,

Report for Gaussian elemination 2019-02-15

Buster Hultgren Wärn - dv17bhn@cs.umu.se

Linear Algebra 7.5p - 5MA160
Umeå University
Department of Mathematics and Mathematical Statistics

I come from a background in studying computing science, so writing algorithms and problem solving is not something which is new for me. The challenges I have had is mostly learning MATLAB, and how it differs from languages I'm more used to such as C and Java. That MATLAB starts indexing from 1 and not 0 is to say the least - mildly frustrating. I would probably prefer doing this assignment in either Java or Python.

One of the harder parts for me is keeping track of rows and columns in matrices. When seeing an actual matrix, it's pretty easy to pick out row and column vectors. However it becomes less intuitive when looping through a matrix with i and j. The solution in the end was to print out the matrices, to see which row / column I was manipulating. I tried using as few implemented matrix operations as possible, and implement my own instead.

Gauss-Jordan elimination is said to be more efficient, since it skips some computations. It would have been fun trying to do Gauss-Jordan elimination instead of regular Gauss-Elimination. When designing my algorithm, I did not take efficiency or smarter solutions into account. I only wanted a solution that works. If I rewrote my program, this is probably the first thing I would change.

One part of my solution is trying to create a diagonal (stretching the definition of diagonal matrices to include mXn) of leading 1's by swapping rows on the diagonal where there are 0's. This was a mistake in designing the algorithm. It does make the Gaussian elimination algorithm a little bit easier, but it comes with more flaws. For a starter, the solution is (at least I think) less efficient since more computations are needed. Then a lot of special cases pop up. For example, if the diagonal on the last row is 0, then my algorithm does not swap this one with any other. Trying to fix this will not be easy. Instead, a better solution would be getting all leading 1's for each row in a way that two leading 1's don't overlap, reduce the rows and the swap rows. This makes less sense when creating an algorithm to solve with pen and paper. But creating an algorithm in code, there are lots more steps which must be followed.

Did this assignment help me understand Gaussian elimination? Both yes and no. I would say that I already know Gaussian elimination pretty well and the hard part is, as we say in Swedish, to keep your tongue right in your mouth (to be care full $=$)). However, it have helped me in keeping track of rows and columns and how to conceptually think of them in another way. I guess for people that are used to math and not programming, they would find designing the algorithm the hardest part - whereas for me I did not.