2 Partial pivoting, LU factorization

2.1 An example

We emphasize again, that all the calculations in the example in Section 1.1.3 were exact, i.e. the amplification of errors was not due to numerical inaccuracy; the given system was *ill-conditioned*. There is not much one could do about it on the mathematical or even computational level, apart from warning the user about the problem. In the worse case one could ask for more accurate data, but it is much better if the modelling of the applied problem is changed to a better conditioned problem.

The following example, however, shows that even a well-conditioned problem can be made unstable by a carelessly designed algorithm. And this is the responsibility of the computational expert.

Suppose that we use a floating point arithmetic with all numbers rounded to three significant digits (i.e. to facilitate the idea we use decimal system). Let

$$A = \begin{pmatrix} 10^{-4} & 1 \\ 1 & 1 \end{pmatrix} \qquad \text{and} \qquad \mathbf{b} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

and consider the system $A\mathbf{x} = \mathbf{b}$. The solution of this system is

$$\mathbf{x}_{true} \approx \begin{pmatrix} 1.00010\ldots \\ 0.99990\ldots \end{pmatrix} \approx \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

Now we apply Gaussian elimination. It is possible to take 10^{-4} as a pivot since it is a nonzero number. The elimination step gives

$$\left(\begin{array}{cc|c} 10^{-4} & 1 & 1 \\ 1 & 1 & 2 \end{array}\right) \Longrightarrow \left(\begin{array}{cc|c} 10^{-4} & 1 & 1 \\ 0 & -9999 & -9998 \end{array}\right)$$

However the numbers 9999 and 9998 are both rounded to the same number 9990, i.e. the computer stores only

$$\left(\begin{array}{cc|c} 10^{-4} & 1 & 1 \\ 0 & -9990 & -9990 \end{array}\right)$$

and this system has a solution

$$\mathbf{x} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

What went wrong? Based upon the example in Section 1.1.3, you might think that the problem is ill-conditioned. Compute the eigenvalues of A (DO IT(*)), and you get that

$$\text{cond}(A) \approx 2.618$$

which is not a big number. It certainly does not justify that a rounding error of relative size 10^{-3} got amplified by a factor of 1000.

On the other hand if first we interchange the two rows, and we choose the $a_{21} = 1$ as a pivot element, then we get

$$\left(\begin{array}{cc|c} 10^{-4} & 1 & 1 \\ 1 & 1 & 2 \end{array}\right) \Longrightarrow \left(\begin{array}{cc|c} 1 & 1 & 2 \\ 10^{-4} & 1 & 1 \end{array}\right) \Longrightarrow \left(\begin{array}{cc|c} 1 & 1 & 2 \\ 0 & 0.999 & 0.999 \end{array}\right)$$

since the true numbers 0.9999 and 0.9998 in the second row are both rounded to 0.999. Now this system has a solution

$$\mathbf{x} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

which is very satisfactory considering the true solution $\mathbf{x}_{true}$.

The first algorithm is an example of an *unstable* method, while the second is stable. Stability of the method expresses how much the rounding errors can accumulate. It is a different issue than conditioning. Recall that being well or ill-conditioned is a property of the *mathematical* problem and it has nothing to do with computations. The output of an ill-conditioned problem will not be reliable even with the most perfect computational methods. However, even a well-conditioned problem could be "spoiled" by an unstable method, which usually means that the reliability (relative output error versus relative input error) of the algorithm is much worse than the condition number predicts. It also means that errors that "pop up" in

the computation from rounding could get amplified along the further steps of the algorithm. Therefore, it is a crucial effort in numerical analysis to come up with methods which do not make things worse than they are.

2.2 Gaussian elimination with partial pivoting

The trouble in the example above was that the pivot was a very small number (relatively to all other numbers). Recall that along the Gauss elimination we have to divide by the pivot and dividing by small numbers in general is dangerous. But it is very easy to eliminate this problem just by a row exchange. Recall that at every step of the Gauss elimination one is allowed to choose any nonzero element in the corresponding column (of course always from *below* the fully pivoted rows).

Here is an example. Suppose we have the following form of the matrix $[A|\mathbf{b}]$ as an intermediate step:

$$\left(\begin{array}{ccccccc|c} \underline{1} & * & * & * & * & \dots & * & * \\ 0 & 0 & \underline{1} & * & * & \dots & * & * \\ 0 & 0 & 0 & p_1 & * & \dots & * & * \\ 0 & 0 & 0 & p_2 & * & \dots & * & * \\ \vdots & \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & p_m & * & \dots & * & * \end{array}\right)$$

In the original Gauss elimination we choose p_1 as the next pivot if it is nonzero and we swapped rows only if $p_1 = 0$. In the **Gaussian elimination with partial pivoting** we bring the row with the biggest p_i in absolutely value into the pivoting position, i.e. if $|p_i| = \max_{j=1,2,\dots m} |p_j|$, then we swap the row of p_1 with the row of p_i and use p_i as a pivot.

This simple idea results in a dramatic improvement in the stability of the Gaussian elimination. Later we will learn better but more complicated methods to replace Gaussian elimination. But whenever Gaussian elimination is used in applications, its stability should always

be improved at least by this very simple "trick" of partial pivoting.

2.3 LU factorization

In order to understand better why partial pivoting worked so well and to develop further improvements, we need a deeper insight into Gaussian elimination.

For simplicity of the discussion we assume that A is a regular square matrix. All concepts have analogues for the general case, but we wish only to highlight the basic idea.

Furthermore, we assume that during the Gaussian elimination no row exchange was needed, i.e. after pivoting the $(k-1)$-th column, the element a_{kk} is not zero, hence it can be used as a pivot. This can always be achieved by rearranging the rows of the matrix in advance. [In parenthesis: of course it is hard to see whether rearranging is necessary without actually doing the elimination. In fact you never have to swap rows during the elimination of a square matrix A if and only if all the upper-left square submatrices of A are regular. (PROVE IT(*))]

Finally, we will not insist on ones in the pivot position. Recall that there are two equivalent conventions during Gauss elimination: either one always divides each row by the pivot or not. Here we use the second convention, which is used by [D].

Suppose we are at the following stage of the elimination (for simplicity we omitted the augmented **b** part)

$$\begin{pmatrix} \underline{*} & * & * & * & * & \dots & * \\ 0 & \underline{*} & * & * & * & \dots & * \\ 0 & 0 & \underline{p_{33}} & * & * & \dots & * \\ 0 & 0 & 0 & * & * & \dots & * \\ 0 & 0 & 0 & * & * & \dots & * \\ 0 & 0 & p_{63} & * & * & \dots & * \\ 0 & 0 & p_{73} & * & * & \dots & * \\ \vdots & \vdots & \vdots & \vdots & \vdots & & \vdots \end{pmatrix}$$

and we wish to eliminate p_{63} by substracting $\frac{p_{63}}{p_{33}}$-times the third row from the sixth. This means that the above matrix has the be multiplied from the left by the following very simple

matrix (CHECK(*)!)

$$L_{63}\Big(-\frac{p_{63}}{p_{33}}\Big) = \begin{pmatrix} 1 & & & & & & & \\ & 1 & & & & & & \\ & & 1 & & & & & \\ & & & 1 & & & & \\ & & & & 1 & & & \\ & & -\frac{p_{63}}{p_{33}} & & & 1 & & \\ & & & & & & 1 & \\ & & & & & & & \ddots \end{pmatrix}$$

where all other elements are zero. This is a lower triangular matrix with ones in the diagonal and only one element (in the sixth row, third column) is nonzero, namely $-\frac{p_{63}}{p_{33}}$. In general, let

$$L_{ij}(z) := \begin{pmatrix} 1 & & & & & & & \\ & 1 & & & & & & \\ & & \ddots & & & & & \\ & & & 1 & & & & \\ & & & & \ddots & & & \\ & & & z & & 1 & & \\ & & & & & & \ddots & \\ & & & & & & & 1 \end{pmatrix} \tag{2.1}$$

where the nonzero element z stands in the i-th row and j-th column ($j < i$).

From the example above it is clear that Gaussian elimination (without row swapping) actually consists of multiplications from the left by matrices of the form $L_{ij}(z)$ with appropriately chosen z.

Definition 2.1 *An $n \times n$ matrix L is called* **lower triangular** *if all entries above the diagonal are zero. L is called* **unit lower triangular** *if, in addition, the diagonal elements are all ones. Similarly we define the upper triangular and unit upper triangular matrices.*

It is easy to check (CHECK(*)) that the product of two lower triangular matrices is lower triangular and the product of two unit lower triangular matrices is again unit lower triangular. Similarly for upper triangular matrices.

Finally, the unit lower triangular matrices are invertible (WHY(*)?) and in particular

$$L_{ij}(z)L_{ij}(-z) = I \tag{2.2}$$

(CHECK (*))

From all these information we have

Theorem 2.2 (LU factorization) *Let A be an $n \times n$ regular matrix such that all upper-left submatrices are regular (determinant nonzero). Then there exist a unit lower triangular matrix L and an upper triangular matrix U such that*

$$A = LU$$

This factorization is unique.

(Similar theorem is true for nonsingular or rectangular matrices, but then permutation matrices must be allowed for row-swapping and the formulation of the uniqueness statement requires more care.)

Proof: We have all ingredients. The conditions imply that there is no row exchange in the Gauss elimination. After completing the Gauss elimination, we arrive at an upper triangular matrix A, i.e.

$$L_{n,n-1}L_{n,n-2}L_{n-1,n-2}\dots L_{n2}\dots L_{42}L_{32}L_{n1}\dots L_{31}L_{21}A = U$$

for appropriately chosen elementary unit lower triangular matrices of the form (2.1). After inverting them and using that their inverses and their products are also unit lower triangular matrices, we get

$$A = LU$$

with $L = L_{21}^{-1}L_{31}^{-1}\dots L_{n1}^{-1}L_{32}^{-1}L_{42}^{-1}\dots L_{n2}^{-1}\dots L_{n-1,n-2}^{-1}L_{n,n-2}^{-1}L_{n,n-1}^{-1}$

The uniqueness follows easily: if $A = L_1U_1 = L_2U_2$ are two different decompositions, then $U_1 = LU_2$ with $L = L_1^{-1}L_2$. Since A was regular, all the diagonal entries of U_2 are nonzero (WHY(*)?). Then CHECK(*) that LU_2 can be upper triangular only if L is the identity (otherwise LU_2 has a nonzero element below the diagonal). Therefore $L_1 = L_2$, and then $U_1 = U_2$. $\square$.

Problem 2.3 *Find the LU factorization of* $A = \begin{pmatrix} 1 & 1 & -2 \\ 1 & 3 & -1 \\ 1 & 5 & 1 \end{pmatrix}$

SOLUTION: This is the same problem as the Gaussian elimination, just record the steps by recording the elementary L matrices of the form (2.1). Clearly

$$\begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 & -2 \\ 1 & 3 & -1 \\ 1 & 5 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 1 & -2 \\ 0 & 2 & 1 \\ 1 & 5 & 1 \end{pmatrix}$$

which is exactly the first step in the elimination: subtracting the first row from the second. The next step is

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 & -2 \\ 0 & 2 & 1 \\ 1 & 5 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 1 & -2 \\ 0 & 2 & 1 \\ 0 & 4 & 3 \end{pmatrix}$$

Notice that you could have done the two steps together, i.e. multiplication by one lower triangular matrix can eliminate all elements below the pivot in the first row

$$\begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 & -2 \\ 1 & 3 & -1 \\ 1 & 5 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 1 & -2 \\ 0 & 2 & 1 \\ 0 & 4 & 3 \end{pmatrix}$$

Next we use

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -2 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 & -2 \\ 0 & 2 & 1 \\ 0 & 4 & 3 \end{pmatrix} = \begin{pmatrix} 1 & 1 & -2 \\ 0 & 2 & 1 \\ 0 & 0 & 1 \end{pmatrix}$$

Altogether

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -2 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 & -2 \\ 1 & 3 & -1 \\ 1 & 5 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 1 & -2 \\ 0 & 2 & 1 \\ 0 & 0 & 1 \end{pmatrix}$$

Now we invert the lower triangular matrices (be careful with the order!)

$$\begin{pmatrix} 1 & 1 & -2 \\ 1 & 3 & -1 \\ 1 & 5 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 2 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 & -2 \\ 0 & 2 & 1 \\ 0 & 0 & 1 \end{pmatrix}$$

i.e.

$$A = \begin{pmatrix} 1 & 1 & -2 \\ 1 & 3 & -1 \\ 1 & 5 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 2 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 & -2 \\ 0 & 2 & 1 \\ 0 & 0 & 1 \end{pmatrix} = LU$$

This finishes the LU-decomposition of A.

WARNING: Notice that the inverse of a unit lower triangular matrix with only one nonzero column (apart from the diagonal) can be taken just by reversing the sign of the offdiagonal elements, i.e. not just (2.2) is true, but also

$$\begin{pmatrix} 1 & 0 & 0 & \dots \\ a & 1 & 0 & \dots \\ b & 0 & 1 & \dots \\ c & 0 & 0 & \dots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & \dots \\ -a & 1 & 0 & \dots \\ -b & 0 & 1 & \dots \\ -c & 0 & 0 & \dots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix} = I$$

(CHECK(*)). But the inverse of a full unit lower triangular matrix cannot be taken just by changing the sign:

$$\begin{pmatrix} 1 & 0 & 0 \\ a & 1 & 0 \\ b & c & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ -a & 1 & 0 \\ -b & -c & 1 \end{pmatrix} \neq I$$

2.4 Gaussian elimination revisited

Once you have an LU decomposition, you can solve $A\mathbf{x} = \mathbf{b}$. What Gaussian elimination actually does is that it inverts L, so that from $A\mathbf{x} = LU\mathbf{x} = \mathbf{b}$ we have $U\mathbf{x} = L^{-1}\mathbf{b}$, then backsolving actually finds $\mathbf{x}$. This second step is much easier, since now U is upper triangular.

In other words, we split the problem $A\mathbf{x} = LU\mathbf{x} = \mathbf{b}$ into two parts. First we solve

$$L\mathbf{y} = \mathbf{b}$$

then we solve

$$U\mathbf{x} = \mathbf{y}$$

Now we can see what went wrong in the example in Section 2.1. We started with a well conditioned problem $A\mathbf{x} = \mathbf{b}$ with $\text{cond}(A) \approx 2.61$. But the elimination actually splits it into two easier problems, which happen to be ill-conditioned. To see that, compute the LU decomposition of that example to get

$$A = \begin{pmatrix} 10^{-4} & 1 \\ 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 10000 & 1 \end{pmatrix} \begin{pmatrix} 10^{-4} & 1 \\ 0 & -9999 \end{pmatrix} = LU$$

In order to compute the condition number of L and U, we need to compute the eigenvalues of LL^t and UU^t (see Theorem 1.3). We get

$$\lambda_1(LL^t) \approx 10^8 \qquad \lambda_2(LL^t) \approx 10^{-8}$$

hence $\|L\| \approx 10^4$ and $\|L^{-1}\| \approx 10^4$, so $\text{cond}(L) \approx 10^8$. Similarly we can compute

$$\lambda_1(UU^t) \approx 10^8 \qquad \lambda_2(UU^t) \approx 10^{-8}$$

hence $\|U\| \approx 10^4$ and $\|U^{-1}\| \approx 10^4$, so $\text{cond}(U) \approx 10^8$.

So the Gaussian elimination in fact turned a well conditioned problem into two very ill-conditinioned ones.

Now let us see how the partial pivoting (row swap) helps. Then instead of A, we actually factorize

$$A' = \begin{pmatrix} 1 & 1 \\ 10^{-4} & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ -10^{-4} & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 0 & 1 + 10^{-4} \end{pmatrix} = L'U'$$

and we easily compute that

$$\text{cond}(L') \approx 1 \qquad \text{cond}(U') \approx 2.61$$

i.e. we split the problem into two very well conditioned problems. In fact we essentially did not lose anything compared to the original problem, since $\text{cond}(A) \approx 2.61$ as well.

The situation is not always as good as here. First we remark that with any factorization one loses accuracy, since

$$\text{cond}(A) = \text{cond}(LU) \leq \text{cond}(L)\text{cond}(U)$$

(PROVE IT(*)). This means that the true error amplification $\text{cond}(A)$, which is inherently present in the problem, is smaller (at most equal) than the successive error amplification when we solve $L\mathbf{y} = \mathbf{b}$ and $U\mathbf{x} = \mathbf{y}$ separately (notice the the output of the first problem is to be put into the input of the second, so the error amplifications really get multiplied, in the worst case).

Even with partial pivoting in general we have strict inequality. Consider the example

$$A = \begin{pmatrix} 1 & 1 & -2 \\ 1 & 3 & -1 \\ 1 & 5 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 2 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 & -2 \\ 0 & 2 & 1 \\ 0 & 0 & 1 \end{pmatrix} = LU$$

worked out above. Here

$$\text{cond}(A) = 84, \qquad \text{cond}(L) = 16, \qquad \text{cond}(U') = 16$$

i.e. the error amplification is 256 in the algorithm, instead of the maximal error amplification 84 of the original problem.

Notice that there was no partial pivoting, since in the last step we actually used 2 as a pivot instead of swapping the second and third rows (see Problem 2.3).

Consider now

$$A' = \begin{pmatrix} 1 & 1 & -2 \\ 1 & 5 & 1 \\ 1 & 3 & -1 \end{pmatrix}$$

i.e., where we already swapped the two rows. It is easy to see that in each step of the elimination the entry on the diagonal is the biggest possible pivot, hence this is really the same as a Gaussian elimination with partial pivoting for A.

Compute the LU-decomposition

$$A' = \begin{pmatrix} 1 & 1 & -2 \\ 1 & 5 & 1 \\ 1 & 3 & -1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 0.5 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 & -2 \\ 0 & 4 & 3 \\ 0 & 0 & -0.5 \end{pmatrix} = L'U'$$

Now

$$\text{cond}(A') = 84, \qquad \text{cond}(L') = 5, \qquad \text{cond}(U') = 47.25$$

hence error amplification of second method is 236, in contrast to 84, which is the error amplification of the posed problem.

We can see that partial pivoting helped a bit (we have 236 instead of 256), but not too much. The effect is not so spectacular as in the example of Section 2.1. However, since partial pivoting is very easy to implement, it is always recommended to use it.