Lecture 21. Pivoting

In the last lecture we saw that Gaussian elimination in its pure form is unstable. The instability can be controlled by permuting the order of the rows of the matrix being operated on, an operation called *pivoting*. Pivoting has been a standard feature of Gaussian elimination computations since the 1950s.

Pivots

At step k of Gaussian elimination, multiples of row k are subtracted from rows $k+1,\ldots,m$ of the working matrix X in order to introduce zeros in entry k of these rows. In this operation row k, column k, and especially the entry x_{kk} play special roles. We call x_{kk} the *pivot*. From every entry in the submatrix $X_{k+1:m,k:m}$ is subtracted the product of a number in row k and a number in column k, divided by x_{kk}:

$$\begin{bmatrix} \times & \times & \times & \times & \times \\ & x_{kk} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \end{bmatrix} \longrightarrow \begin{bmatrix} \times & \times & \times & \times & \times \\ & x_{kk} & \times & \times & \times \\ & \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ & \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ & \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \end{bmatrix}.$$

However, there is no reason why the kth row and column must be chosen for the elimination. For example, we could just as easily introduce zeros in column k by adding multiples of some row i with $k < i \le m$ to the other rows

$k, \ldots, m$. In this case, the entry x_{ik} would be the pivot. Here is an illustration with $k = 2$ and $i = 4$:

$$\begin{bmatrix} \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & x_{ik} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ & \times & \times & \times & \times \end{bmatrix} \longrightarrow \begin{bmatrix} \times & \times & \times & \times & \times \\ & \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ & \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ & x_{ik} & \times & \times & \times \\ & \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \end{bmatrix}.$$

Similarly, we could introduce zeros in column j rather than column k. Here is an illustration with $k = 2$, $i = 4$, $j = 3$:

$$\begin{bmatrix} \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \boldsymbol{\times} & x_{ij} & \boldsymbol{\times} & \boldsymbol{\times} \\ & \times & \times & \times & \times \end{bmatrix} \longrightarrow \begin{bmatrix} \times & \times & \times & \times & \times \\ & \boldsymbol{\times} & \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} \\ & \boldsymbol{\times} & \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} \\ & \times & x_{ij} & \times & \times \\ & \boldsymbol{\times} & \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} \end{bmatrix}.$$

All in all, we are free to choose any entry of $X_{k:m,k:m}$ as the pivot, as long as it is nonzero. The possibility that an entry $x_{kk} = 0$ might arise implies that some flexibility of choice of the pivot may sometimes be necessary, even from a pure mathematical point of view. For numerical stability, however, it is desirable to pivot even when x_{kk} is nonzero if there is a larger element available. In practice, it is common to pick as pivot the largest number among a set of entries being considered as candidates.

The structure of the elimination process quickly becomes confusing if zeros are introduced in arbitrary patterns through the matrix. To see what is going on, we want to retain the triangular structure described in the last lecture, and there is an easy way to do this. We shall not think of the pivot x_{ij} as left in place, as in the illustrations above. Instead, at step k, we shall imagine that the rows and columns of the working matrix are permuted so as to move x_{ij} into the (k, k) position. Then, when the elimination is done, zeros are introduced into entries $k+1, \ldots, m$ of column k, just as in Gaussian elimination without pivoting. This interchange of rows and perhaps columns is what is usually thought of as *pivoting*.

The idea that rows and columns are interchanged is indispensable conceptually. Whether it is a good idea to interchange them physically on the computer is less clear. In some implementations, the data in computer memory are indeed swapped at each pivot step. In others, an equivalent effect is achieved by indirect addressing with permuted index vectors. Which approach is best varies from machine to machine and depends on many factors.

Partial Pivoting

If every entry of $X_{k:m,k:m}$ is considered as a possible pivot at step k, there are $O((m-k)^2)$ entries to be examined to determine the largest. Summing over

m steps, the total cost of selecting pivots becomes $O(m^3)$ operations, adding significantly to the cost of Gaussian elimination, not to mention the potential difficulties of global communication in an unpredictable pattern across all the entries of a matrix. This expensive strategy is called *complete pivoting*.

In practice, equally good pivots can be found by considering a much smaller number of entries. The standard method for doing this is *partial pivoting*. Here, only rows are interchanged. The pivot at each step is chosen as the largest of the $m-k+1$ subdiagonal entries in column k, incurring a total cost of only $O(m-k)$ operations for selecting the pivot at each step, hence $O(m^2)$ operations overall. To bring the kth pivot into the (k,k) position, no columns need to be permuted; it is enough to swap row k with the row containing the pivot.

$$\underset{\text{Pivot selection}}{\begin{bmatrix} \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & x_{ik} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ & \times & \times & \times & \times \end{bmatrix}} \xrightarrow{P_1} \underset{\text{Row interchange}}{\begin{bmatrix} \times & \times & \times & \times & \times \\ & x_{ik} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ & \times & \times & \times & \times \\ & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ & \times & \times & \times & \times \end{bmatrix}} \xrightarrow{L_1} \underset{\text{Elimination}}{\begin{bmatrix} \times & \times & \times & \times & \times \\ & x_{ik} & \times & \times & \times \\ & \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ & \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ & \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \end{bmatrix}}.$$

As usual in numerical linear algebra, this algorithm can be expressed as a matrix product. We saw in the last lecture that an elimination step corresponds to left-multiplication by an elementary lower-triangular matrix L_k. Partial pivoting complicates matters by applying a permutation matrix P_k on the left of the working matrix before each elimination. (A permutation matrix is a matrix with 0 everywhere except for a single 1 in each row and column. That is, it is a matrix obtained from the identity by permuting rows or columns.) After $m-1$ steps, A becomes an upper-triangular matrix U:

$$L_{m-1}P_{m-1}\cdots L_2P_2L_1P_1A = U. \tag{21.1}$$

Example

To see what is going on, it will be helpful to return to the numerical example (20.3) of the last lecture,

$$A = \begin{bmatrix} 2 & 1 & 1 & 0 \\ 4 & 3 & 3 & 1 \\ 8 & 7 & 9 & 5 \\ 6 & 7 & 9 & 8 \end{bmatrix}. \tag{21.2}$$

With partial pivoting, the first thing we do is interchange the first and third rows (left-multiplication by P_1):

$$\begin{bmatrix} & & 1 & \\ & 1 & & \\ 1 & & & \\ & & & 1 \end{bmatrix}\begin{bmatrix} 2 & 1 & 1 & 0 \\ 4 & 3 & 3 & 1 \\ 8 & 7 & 9 & 5 \\ 6 & 7 & 9 & 8 \end{bmatrix} = \begin{bmatrix} 8 & 7 & 9 & 5 \\ 4 & 3 & 3 & 1 \\ 2 & 1 & 1 & 0 \\ 6 & 7 & 9 & 8 \end{bmatrix}.$$

The first elimination step now looks like this (left-multiplication by L_1):

$$\begin{bmatrix} 1 & & & \\ -\frac{1}{2} & 1 & & \\ -\frac{1}{4} & & 1 & \\ -\frac{3}{4} & & & 1 \end{bmatrix}\begin{bmatrix} 8 & 7 & 9 & 5 \\ 4 & 3 & 3 & 1 \\ 2 & 1 & 1 & 0 \\ 6 & 7 & 9 & 8 \end{bmatrix} = \begin{bmatrix} 8 & 7 & 9 & 5 \\ & -\frac{1}{2} & -\frac{3}{2} & -\frac{3}{2} \\ & -\frac{3}{4} & -\frac{5}{4} & -\frac{5}{4} \\ & \frac{7}{4} & \frac{9}{4} & \frac{17}{4} \end{bmatrix}.$$

Now the second and fourth rows are interchanged (multiplication by P_2):

$$\begin{bmatrix} 1 & & & \\ & & & 1 \\ & & 1 & \\ & 1 & & \end{bmatrix}\begin{bmatrix} 8 & 7 & 9 & 5 \\ & -\frac{1}{2} & -\frac{3}{2} & -\frac{3}{2} \\ & -\frac{3}{4} & -\frac{5}{4} & -\frac{5}{4} \\ & \frac{7}{4} & \frac{9}{4} & \frac{17}{4} \end{bmatrix} = \begin{bmatrix} 8 & 7 & 9 & 5 \\ & \frac{7}{4} & \frac{9}{4} & \frac{17}{4} \\ & -\frac{3}{4} & -\frac{5}{4} & -\frac{5}{4} \\ & -\frac{1}{2} & -\frac{3}{2} & -\frac{3}{2} \end{bmatrix}.$$

The second elimination step then looks like this (multiplication by L_1):

$$\begin{bmatrix} 1 & & & \\ & 1 & & \\ & \frac{3}{7} & 1 & \\ & \frac{2}{7} & & 1 \end{bmatrix}\begin{bmatrix} 8 & 7 & 9 & 5 \\ & \frac{7}{4} & \frac{9}{4} & \frac{17}{4} \\ & -\frac{3}{4} & -\frac{5}{4} & -\frac{5}{4} \\ & -\frac{1}{2} & -\frac{3}{2} & -\frac{3}{2} \end{bmatrix} = \begin{bmatrix} 8 & 7 & 9 & 5 \\ & \frac{7}{4} & \frac{9}{4} & \frac{17}{4} \\ & & -\frac{2}{7} & \frac{4}{7} \\ & & -\frac{6}{7} & -\frac{2}{7} \end{bmatrix}.$$

Now the third and fourth rows are interchanged (multiplication by P_3):

$$\begin{bmatrix} 1 & & & \\ & 1 & & \\ & & & 1 \\ & & 1 & \end{bmatrix}\begin{bmatrix} 8 & 7 & 9 & 5 \\ & \frac{7}{4} & \frac{9}{4} & \frac{17}{4} \\ & & -\frac{2}{7} & \frac{4}{7} \\ & & -\frac{6}{7} & -\frac{2}{7} \end{bmatrix} = \begin{bmatrix} 8 & 7 & 9 & 5 \\ & \frac{7}{4} & \frac{9}{4} & \frac{17}{4} \\ & & -\frac{6}{7} & -\frac{2}{7} \\ & & -\frac{2}{7} & \frac{4}{7} \end{bmatrix}.$$

The final elimination step looks like this (multiplication by L_3):

$$\begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & -\frac{1}{3} & 1 \end{bmatrix}\begin{bmatrix} 8 & 7 & 9 & 5 \\ & \frac{7}{4} & \frac{9}{4} & \frac{17}{4} \\ & & -\frac{6}{7} & -\frac{2}{7} \\ & & -\frac{2}{7} & \frac{4}{7} \end{bmatrix} = \begin{bmatrix} 8 & 7 & 9 & 5 \\ & \frac{7}{4} & \frac{9}{4} & \frac{17}{4} \\ & & -\frac{6}{7} & -\frac{2}{7} \\ & & & \frac{2}{3} \end{bmatrix}.$$

$PA = LU$ Factorization and a Third Stroke of Luck

Have we just computed an LU factorization of A? Not quite, but almost. In fact, we have computed an LU factorization of PA, where P is a permutation matrix. It looks like this:

$$\underbrace{\begin{bmatrix} & & 1 & \\ & & & 1 \\ & 1 & & \\ 1 & & & \end{bmatrix}}_{P}\underbrace{\begin{bmatrix} 2 & 1 & 1 & 0 \\ 4 & 3 & 3 & 1 \\ 8 & 7 & 9 & 5 \\ 6 & 7 & 9 & 8 \end{bmatrix}}_{A} = \underbrace{\begin{bmatrix} 1 & & & \\ \frac{3}{4} & 1 & & \\ \frac{1}{2} & -\frac{2}{7} & 1 & \\ \frac{1}{4} & -\frac{3}{7} & \frac{1}{3} & 1 \end{bmatrix}}_{L}\underbrace{\begin{bmatrix} 8 & 7 & 9 & 5 \\ & \frac{7}{4} & \frac{9}{4} & \frac{17}{4} \\ & & -\frac{6}{7} & -\frac{2}{7} \\ & & & \frac{2}{3} \end{bmatrix}}_{U}. \tag{21.3}$$

This formula should be compared with (20.5). The presence of integers there and fractions here is not a general distinction, but an artifact of our choice of A. The distinction that matters is that here, all the subdiagonal entries of L are ≤ 1 in magnitude, a consequence of the property $|x_{kk}| = \max_j |x_{jk}|$ in (20.6) introduced by pivoting.

It is not obvious where (21.3) comes from. Our elimination process took the form

$$L_3P_3L_2P_2L_1P_1A = U,$$

which doesn't look lower-triangular at all. But here, a third stroke of good fortune has come to our aid. These six elementary operations can be reordered in the form

$$L_3P_3L_2P_2L_1P_1 = L_3'L_2'L_1'P_3P_2P_1, \tag{21.4}$$

where L_k' is equal to L_k but with the subdiagonal entries permuted. To be precise, define

$$L_3' = L_3, \quad L_2' = P_3L_2P_3^{-1}, \quad L_1' = P_3P_2L_1P_2^{-1}P_3^{-1}.$$

Since each of these definitions applies only permutations P_j with $j > k$ to L_k, it is easily verified that L_k' has the same structure as L_k. Computing the product of the matrices L_k' reveals

$$L_3'L_2'L_1'P_3P_2P_1 = L_3(P_3L_2P_3^{-1})(P_3P_2L_1P_2^{-1}P_3^{-1})P_3P_2P_1 = L_3P_3L_2P_2L_1P_1,$$

as in (21.4).

In general, for an $m \times m$ matrix, the factorization (21.1) provided by Gaussian elimination with partial pivoting can be written in the form

$$(L_{m-1}' \cdots L_2'L_1')(P_{m-1} \cdots P_2P_1)A = U, \tag{21.5}$$

where L_k' is defined by

$$L_k' = P_{m-1} \cdots P_{k+1}L_kP_{k+1}^{-1} \cdots P_{m-1}^{-1}. \tag{21.6}$$

The product of the matrices L'_k is unit lower-triangular and easily invertible by negating the subdiagonal entries, just as in Gaussian elimination without pivoting. Writing $L = (L'_{m-1} \cdots L'_2 L'_1)^{-1}$ and $P = P_{m-1} \cdots P_2 P_1$, we have

$$PA = LU. \tag{21.7}$$

In general, any square matrix A, singular or nonsingular, has a factorization (21.7), where P is a permutation matrix, L is unit lower-triangular with lower-triangular entries ≤ 1 in magnitude, and U is upper-triangular. Partial pivoting is such a universal practice that this factorization is usually known simply as an *LU factorization* of A.

The famous formula (21.7) has a simple interpretation. Gaussian elimination with partial pivoting is equivalent to the following procedure:

1. Permute the rows of A according to P
2. Apply Gaussian elimination without pivoting to PA.

Partial pivoting is not carried out this way in practice, of course, since P is not known ahead of time.

Here is a formal statement of the algorithm.

Algorithm 21.1. Gaussian Elimination with Partial Pivoting

```
$U = A,\ L = I,\ P = I$
for $k = 1$ to $m - 1$
    Select $i \geq k$ to maximize $|u_{ik}|$
    $u_{k,k:m} \leftrightarrow u_{i,k:m}$   (interchange two rows)
    $\ell_{k,1:k-1} \leftrightarrow \ell_{i,1:k-1}$
    $p_{k,:} \leftrightarrow p_{i,:}$                We do not need to get L
    for $j = k + 1$ to $m$
        $\ell_{jk} = u_{jk}/u_{kk}$
        $u_{j,k:m} = u_{j,k:m} - \ell_{jk} u_{k,k:m}$
```

To leading order, this algorithm requires the same number of floating point operations (20.8) as Gaussian elimination without pivoting, namely, $\frac{2}{3}m^3$. As with Algorithm 20.1, the use of computer memory can be minimized if desired by overwriting U and L into the same array used to store A.

In practice, of course, P is not represented explicitly as a matrix. The rows are swapped at each step, or an equivalent effect is achieved via a permutation vector, as indicated earlier.

Complete Pivoting

In complete pivoting, the selection of pivots takes a significant amount of time. In practice this is rarely done, because the improvement in stability is marginal. However, we shall outline how the algebra changes in this case.

In matrix form, complete pivoting precedes each elimination step with a permutation P_k of the rows applied on the left and also a permutation Q_k of the columns applied on the right:

$$L_{m-1}P_{m-1}\cdots L_2P_2L_1P_1AQ_1Q_2\cdots Q_{m-1} \;=\; U. \tag{21.8}$$

Once again, this is not quite an LU factorization of A, but it is close. If the L_k' are defined as in (21.6) (the column permutations are not involved), then

$$(L_{m-1}'\cdots L_2'L_1')(P_{m-1}\cdots P_2P_1)A(Q_1Q_2\cdots Q_{m-1}) = U. \tag{21.9}$$

Setting $L = (L_{m-1}'\cdots L_2'L_1')^{-1}$, $P = P_{m-1}\cdots P_2P_1$, and $Q = Q_1Q_2\cdots Q_{m-1}$, we obtain

$$PAQ = LU. \tag{21.10}$$

Exercises

21.1. Let A be the 4×4 matrix (20.3) considered in this lecture and the previous one.

(a) Determine $\det A$ from (20.5).

(b) Determine $\det A$ from (21.3).

(c) Describe how Gaussian elimination with partial pivoting can be used to find the determinant of a general square matrix.

21.2. Suppose $A\in\mathbb{C}^{m\times m}$ is banded with bandwidth $2p+1$, as in Exercise 20.2, and a factorization $PA = LU$ is computed by Gaussian elimination with partial pivoting. What can you say about the sparsity patterns of L and U?

21.3. Consider Gaussian elimination carried out with pivoting by columns instead of rows, leading to a factorization $AQ = LU$, where Q is a permutation matrix.

(a) Show that if A is nonsingular, such a factorization always exists.

(b) Show that if A is singular, such a factorization does not always exist.

21.4. Gaussian elimination can be used to compute the inverse A^{-1} of a nonsingular matrix $A\in\mathbb{C}^{m\times m}$, though it is rarely really necessary to do so.

(a) Describe an algorithm for computing A^{-1} by solving m systems of equations, and show that its asymptotic operation count is $8m^3/3$ flops.

(b) Describe a variant of your algorithm, taking advantage of sparsity, that reduces the operation count to $2m^3$ flops.

(c) Suppose one wishes to solve n systems of equations $Ax_j = b_j$, or equivalently, a block system $AX = B$ with $B \in \mathbb{C}^{m\times n}$. What is the asymptotic operation count (a function of m and n) for doing this (i) directly from the LU factorization and (ii) with a preliminary computation of A^{-1}?

21.5. Suppose $A \in \mathbb{C}^{m\times m}$ is hermitian, or in the real case, symmetric (but not necessarily positive definite).

(a) Describe a strategy of *symmetric pivoting* to preserve the hermitian structure while still leading to a unit lower-triangular matrix with entries $|\ell_{ij}| \le 1$.

(b) What is the form of the matrix factorization computed by your algorithm?

(c) What is its asymptotic operation count?

21.6. Suppose $A \in \mathbb{C}^{m\times m}$ is *strictly column diagonally dominant*, which means that for each k,

$$|a_{kk}| > \sum_{j\ne k} |a_{jk}|. \tag{21.11}$$

Show that if Gaussian elimination with partial pivoting is applied to A, no row interchanges take place.

21.7. In Lecture 20 the "two strokes of luck" were explained by the use of the vectors e_k and ℓ_k. Give an explanation based on these vectors for the "third stroke of luck" in the present lecture.