

# Gaussian Elimination the way Humans do It

<http://marekrychlik.com/node/34>

{The software link: </cgi-bin/gauss.cgi>} Click here/a/.

{I wrote a program several year ago for the purpose of edification of my students, which produces step-by-step solutions to linear systems using Gaussian elimination. The core of the program was written in Lisp (one of my favorite programming languages) but the Web interface was done in Perl. In recent days, I rewrote the program completely and entirely in Lisp. If you are a Lisper, you may be interested in the fact that the matrix can be entered in either MATLAB or Lisp syntax, thanks to the Lisp reader hooks.

{Gaussian elimination is not a difficult algorithm, and there are many pieces of software that provide an implementation for floating point coefficients. Therefore, let us pause for a moment and consider a question: why would we need another piece of software to do this? From a point of view of a mathematics instructor, the answer is simple: The existing packages are not meant to provide an explanation of how they arrive at an answer. In contrast, our package draws upon the ideas and tools of Artificial Intelligence and in addition to giving an answer, it gives us a detailed account of the intermediate steps.

{With most other packages, all the work is done (very fast!) within the machine, and the only result is the final answer. Due to the floating point arithmetic, the answer is almost never 100% accurate, and the estimate of the size of the error is performed using condition numbers, which are also computed using numerical methods. We recall that the condition number of a square matrix  $A$  is defined as:

$$\{ \begin{aligned} & \\ & K(A) = \|A\| \cdot \|A^{-1}\|. \end{aligned}$$

{The property of the condition number is that if we solve the system  $Ax = b$ , but instead of  $b$  we only have  $b + \delta b$  where  $\delta b$  is the error of the right-hand side, then the solution is  $x + \delta x$  where:

$$\frac{\|\delta x\|}{\|x\|} \leq K(A) \cdot \frac{\|\delta b\|}{\|b\|}.$$

Thus,  $K(A)$  expresses the magnification of relative error when solving a linear system. Unfortunately, to know  $K(A)$  exactly we need to know the norm of  $A^{-1}$ . Moreover, the exact norm of the matrix with respect to the Euclidean norm can only be expressed in terms of the eigenvalues of  $\|A^T A\|$ . If the inverse is known, the norm can be estimated from above in a variety of ways with only minimal effort. Thus, a bit of a circular argument develops because to know the numerical error of  $A^{-1}$ . we need to know  $A^{-1}$ .

{As a result, one uses a proxy for  $K(A)$  which is obtained by approximately solving  $Ax = b$  for several randomly chosen vectors  $b$  and using these solutions together with (??) to deduce an upper bound on  $K(A)$ .

{Computer algebra systems solve linear system symbolically. However, they are also not geared towards explaining the steps of the solution. Often the symbolic computations are slow and blow up in size as without the help from the rounding, the coefficients grow fast in size. If you need to work out modest in size systems of linear equations, and need to understand how Gaussian elimination deals with the particular set of coefficients with respect to scaling and exchanging rows, this program may be of interest to you, as an alternative to working the example by hand.

{The program understands arbitrary precision integers and fractions. If all coefficients of the input are rational, so is the answer and all intermediate steps. Thus, the result has infinite precision. The program also understands complex coefficients. Again, if the real and imaginary parts of all coefficients are rational, all calculations are performed with 100% accuracy, and the result is given in terms of complex numbers with fractions as real and imaginary parts. The system supports single and double precision calculations, too. Thus, one can compare easily exact and approximate solutions to linear systems. Again, all intermediate steps are produced and can be inspected for unusual conditions which lead to the loss of precision.