Efficient Modular Matrix Multiplication on GPU for Polynomial System solving

Dimitri Lesnoff

February 24, 2023

Modeling problems from biology, coding theory, combinatorics, aerospace engineering often rely on solving polynomial systems $f_1 = \cdots = f_m = 0$ in variables $x_1, \ldots, x_n$ exactly over a finite field or rational numbers.

To solve exactly the polynomial systems, we rely on Gröbner basis computations. This requires first computing a Gröbner basis for a total degree order using Buchberger or F_4/F_5 [Fau99; Fau02]. Then, we apply a change of order algorithm like FGLM [Fau+93] or Sparse-FGLM [FM11; FM17].

While the second step is only singly exponential in the number of variables, it is often the bottleneck of the computation in practice and consists in iterated matrix-vector and matrix thin-matrix products. Efficient implementations require computers with an architecture dedicated to parallelism. Most supercomputers in the Top500 use GPUs which outperform CPUs in terms of theoretical peak performance. GPUs permit heterogeneous computing. They have dedicated units for linear algebra tasks that are omnipresent in graphics applications. Their architecture well suited to parallelism seems to fit best our problematic.

Arithmetic libraries (FFLAS [gro19], FLINT [HJP13], NTL [Sho21]) on CPU use floating-point types to compute over a finite field. Even though some fields can only be represented on a single word with integers, there are efficient instructions for parallelism (AVX on CPU) and libraries (BLAS [ZQG11]) to optimize cache memory.

In this work we present two novel contributions. First, we have ported existing single-word algorithms with delayed modular reduction present in these libraries to GPU and obtained a competitive implementation. These algorithms are limited to 26 bits prime. Therefore, our second contribution is to develop a novel multiword algorithm that can handle characteristic with more than 26 bits by dividing each of our input matrices into multiple matrices with small and high parts of our coefficients. Despite the good performance we achieved, I will also mention some improvements possible with Karatsuba integer multiplication scheme. GPUs VRAM is much more limited than CPUs RAM. Thus we need to be cautious with the amount of memory used by these multiword algorithms. Yet, they offer a three-fold advantage since they make it possible to increase modular reduction delay, larger characteristics and the use of more efficient smaller precisions.

This is a joint work with Jérémy Berthomieu, Stef Graillat and Théo Mary.

References

[Fau+93] J.C. Faugère et al. “Efficient Computation of Zero-dimensional Gröbner Bases by Change of Ordering”. In: *Journal of Symbolic Computation* 16.4 (1993), pp. 329–344. ISSN: 0747-7171. DOI: https://doi.org/10.1006/jsco.1993.1051. URL: https://www.sciencedirect.com/science/article/pii/S0747717183710515.

[Fau02] Jean-Charles Faugère. “A new efficient algorithm for computing Gröbner bases without reduction to zero (F_5)”. In: *Proceedings of the 2002 international symposium on Symbolic and algebraic computation - ISSAC '02*. the 2002 international symposium. Lille, France: ACM Press, 2002, pp. 75–83. ISBN: 978-1-58113-484-1. DOI: 10.1145/780506.780516. URL: http://portal.acm.org/citation.cfm?doid=780506.780516 (visited on 02/13/2022).

[Fau99] Jean-Charles Faugère. “A new efficient algorithm for computing Gröbner bases (F_4)”. In: *Journal of Pure and Applied Algebra* (1999), p. 28.

[FM11] Jean-Charles Faugère and Chenqi Mou. “Fast Algorithm for Change of Ordering of Zero-Dimensional Gröbner Bases with Sparse Multiplication Matrices”. In: *Proceedings of the 36th International Symposium on Symbolic and Algebraic Computation*. ISSAC '11. San Jose, California, USA: Association for Computing Machinery, 2011, pp. 115–122. ISBN: 9781450306751. DOI: 10.1145/1993886.1993908. URL: https://doi.org/10.1145/1993886.1993908.

[FM17] Jean-Charles Faugère and Chenqi Mou. “Sparse FGLM algorithms”. In: *Journal of Symbolic Computation* 80 (2017), pp. 538–569. ISSN: 0747-7171. DOI: https://doi.org/10.1016/j.jsc.2016.07.025. URL: https://www.sciencedirect.com/science/article/pii/S0747717116300700.

[gro19] The FFLAS-FFPACK group. *FFLAS-FFPACK: Finite Field Linear Algebra Subroutines / Package*. v2.4.1. http://github.com/linbox-team/fflas-ffpack. 2019.

[HJP13] W. Hart, F. Johansson, and S. Pancratz. *FLINT: Fast Library for Number Theory*. Version 2.4.0, http://flintlib.org. 2013.

[Sho21] Victor Shoup. *NTL: a library for doing numbery theory*. 2021. URL: http://www.shoup.net.

[ZQG11] Wang Zhang Xianyi, Werner Saar Qian, and Kazushige Goto. *OpenBLAS*. 2011. URL: https://www.openblas.net/.