we need not use $y_{1,1}$ to eliminate all entries below it. Instead, we may use $y_{2,1}$ to eliminate all other entries in the first column, or we may even use $y_{2,1}$ to eliminate all entries in the second column of $\widetilde{A}_k$. Thus, we may choose any non-zero entry in $\widetilde{A}_k$ to perform elimination. The numerically stable strategy is to choose the value that has the largest magnitude.
Choosing an entry that is not $y_{1,1}$ doesn't presere the lower-upper triangular geometric configuration of the resulting factorization, unless we permute the columns and rows of A so that the permuted matrix has the entry we desired in the $(1,1)$ position of $\widetilde{A}_k$. This permutation to maximize the $(1,1)$ entry is called pivoting. Here are three popular forms of pivoting:

- Full pivoting: Find the maximum-magnitude entry of $\widetilde{A}_k$. Define P and Q as row and column permutation matrices, respectively, so that the $(1,1)$ entry of $P\widetilde{A}_kQ$ is this maximum-magnitude entry.

- Partial pivoting: Find the maximum-magnitude entry of the first column of $\widetilde{A}_k$. Define P as a row permutation matrix so that the $(1,1)$ entry of $P\widetilde{A}_k$ is this maximum-magnitude entry.
- Rook pivoting: Find the maximum-magnitude entry among both the first row and column of $\widetilde{A}_k$. Define P and Q as row and column permutation matrices, respectively, so that the $(1,1)$ entry of $P\widetilde{A}_kQ$ is this maximum-magnitude entry.

The most popular version is partial pivoting since this is the least expensive of the above options and produces stable results. The more expensive full pivoting is the ideal choice, but is expensive and the observed marginal benefit over partial pivoting is small. Rook pivoting is a sort of compromise between the two methods.
Let us concentrate on partial pivoting: at each stage k, a row permutation matrix P_k is defined that operates on the full matrix A, and collecting all these operations, we have

$$P_{n-1}P_{n-2}\cdots P_2P_1A \coloneqq PA = LU.$$

One can show that the matrix P is itself a permutation matrix. What is more difficult ot discern from the above is that such a partial pivoting strategy allows us to complete the LU factorization process without running into the problems we encountered with standard Gaussian elimination:

Theorem 1. *Let A be any invertible $n \times n$ matrix. Then the LU procedure with partial pivoting always terminates successfully with the decomposition*

$$PA = LU.$$