FACULTY
OF INFORMATION
TECHNOLOGY
CTU IN PRAGUE

# Review report of a final thesis

| | |
|---|---|
| **Reviewer:** | Ing. Tomáš Oberhuber, Ph.D. |
| **Student:** | Emil Eyvazov |
| **Thesis title:** | Gauss-Jordan Solver of Linear Equation Systems on GPU |
| **Branch / specialization:** | Computer Science |
| **Created on:** | January 31, 2022 |

## Evaluation criteria

### 1. Fulfillment of the assignment

[1] assignment fulfilled
[2] assignment fulfilled with minor objections
▸ **[3] assignment fulfilled with major objections**
[4] assignment not fulfilled

Comparison of the performance of the CPU and the GPU solver (page 17) is rather suspicious. The author reports speedup 1785/30=59. This seems to be too much to me since the Gauss-Jordan algorithm is limited mainly by the memory bandwidth of both CPU and GPU. Since the memory bandwidth of GPU is approximately 20x higher compared to CPU, speedup 59x does not seem to be realistic. It indicates that the CPU code is not optimal. Moreover, the author never says what CPU was used for the testing - at least I did not find it in the text.
In addition, the GPU implementation presented by the author is not optimal in my opinion. To get more parallelism, one should eliminate the elements above the diagonal simultaneously with those below the diagonal, i.e. Algorithms 7 and 8. In this way, the matrix can be transformed into echelon form in one stage which exhibits more parallelism. The author also uses an auxiliary array referred to as multiplicatives. In my opinion, dividing the row with the pivot and avoiding the use of this array would be more efficient. The author also maps 1024 threads into a CUDA block which is usually not optimal. The author should present some tests with different numbers of CUDA threads in the CUDA block or at least discuss this choice.
Although, the author defines "average correctness of solver" which is average of the errors made by the solver on all tested matrices. This is a rather strange metric, one should look for the maximum error, analyze the matrix for which the error occurred and discuss it. If my understanding is correct, the author generates random matrices. The issue if this approach is that we do not know conditional number of such matrices which is proportional to an error of the Gauss-Jordan algorithm. When it comes to correctness, the author should have used some real matrices from matrix markets available on the internet.

## 2. Main written part 50/100 (E)

The written part is of rather low quality. Some sentences do not make sense like "The goal of this project is to implement System of Linear Equations ..." (page 1) or "As an additional solver, cuSOLVE solver will." (page 29) there sentences with repeating words like "As SLE are used widely used ..." (page 1) or the grammar is not correct "Initialize shared memory, copy to it values from multiplicatives..." (page 15) to name a few. The author also uses the word "computability" instead of "compute capability" (page 7).

The structure of the text is rather chaotic. In part 3.2, the author describes an implementation for CPU, but there is only one algorithm and the second one is in the appendix which does not make sense. The part 4.2.1 should be moved to Chapter 5. Parts 4.7 and 4.8 describing cuBLAS and cuSOLVE should be in Chapter 3.

On the page 3, the author defines the linear systems of matrices having m rows and n columns. The Gauss-Jordan algorithm is, however, implemented only of square matrices and this is not commented. Next, he defines the echelon form of a matrix. He says that "Any rows consisting entirely of zeros occurs at the bottom of the matrix." (page 3) which suggests singular matrices but the Gauss-Jordan algorithm is implemented only for regular matrices. Also he says "For each row that does not contain entirely zeros, the first non-zero entry is 1." However, the Algorithms 1, 4, 7 and 8 do not divide the row with the pivot by the pivot and so there will not be ones on the diagonal.

The architecture of the GPU is not described well. For example, streaming multiprocessors and their connection to CUDA blocks, is not mentioned at all. Coalesced memory accesses to the global memory are not explained either though they are crucial for the Gauss-Jordan algorithm.

Some algorithms, like Algorithm 7, should be presented as real code instead of pseudocode. I do not understand point 2 on page 15: "Id of the shared memory array that will be used for accessing shared memory array." or "... means copy of the lines 3-17 Algorithm 7." Both are not sentences at all and do not make any sense to me.

Images 4, 7, 8, 10 and 11 are blurred and they are hard to read.

Name of part 4.5 - "Full partial pivoting" sounds strange to me.

On page 32, the author says: "Reference solver on CPU graph is not included into the plot, as CPU is very slow compared to GPU for SLE solving, so reference solver would shift plot high enough and all GPU solver's graphs would collide with each other." One may use logarithmic scale in this case which would help to better presentation of the results in general.

## 3. Non-written part, attachments 50/100 (E)

In my opinion, the GPU algorithm presented by the author is not optimal. As I mentioned, the parts described in Algorithms 7 and 8, i.e. elimination of the elements above and below the diagonal, should be performed simultaneously. This would give more parallelism and it would avoid the necessity of the back substitution. The author does not present the CPU implementation which must be also suboptimal since the reported speedup, close to 60x, is rather suspicious - the algorithms are limited by the memory bandwidth mainly. The effect of the pivoting should be presented on real matrices for which the conditional number is known.

### 4. Evaluation of results, publication outputs and awards 50/100 (E)

The presented algorithms are significantly suboptimal as demonstrated on Figure 9 (page 36) for example. There is no reason for using them.

## The overall evaluation 50/100 (E)

The author presents suboptimal algorithms which are a significant issue in case of thesis dealing with programming of GPUs. The written part is not structured well, English grammar is not correct. Presentation of the results is not good.

## Questions for the defense

1. What CPU was used for the comparison.
2. Could the author comment about the achieved speedup? The speedup 60x seems to be too much since the Gauss-Jordan algorithm is limited by the memory bandwidth which is approximately 20x higher on GPU compared to CPU.
3. Does the presented implementation of the Gauss-Jordan algorithm really transform the matrix of the linear system into an echelon form with ones on the diagonal? I have not found division of the row with the pivot in any of the presented algorithms.
4. Why does the author map 1024 CUDA threads into a CUDA block? Did he try other values like 256?

# Instructions

## Fulfillment of the assignment

Assess whether the submitted FT defines the objectives sufficiently and in line with the assignment; whether the objectives are formulated correctly and fulfilled sufficiently. In the comment, specify the points of the assignment that have not been met, assess the severity, impact, and, if appropriate, also the cause of the deficiencies. If the assignment differs substantially from the standards for the FT or if the student has developed the FT beyond the assignment, describe the way it got reflected on the quality of the assignment's fulfilment and the way it affected your final evaluation.

## Main written part

Evaluate whether the extent of the FT is adequate to its content and scope: are all the parts of the FT contentful and necessary? Next, consider whether the submitted FT is actually correct – are there factual errors or inaccuracies?

Evaluate the logical structure of the FT, the thematic flow between chapters and whether the text is comprehensible to the reader. Assess whether the formal notations in the FT are used correctly. Assess the typographic and language aspects of the FT, follow the Dean's Directive No. 52/2021, Art. 3.

Evaluate whether the relevant sources are properly used, quoted and cited. Verify that all quotes are properly distinguished from the results achieved in the FT, thus, that the citation ethics has not been violated and that the citations are complete and in accordance with citation practices and standards. Finally, evaluate whether the software and other copyrighted works have been used in accordance with their license terms.

## Non-written part, attachments

Depending on the nature of the FT, comment on the non-written part of the thesis. For example: SW work – the overall quality of the program. Is the technology used (from the development to deployment) suitable and adequate? HW – functional sample. Evaluate the technology and tools used. Research and experimental work – repeatability of the experiment.

## Evaluation of results, publication outputs and awards

Depending on the nature of the thesis, estimate whether the thesis results could be deployed in practice; alternatively, evaluate whether the results of the FT extend the already published/known results or whether they bring in completely new findings.

## The overall evaluation

Summarize which of the aspects of the FT affected your grading process the most. The overall grade does not need to be an arithmetic mean (or other value) calculated from the evaluation in the previous criteria. Generally, a well-fulfilled assignment is assessed by grade A.