

The Gaussian elimination algorithm

T. PERUTZ

These notes are to be studied *after* you have read Section 2.1 of Andrilli–Hecker, and *after* you have worked through some examples of Gaussian elimination as it is presented there. The purpose of the notes is to explain how things work *systematically*.

1 Row operations and row echelon form

Let A be an $m \times n$ matrix.

Definition 1 A *row operation* on A does one of the following things:

- I. Scales one of the rows (the i th, say) by a non-zero number c . Symbolically,

$$\text{row}(i) \leftarrow c \text{row}(i).$$

[The notation means: take $c \text{row}(i)$, and rename that as $\text{row}(i)$.]

- II. Adds s times the q th row to the p th row (for some p, q between 1 and m , and some $s \in \mathbb{R}$):

$$\text{row}(p) \leftarrow \text{row}(p) + s \text{row}(q).$$

- III. Interchanges the p th and q th rows:

$$\text{row}(p) \leftrightarrow \text{row}(q).$$

Notice that row operations are *reversible*: if A' is obtained from A by a row operation, then A can be obtained from A' by another row operation (prove it!).

Definition 2 An $m \times n$ matrix $B = (b_{ij})$ is in *row echelon form of row-rank R* (for some integer R with $0 \leq R \leq m$) if the following three conditions hold.

- The first R rows contain a non-zero entry, while the remaining $m - R$ are zero.

For $i = 1, \dots, R$, let $(i, \ell(i))$ be the leading entry of row i . That is, $a_{i, \ell(i)} \neq 0$ while $a_{ij} = 0$ for $1 \leq j < \ell(i)$.

- $a_{i, \ell(i)} = 1$ for each i .
- $\ell(1) < \ell(2) < \dots < \ell(R)$.

Exercise 3 Draw some 3×4 matrices in row echelon form, as well as some not in echelon form.

2 Gaussian elimination

The *Gaussian elimination algorithm* inputs an $m \times n$ matrix B and performs a sequence of row operations on it. The output is an $m \times n$ matrix E in row echelon form.

When we have set up Gaussian elimination, we'll have proved the following theorem:

Theorem 4 *Let B be any matrix. Then there is a finite sequence of row operations which transforms B into a matrix E in row echelon form. In fact, there is a mechanical procedure which determines E from the input B .*

Remark The procedure is 'canonical', that is, it does not involve any undetermined choices. Thus we can think of the row echelon matrix $E = E(B)$ as a function of B .

2.1 The n steps

Let us say a matrix B is in j -echelon form if the matrix formed by the first j columns of B is in row echelon form. So, an $m \times n$ matrix in n -echelon form is actually in row echelon form. We consider any matrix to be in 0-echelon form.

Gaussian elimination is an n -step process. The input for step 1 is matrix $B^{(0)} := B$. The output for step 1 is a new matrix $B^{(1)}$ in 1-echelon form. This is the input for step 2. The output for step 2 is a new matrix $B^{(2)}$ in 2-echelon form. The input for the j th step is a matrix $B^{(j-1)}$ in $(j-1)$ -echelon form, while the output for the j th step is a matrix $B^{(j)}$ in j -echelon form.

2.2 What happens at the j th step

At the j th step, we are given a matrix $B^{(j-1)}$ in $(j-1)$ -echelon form. Its row-rank is $R^{(j-1)}$. We examine the j th column of $B^{(j-1)}$. More precisely, we look at the entries below the $R^{(j-1)}$ th row, i.e., the entries

$$b_{R^{(j-1)}+1,j}, b_{R^{(j-1)}+2,j}, \dots, R_{m,j}.$$

We ask: *are these entries all zero?*

If yes: the matrix $B^{(j-1)}$ is already in j -echelon form. We put $B^{(j)} = B^{(j-1)}$ and $R^{(j)} = R^{(j-1)}$ and move on to the next j .

If no: we go on to the swapping procedure.

2.3 Swapping

Suppose that at least one of the entries

$$b_{R^{(j-1)+1,j}}, b_{R^{(j-1)+2,j}, \dots, R_{m,j}}$$

is non-zero. Let b_{ij} be the first one which is non-zero. We perform a Type III row operation

$$\text{row}(i) \leftrightarrow \text{row}(R^{(j-1)+1})$$

so as to bring the row containing b_{ij} just below the rows which have a non-zero entry further left than the j th column. Move on to the pivoting procedure.

2.4 Pivoting

We're already focusing our attention on the j th column. We'll call this the pivot column. Let $p = R^{(j-1)} + 1$. Call $\text{row}(p)$ the pivot row, and the entry (p, j) the pivot entry. Notice that $a_{p,j}$ is non-zero! Use a Type I operation

$$\text{row}(p) \leftarrow a_{p,j}^{-1} \text{row}(p)$$

to make it 1. Now kill all the entries below the pivot entry by using Type II operations to subtract off multiples of the pivot row:

$$\text{row}(q) \leftarrow \text{row}(q) - a_{qj} \text{row}(p), \quad p < q \leq m.$$

Call the resulting matrix $B^{(j)}$. Let $R^{(j)} = R^{(j-1)} + 1$. End of the j th step.

2.5 Why swapping/pivoting worked

Observe that $B^{(j)}$ is in j -echelon form of rank $R^{(j)} = R^{(j-1)} + 1$. Why so? If we look at the matrix C formed by its first j columns then the first $R^{(j)}$ rows of C are non-zero (we didn't touch the first $R^{(j-1)}$ rows, so these are already non-zero; and we made sure by the swapping procedure that the $R^{(j-1)} + 1$ th row is also non-zero.) The last $m - R^{(j)}$ rows of C are zero as a result of pivoting. The leading entry in the $R^{(j)}$ th row of C is in the j th row—further right than that of the $R^{(j-1)}$ th row, which is in the $R^{(j-1)}$ th row. This completes the argument.

We have now completed the proof of Theorem 4.

Exercise 5 Write down a 3×4 matrix of your choice. Carefully follow the Gaussian elimination algorithm to obtain a row echelon matrix E . Did you use any Type III row operations? If not, write down another matrix for which the algorithm does use Type III operations. Do the same with a 4×3 matrix.

Exercise 6 How long does it take a computer to perform Gaussian elimination?

- (1) A Type I row operation $\text{row}(i) \leftarrow a_{ij}^{-1} \text{row}(i)$ involves one division (to calculate a_{ij}^{-1}), n multiplications, and n assignments to computer memory. What kind of operations are involved in performing a Type II and III row operation—and how many of them?
- (2) Explain why the total number of Type I operations required is at most m .
- (3) Explain why the total number of Type II operations required is at most $\frac{1}{2}m(m-1)$.
- (4) Explain why the total number of Type III operations required is at most m .
- (5) Explain why the total number of times the computer has to check whether an entry is zero is at most mn .
- (6) Deduce that the total number of additions, multiplications, divisions, comparisons to zero and assignments to memory is bounded above by

$$m^2n + \text{lower order terms.}$$

(The ‘lower order terms’ include terms such as mn .)

A computer that can perform, say, 10^{10} operations per second has a good chance of performing Gaussian elimination on a 1000×1000 matrix of (smallish) integers in reasonable time. But Gaussian elimination on a million-by-million matrix would take far too long.