

Handout 1

Systems of linear equations

Gaussian elimination

1. Linear equations. An equation can have more than one **unknown**, as in

$$x^2 + y^2 = 5$$

An equation with one or more unknowns is called **linear** if the unknowns appear on their own, without exponent (that is to say, with exponent 1). In other words, products of unknowns are not allowed. The example above is not linear (but “quadratic”) because it contains the product of x with itself and also the product of y with itself. The following *is* a linear equation (with three unknowns)

$$3x - 2y + z = 7$$

A general linear equation with n unknowns can be written as

$$a_1x_1 + a_2x_2 + \dots + a_nx_n = b$$

where it is understood that the x 's are the unknowns, the a 's and b are **placeholders** or **parameters** or **coefficients** which in any concrete instance would be actual numbers.

2. Systems of equations. A linear equation which contains more than one variable will usually have many solutions. For example, the equation

$$2x - y = 0$$

has the solutions $x = 1, y = 2$, and $x = 2, y = 4$, and $x = 3, y = 6$, and so on. In order to pin this down to *exactly one solution* we need *as many equations as there are unknowns*. For example, the system of linear equations

$$\begin{aligned} 2x - y &= 0 \\ x + y &= 6 \end{aligned}$$

has only *one* assignment for x and y which makes *both* equations true *simultaneously*, namely, $x = 2, y = 4$.

3. Solving systems of linear equations: Gaussian elimination. In the language of Computer Science, Gaussian elimination is best described as a *recursive algorithm*. It has a **base case** and a **general case**:

Base Case There is only one equation and one unknown: $ax = b$. This can be solved directly: $x = b/a$.

General Case There are n equations and each contains n unknowns: Then we use the first equation to **eliminate** the first unknown in the remaining $n - 1$ equations. Those new $n - 1$ equations then only contain $n - 1$ unknowns and Gaussian elimination can be applied recursively to those.

Once Gaussian elimination returns the values for those $n - 1$ unknowns, they can be substituted into the first equation in which then only one unknown remains. This can be solved with the Base Case.

4. An example.

1	$\begin{aligned} x_1 + 5x_2 - 2x_3 &= -11 \\ 3x_1 - 2x_2 + 7x_3 &= 5 \\ -2x_1 - x_2 - x_3 &= 0 \end{aligned}$
---	---

This is the General Case as there is more than one unknown. We use equation 1 to eliminate x_1 from equations 2 and 3. We do this by subtracting 3 times equation 1 from equation 2, and get:

2

Likewise, we add twice line 1 to line 3:

3

Now we apply Gaussian elimination recursively to the smaller system consisting of equations 2 and 3 which each contain two unknowns, and temporarily forget about equation 1.

Once more we have to follow the General Case: we use the second equation to eliminate x_2 in the third. For this we add 9 times equation 2 to 17 times equation 3:

4

We get one equation (the third one) with *one* unknown. It can be solved by the Base Case: $x_3 = -32/32 = -1$.

Now we go back one step in the recursion and substitute $x_3 = -1$ in the second equation and get:

5

The second equation can now be solved by the Base Case and we get $x_2 = (38 + 13)/(-17) = -3$.

We return to the very first step in the algorithm and replace x_2 and x_3 in the first equation by the values that we have computed:

6

Simplifying the first equation gives $x_1 = -11 - 2 + 15 = 2$. All three unknowns have been computed; the algorithm stops.

5. **A more compact notation.** The unknowns themselves are not affected by Gaussian elimination as all computation happens on the numeric coefficients in front of them. This suggest to only write out the coefficients and leave the unknowns implicit. In our example, the picture looks like this

7

$$\left(\begin{array}{ccc|c} 1 & 5 & -2 & -11 \\ 3 & -2 & 7 & 5 \\ -2 & -1 & -1 & 0 \end{array} \right)$$

We include the vertical line to remind us that the last column refers to the right hand sides of the given equations, not to another unknown.

Rather than go through the same example again, we do Gaussian elimination on a second example in this new notation.

8

$$\left(\begin{array}{ccc|c} 3 & 0 & 5 & 3 \\ -3 & 2 & -2 & -5 \\ 3 & -2 & 7 & 5 \end{array} \right)$$

Eliminating the first variable in equations 2 and 3:

9

Eliminating the second variable in equation 3:

10

Simplifying the last equation:

11

We can now read off that $x_3 = 0$ and as described above, use this value in the first two equations, etc. Alternatively, we can continue working in the compact notation: Using the last line we eliminate the entries in the third column in lines one and two.

12

The entry above “2” is zero already so that step is already done for us. We finish by dividing each line so as to get ones on the diagonal:

13

We can read off the solution by putting back the unknowns:

14

$$\begin{pmatrix} x_1 & = & 1 \\ x_2 & = & -1 \\ x_3 & = & 0 \end{pmatrix}$$

6. Practical advice. In the exam, I expect you to be able to:

- Translate from the explicit notation with unknowns (as in Box 1) to the compact notation introduced in Item 5, Box 7, and back.
- Apply Gaussian elimination using the compact notation.

Notes

- Every unknown corresponds to precisely one column of entries in the compact notation. The last column corresponds to the numbers on the right-hand side of the equations.
- Be careful with the signs of the coefficients when translating, and when copying from one step in the algorithm to the next.
- Always work with the compact notation, not the explicit equations containing variables or you will inevitably get confused.
- Neat work will help you avoid common mistakes, such as dropping a minus sign and mis-copying entries.
- Avoid fractions until the very end. For example, in

$$\left(\begin{array}{cc|c} 2 & -3 & 5 \\ 3 & 7 & -1 \end{array} \right)$$

don't rewrite to

$$\left(\begin{array}{cc|c} 2 & -3 & 5 \\ 0 & 7 - \frac{3}{2} \times (-3) & -1 - \frac{3}{2} \times 5 \end{array} \right) = \left(\begin{array}{cc|c} 2 & -3 & 5 \\ 0 & \frac{23}{2} & -\frac{17}{2} \end{array} \right)$$

by subtracting $\frac{3}{2}$ of the first line, but first multiply suitably as in

$$\left(\begin{array}{cc|c} 2 & -3 & 5 \\ 6 & 14 & -2 \end{array} \right)$$

and then subtract an *integer multiple* (here 3 times the first line):

$$\left(\begin{array}{cc|c} 2 & -3 & 5 \\ 0 & 1 - 3 \times (-3) & -2 - 3 \times 5 \end{array} \right) = \left(\begin{array}{cc|c} 2 & -3 & 5 \\ 0 & 23 & -17 \end{array} \right)$$

- Know your sign rules:
 - subtracting a negative number from a negative number:

$$-x - (-y) \text{ is the same as } -x + y \text{ which is the same as } y - x$$

- multiplying negative numbers:

$$(-x) \times (-y) \text{ is the same as } x \times y$$

- Don't even think about using the highschool "substitution method" if there are more than two equations!
- Abstracting away from the actual values, the first phase of Gaussian elimination leads to a regular even-spaced "staircase" that separates the zeros created by the algorithm from the rest:

$$\left(\begin{array}{cccc|c} * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \end{array} \right) \rightarrow \dots \rightarrow \left(\begin{array}{cccc|c} \bullet & * & * & * & * \\ 0 & \bullet & * & * & * \\ 0 & 0 & \bullet & * & * \\ 0 & 0 & 0 & \bullet & * \end{array} \right)$$

The term often used in books for this pattern is **echelon form**. In the second phase, one creates zeros above the staircase. This can be pictured as follows:

$$\left(\begin{array}{cccc|c} \bullet & * & * & * & * \\ 0 & \bullet & * & * & * \\ 0 & 0 & \bullet & * & * \\ 0 & 0 & 0 & \bullet & * \end{array} \right) \rightarrow \left(\begin{array}{cccc|c} \bullet & * & * & 0 & * \\ 0 & \bullet & * & 0 & * \\ 0 & 0 & \bullet & 0 & * \\ 0 & 0 & 0 & 1 & * \end{array} \right) \rightarrow \left(\begin{array}{cccc|c} \bullet & * & 0 & 0 & * \\ 0 & \bullet & 0 & 0 & * \\ 0 & 0 & 1 & 0 & * \\ 0 & 0 & 0 & 1 & * \end{array} \right) \rightarrow \left(\begin{array}{cccc|c} 1 & 0 & 0 & 0 & * \\ 0 & 1 & 0 & 0 & * \\ 0 & 0 & 1 & 0 & * \\ 0 & 0 & 0 & 1 & * \end{array} \right)$$