Exact Division with Karatsuba Complexity
EXTENDED ABSTRACT *

Tudor Jebelean

Research Institute for Symbolic Computation
A4232 Schloß Hagenberg, Austria

tudor@risc.uni-linz.ac.at

Abstract

When it is known in advance that the remainder is null, division of integers is called *exact* and can be performed 4 times faster than classical integer division by the bidirectional algorithm of Krandick and Jebelean. We combine this algorithm with Karatsuba multiplication by delaying parts of the dividend updates until they can be performed by multiplication between large balanced operands. The new algorithm has (in the average) Karatsuba complexity $O(n^{\log 3/\log 2})$ in the length of the quotient. An implementation using the computer algebra system `saclib` reveals a visible speed-up over the bidirectional algorithm at 40 words and a speed-up of 2 at about 200 words. Up to 1000 words the exact division is about 20% faster than Karatsuba multiplication.

The Karatsuba method for long integer multiplication [4] 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.

However, integer division with remainder does not benefit from this algorithm. Indeed, although theoretically division has the same time complexity as multiplication (see e.g. [5], p. 275), a division algorithm designed along the lines as explained by Knuth will be about 30 times slower than multiplication (see analysis in [6]), which gives a break-even point at about 15,000 words. By making use of the Krandick-Johnson multiplication [8, 7] which computes only the high-order digits, and of a squaring routine which is twice as fast as general multiplication, one may hope to reduce the gap to 15 times, which will still give a break-even point of several thousands words.

In [3] we present a technique which combines Krandick's division algorithm for computing the high-order part of the quotient [6] 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. The update can be delayed because Krandick's algorithm needs only 3 word-updates below the current position in the dividend.

Together with Krandick, in [6] we develop a technique to compute the quotient of an *exact division* starting from both ends of the operands, which saves (roughly) half of the work.

The present work improves this technique in the following way:

*Supported by Austrian Forschungsförderungsfonds (FWF), project P10002-TEC.

- The algorithm from [3] is used for finding the most significant half of the quotient in a *left-to-right* manner.
- The least significant part of the quotient is computed in a *right-to-left* manner using a new version of the exact division method introduced in [2]. This new version of the method delays some of the digit multiplications until the Karatsuba algorithm can be used for performing them on larger parts of the operands. The technique used for this delaying is similar to the one presented in [3], with some simplification because the problem of the carry propagation easier to handle in exact division. Namely, the extra computation on three additional digits and also the condition to verify correctness at the end are not needed.

It is straightforward to show that the new algorithm has (in the average) Karatsuba complexity $O(n^{\log 3/\log 2})$ in the length of the quotient.

We performed practical experiments using the computer algebra system `saclib` [1] and the implementation of Krandick of The experiments reveal a visible speed-up over the bidirectional algorithm at 40 words and a speed-up of 2 at about 200 words. Up to 1000 words the exact division is about 20% faster than Karatsuba multiplication.

Acknowledgements I express special thanks to Werner Krandick for the excellent work in designing, implementing, and describing the high-order division, which constitutes an essential ingredient of this method.

References

[1] G. E. Collins, B. Buchberger, M. J. Encarnacion, H. Hong, J. R. Johnson, W. Krandick, R. Loos, A. M. Mandache, A. Neubacher, and H. Vielhaber. SACLIB 1.1 User's Guide. Technical Report 93–19, RISC–Linz, 1993.

[2] T. Jebelean. An algorithm for exact division. *Journal of Symbolic Computation*, 15(2):169–180, February 1993.

[3] T. Jebelean. Practical integer division with Karatsuba complexity. In W. Kuechlin, editor, *ISSAC'97 (International Symposium on Symbolic and Algebraic Computation, Maui, Hawaii, July 21-23, 1997)*, pages 339–341. ACM Press, 1997.

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

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

[6] W. Krandick and T. Jebelean. Bidirectional exact integer division. *Journal of Symbolic Computation*, 21:441–455, 1996.

[7] W. Krandick and J. R. Johnson. Efficient multiprecision floating point multiplication with exact rounding. Technical Report 93-76, RISC-Linz, RISC-Linz, Johannes Kepler University, A-4040 Linz, Austria, 1993. presented at the Rhine Workshop on Computer Algebra, Karlsruhe, Germany, 1994.

[8] W. Krandick and J. R. Johnson. Efficient multiprecision floating point multiplication with optimal directional rounding. In Earl Swartzlander, Jr., Mary Jane Irwin, and Graham Jullien, editors, *Proceedings of the 11th IEEE Symposium on Computer Arithmetic*, pages 228–233, P.O.Box 3041, Los Alamitos, CA 90720-126, Phone: 714 821-838, 1993. IEEE, IEEE Computer Society Press.