On the Application of Modular Arithmetic for Cumbersome Computational Tasks

Victor F. Edneral

We consider applications of the modular arithmetic to cumbersome computational tasks, i.e. to problems with a lot of operations with cumbersome numbers. Such problems often arise in computer algebra tasks. We mean evaluations of long polynomials with huge numerical coefficients. Traditionally a modular arithmetic is used for each separate arithmetic operation. But It is more effective to execute the programs from the beginning till the end modulo one prime. After several such calculations in modulo different primes we can finally restore the right values of all numbers of the result. With respect of the Chinese remainder theorem if you know remainders from division of a natural number by a number of noncomparable natural numbers you can restore the original number itself if it is not more than multiplication of all these divisors [1, 2]. We assume that all numbers in the problem are integer or rational. There is an generalization of the Chinese theorem for integer and rational numbers.

What advantages gives us modular approach on multiprocessor platforms? The main are:

- if we choose modulo numbers as primes a bit smaller than maximal integer for a used platform (2^{32} or 2^{64}), then in each process in modulo a prime we will have all integers of length shorter than this maximal integer. So if exclude subsidiary operations and restoring the final result integers from modular representation, we will have acceleration approximately as relation of operation of a long integers arithmetic in comparison with ordinary arithmetic. But how many processors do we need? An ordinary length of integer now is $\sim 2^{64} \sim 10^{20}$. It is possible to choose a number of primes a bit smaller than 10^{20} and to split our task on the corresponding number of processor. If we wish to treat integers about $2^{1024} \sim 10^{309}$ we need $log_{10}(2^{1024})/log_{10}(2^{64}) = 16$ processors and for 2^{2048} we need $2 * 16 = 32$ processor units. It is not too much;
- we suppose that each processor has its own RAM. But instead of integers of the original task in our case each processor will keep in RAM integers of an

ordinary length. So, we will have economization of RAM approximately 16 or 256 times in examples above. It can be critical important for large problems.

Takeing into account that all branches of such calculations do not demand any synchronization. It is case of so called *natural parallelization* [3, 4]. So, we can use a multicore computer or clouds or grid calculations.

Of course there are a lot of problems. Main problems are:

- if we need use rational numbers we should be sure that all denominators in the each calculation process are indivisible on the corresponding prime. But any chosen prime is a long integer and in some problems (for example at the normal form calculation [5]) denominators can be very long but consist of short prime factors only. Any case it is simpler to use this approach for integer problems;
- It can be problems with algorithms which have internal branching. But for some methods for example for a Gr´obner basis calculation there are conditions of applicability of the modular approach [6] . So, complicated algorithms should be investigated additionally.

Of course the written above should be discussed.

References

[1] https://habrahabr.ru/post/144886/

[2] Amos Omondi, Benjamin Premkumar, *Residue Number Systems: Theory and Implementation*, 2007.

[3] S.N. Andrianov, A.B. Degtyarev, *Parallel and distributed computing.* Sankt-Petersburg, Solo, pp. 59 (2007), ISBN: 978-5-98340-073-3. In Russian.

[4] Strzodka R. *The Natural Parallelism.* In collection *Facing the Multicore-Challenge*, LNCS 6310, chapter 3 (2010). http://dx.doi.org/10.1007/978-3-642-16233-6_3.

[5] V.F. Edneral, *On algorithm of the normal form building.* Proceed. of the 10th International Workshop on Computer Algebra in Scientific Computing (CASC 2007, September 16-20, 2007. Bonn, Germany), ed. by Ganzha et al., Springer-Verlag, LNCS 4770, (2007) 134142.

[6] Elizabeth A. Arnold, *Modular algorithms for computing Gr´obner bases.* Journal of Symbolic Computation **35** (2003) 403419

Victor F. Edneral
Skobeltsyn Institute of Nuclear Physics
Lomonosov Moscow State University
Leninskie Gory 1(2), Moscow. 119991, Russia
e-mail: `edneral@theory.sinp.msu.ru`