Proceedings of Algoritmy
2000, pp. 1–10

# PARALLEL ALGORITHMS FOR PRACTICAL MULTIPRECISION ARITHMETIC USING THE KARATSUBA METHOD*

TUDOR JEBELEAN†, MIRCEA DRAGAN‡, DORIN TEPENEU§, AND VIOREL NEGRU¶

**Abstract.** We present a parallel implementation of the Karatsuba method for multiplication of multiprecision integers and its usage for integer division with remainder and for bidirectional exact division. Both the sequential and the parallel version exhibit significant speed-up over the currently used methods for exact arithmetic in scientific computing. The practical experiments use the SGI Origin shared memory architecture with 20 processors, and show an acceptable efficiency (70% to 80%).

**Key words.** multiprecision arithmetic, Karatsuba multiplication, exact division

**AMS subject classifications.** 11A05, 68Q22

**Extended Abstract for Poster Presentation.** The need for arbitrary precision arithmetic arises in various applications of scientific computing: solving unstable dynamical systems, ill-behaved differential equations, nonlinear parametric systems, etc. The basis of such computation is constituted by multiple precision rational and integer arithmetic routines. We present in this poster a parallel implementation of the most important arithmetic algorithms, which is being successfully integrated in a flexible system for scientific computation [6] currently under development at the University of Timisoara (Romania) in cooperation with RISC-Linz (Austria).

One of the basic ingredients of these routines is the Karatsuba method for long integer multiplication [3], which is probably the only asymptotically fast algorithm of practical use for integer arithmetic. Depending on the implementation, the break-even point against the classical algorithm is typically between 5 and 50 words. Furthermore, this method admits parallelization since it is based on the *divide and conquer* approach. We present our parallel implementation which exhibits a speed-up between 6 and 15 on 9 to 20 processor and an efficiency between 70% and 80% for operands between 50 and 500 computer words.

Until recently, integer division did not benefit from the Karatsuba method. Indeed, although theoretically division has the same time complexity as multiplication (see e.g. [4], p. 275), a division algorithm designed along the lines as explained by Knuth will be about 30 times slower than multiplication (see analysis in [5]). In [2] we introduce a technique which combines Krandick's [5] division algorithm for computing the high-order part of the quotient with Karatsuba multiplication, leading to a division algorithm which is about two times slower than Karatsuba multiplication. The main idea of the method is to delay the update of part of the dividend until this can be done by multiplication of large balanced operands. Based on this technique we use the parallel Karatsuba implementation for long integer division, obtaining a

*Partially supported by project "Intelligent Environment for Scientific Computing" – grant nr. 12623/98 from World Bank and the Romanian Government.

†Research Institute for Symbolic Computation, A-4232 Hagenberg, Austria (tudor@risc.uni-linz.ac.at).
‡West University of Timisoara, RO-1900 Timisoara, Romania.
§West University of Timisoara, RO-1900 Timisoara, Romania.
¶West University of Timisoara, RO-1900 Timisoara, Romania.

similar speed-up and a significant improvement over the classical division method (10 times on 9 processors).

A routine which is much used in exact arithmetic is the exact division algorithm [1]. (Exact division is integer division in which it is known in advance that the remainder is null.) This algorithm also benefits from the parallel Karatsuba technique and in fact another factor of parallelism is added by combining left-to-right with right-to-left division as in [5] by bidirectional computation. We experiment with various parallel versions of this algorithm: First, parallelization on 2 processors is done by performing the two directions of the division in parallel. Second, more parallel processors are used by performing the 3 branches of the Karatsuba multiplication in parallel (recursively this allows to use 9, 27, 81 processes).

We perform practical experiments using the multiple precision package GNU-MP on the architecture SGI Origin 2000 equipped with 64 R12000 300 MHZ processors. The parallelization is using the POSIX Threads library from SGI based on the shared memory pardigm. The length of the multiple precision integers ranges between 50 and 500 words (of 32 bits).

The experiments demonstrate the effectiveness of this approach for speeding up exact division (also integer division and multiplication) on parallel architectures (even with a low number of processors), and therefore opens the possibility of tackling large problems in primality testing, cryptography, and other applications of number theory.

## REFERENCES

[1] T. Jebelean, An algorithm for exact division, Journal of Symbolic Computation, 15 (1993), pp. 169–180.

[2] T. Jebelean, *Practical integer division with Karatsuba complexity*, in ISSAC'97 (International Symposium on Symbolic and Algebraic Computation), W. Kuechlin, ed., ACM Press, 1997, pp. 339–341.

[3] A. Karatsuba and Yu Ofman, *Multiplication of multidigit numbers on automata*, Sov. Phys. Dokl., 7 (1962), pp. 595–596.

[4] D. E. Knuth, *The art of computer programming, vol. 2*, Addison-Wesley, 1981.

[5] W. Krandick and T. Jebelean, *Bidirectional exact integer division*, Journal of Symbolic Computation, 21 (1996), pp. 441–455.

[6] V. Negru and C. Sandru and M. Rotaru, *Multi-agent system for scientific problem solving*, CEEMAS'99 (Proc. of International Workshop of Central and Eastern Europe of Multi-agent Systems, St. Petersburg), pp. 173–180.