Towards Optimal Use of Multi-Precision Arithmetic: A Remark

Vladik Kreinovich[1] and Siegfried Rump[2]

[1]Department of Computer Science, University of Texas at El Paso
El Paso, TX 79968, USA, vladik@utep.edu

[2]Institute for Reliable Computing, Hamburg University of Technology
Schwarzenbergstr. 95, D-21071 Hamburg, Germany,
and Waseda University, Faculty of Science and Engineering,
2-4-12 Okubo, Shinjuku-ku, Tokyo 169-0072, Japan,
e-mail rump@tu-harburg.de

Abstract

If standard-precision computations do not lead to the desired accuracy, then it is reasonable to increase precision until we reach this accuracy. What is the optimal way of increasing precision? One possibility is to choose a constant $q > 1$, so that if the precision which requires the time t did not lead to a success, we select the next precision that requires time $q \cdot t$. It was shown that among such strategies, the optimal (worst-case) overhead is attained when $q = 2$. In this paper, we show that this "time-doubling" strategy is optimal among all possible strategies, not only among the ones in which we always increase time by a constant $q > 1$.

Formulation of the problem. In multi-precision arithmetic, it is possible to pick a precision and make all computations with this precision; see, e.g., [1, 2]. If we use validated computations, then after the corresponding computations, we learn the accuracy of the results.

Usually, we want to compute the result of an algorithm with a given accuracy. We can start with a certain precision. If this precision leads to the desired results accuracy, we are done; if not, we repeat the computations with the increased precision, etc.

The question is: what is the best approach to increasing precision?

A natural approach to solving this problem. We usually have some idea of how the computation time t depends on the precision: e.g., for addition, the computation time grows as the number d of digits; for the standard multiplication, the time grows as d^2, etc.

In view of this known dependence, we can easily transform the precision (number of digits) into time and vice versa. Therefore, the problem of selecting the precision d can be reformulated as the problem of selecting the corresponding computation time t.

In other words, what we need to describe is as follows: if computations with time t are not sufficient, then we must select computations with larger precision which require a longer computation time $t'(t) > t$. The question is: given t, what $t'(t) > t$ should we choose.

Continuous approximation. In reality, we can only choose between integer number of digits 1, 2, 3, ... However, the need for high precision arises only when normal accuracy, with 32, 64, or 128 bits, is not sufficient. For the resulting huge number of bits, the difference between, say, 128 and 129 bit precision is not so large, so we can safely ignore the discreteness and assume that d (and hence t) can take all real values.

It is also reasonable to assume that close values of t should lead to close values of t', i.e., that the function $t \to t'$ is continuous.

Towards formulating the problem in precise terms. Let us start with a precision corresponding to time t_0. If this precision is not sufficient, we select a precision corresponding to time $t_1 \stackrel{\text{def}}{=} t'(t_0) > t_0$. If the precision corresponding to time t_1 is not sufficient, we select a precision corresponding to time $t_2 \stackrel{\text{def}}{=} t'(t_1) > t_1$, etc.

Similarly, since the function $t'(t)$ is continuous, there should exist a value $t_{-1} < t_0$ for which $t'(t_{-1}) = t_0$; similarly, there should exist a value $t_{-2} < t_{-1}$ for which $t'(t_{-2}) = t_{-1}$, etc.

As a result, a get a doubly infinite sequence of positive values

$$\ldots < t_{-n} < \ldots < t_{-2} < t_{-1} < t_0 < t_1 < t_2 < \ldots < t_n < \ldots \tag{1}$$

The meaning of this sequence is as follows: if we tried precision corresponding to some starting value t_s and did not succeed, then we try values t_{s+1}, t_{s+2}, ..., until we succeed.

How can we compare which sequence is the best? If we start with a value t_s and the first successful value, after trying $t_s, t_{s+1}, \ldots, t_{v-1}$, is t_v, this means that the smallest precision needed for our accuracy corresponds to the time t which is somewhere between t_{v-1} and t_v. In the ideal world, we could just spend the time t and get the desired accuracy. Instead, we spend the time $t_s + t_{s+1} + \ldots + t_{v-1} + t_v$. We will say that there is an overhead C if the actually spent time never exceeds $C \cdot t$. Thus, we arrive at the following definition.

Definition. *Let $C > 0$ be a real number. We say that a sequence (1) has an* overhead C *if for every two integers $s < v$ and for every real number t from the interval $(t_{v-1}, t_v]$, we have*

$$t_s + t_{s+1} + \ldots + t_{v-1} + t_v \le C \cdot t. \tag{2}$$

What was known. It is known [3] that for the sequence $t_i = 2^i \cdot t_0$, we have $C = 4$, and that this is the smallest overhead that can be achieved for sequences of the type $t_i = q^i \cdot t_0$.

To prove this result, let us simplify the condition (2). Since the inequality (2) is true for all $t \in (t_{v-1}, t_v]$, by tending to the limit $t \to t_{v-1}$, we conclude that

$$t_s + t_{s+1} + \ldots + t_{v-1} + t_v \le C \cdot t_{v-1}. \tag{3}$$

Vice versa, if (3) is true, then, since $t_{v-1} < t$, (2) is also true. Thus, the inequality (2) is equivalent to inequality (3).

Similarly, since (3) is true for every s, by tending to the limit $s \to -\infty$, we conclude that

$$\ldots + t_s + t_{s+1} + \ldots + t_{v-1} + t_v \le C \cdot t_{v-1}. \tag{4}$$

Vice versa, if (4) is true, then, by deleting some positive terms $t_{s-1}, \ldots$, we get a smaller left-hand side and thus, the inequality (3). Thus, the original inequality (2) holds if and only if the inequality (4) holds for every v.

For $t_i = t_0 \cdot q^i$, by dividing both sides of the inequality (4) by $t_0 \cdot q^{v-1}$, we conclude that

$$q + 1 + \frac{1}{q} + \frac{1}{q^2} + \ldots \le C,$$

i.e.,

$$\frac{q}{1 - \frac{1}{q}} \le C,$$

or, simplifying,

$$\frac{q^2}{q-1} \le C.$$

Thus, the smallest value of C can be attained when the ratio $\dfrac{q^2}{q-1}$ is the smallest among all the values $q > 1$. Differentiating this expression with respect to q and equating the derivative to 0, we conclude that $q = 2$. For $q = 2$, the ratio $\dfrac{q^2}{q-1}$ is equal to 4, so we have an overhead $C = 4$.

This result is used in practice. The above result is actually used to optimize computations with multiple precision; see, e.g., [5] and references therein.

In this paper, we prove that the overhead is the smallest possible not only among sequences of the type $t_0 \cdot q^i$, but also among all possible sequences.

Proposition. *For every sequence, the overhead is greater than or equal to 4.*

Comment. Thus, the "time-doubling" scheme is indeed optimal.

Proof. We will prove this result by reduction to a contradiction. Let us assume that there is a sequence $\{t_i\}$ with an overhead $C < 4$. From (4), we can now conclude that $t_{v-1} + t_v \le C \cdot t_{v-1}$, i.e., that $t_v \le \delta_1 \cdot t_{v-1}$, where $\delta_1 = C - 1$.

Since this is true for every v, we conclude that $t_{v-1} \le \delta_1 \cdot t_{v-2}$, hence $t_{v-2} \ge \dfrac{1}{\delta_1} \cdot t_{v-1}$. Similarly, $t_{v-3} \ge \dfrac{1}{\delta_1} \cdot t_{v-2}$ hence $t_{v-3} \ge \dfrac{1}{\delta_1^2} \cdot t_{v-1}$ etc. Thus,

$$\ldots + t_s + \ldots + t_{v-2} + t_{v-1} \ge t_{v-1} \cdot \left(1 + \frac{1}{\delta_1} + \frac{1}{\delta_1^2} + \ldots\right) = t_{v-1} \cdot \frac{1}{1 - \dfrac{1}{\delta_1}} = t_{v-1} \cdot \frac{\delta_1}{\delta_1 - 1}. \qquad (5)$$

Subtracting (5) from (4), we conclude that $t_v \le \delta_2 \cdot t_{v-1}$, where $\delta_2 = C - \dfrac{\delta_1}{\delta_1 - 1}$.

Let us show that $\delta_2 < \delta_1$. Indeed, the difference $\delta_1 - \delta_2$ is equal to

$$\delta_1 + \frac{\delta_1}{\delta_1 - 1} - C = \frac{\delta_1^2}{\delta_1 - 1} - C.$$

We have already mentioned that the smallest possible value of $\dfrac{\delta_1^2}{\delta_1 - 1}$ is equal to $4 > C$, so we indeed get $\delta_2 < \delta_1$.

Similarly, from the fact that $t_v \le \delta_2 \cdot t_{v-1}$ for all v, we conclude that $t_v \le \delta_3 \cdot t_{v-1}$, where $\delta_3 = C - \dfrac{\delta_2}{\delta_2 - 1} < \delta_2$, etc.

Since $t_v > t_{v-1}$, we have $\delta_j > 1$ for all j. The sequence $\delta_1 > \delta_2 > \delta_3 > \ldots$ is a decreasing sequence which is bounded by 1 from below; thus, this sequence has a limit $\delta \ge 1$. From the definition $\delta_{j+1} = C - \dfrac{\delta_j}{\delta_j - 1}$, in the limit $j \to \infty$, we conclude that $\delta = C - \dfrac{\delta}{\delta - 1}$, i.e., that $C = \delta + \dfrac{\delta}{\delta - 1} = \dfrac{\delta^2}{\delta - 1}$. However, we know that the smallest possible value of $\dfrac{\delta^2}{\delta - 1}$ is 4, so $\dfrac{\delta^2}{\delta - 1}$ cannot be equal to $C < 4$. The contradiction proves that a sequence cannot have an overhead $C < 4$. The proposition is proven.

Comment. This proof is mathematically similar to the proof of a different optimality result presented in [4].

Acknowledgments. This work was supported in part by NASA under cooperative agreement NCC5-209, by NSF grants EAR-0225670 and DMS0-0532645, Army Research Lab grant DATM-05-02-C-0046, Star Award from the University of Texas System, and by the Texas Department of Transportation grant No. 0-5453. The authors are greatly thankful to the organizers of the Dagstuhl Seminar on Reliable Implementation of Real Number Algorithms: Theory and Practice, Dagstuhl, Germany, January 8–13, 2006, for the great collaboration opportunity.

The authors are also thankful to the anonymous referees for helpful suggestions.

References

[1] L. Fousse, G. Hanrot, V. Lefévre, P. Pélissier, and P. Zimmermann, *MPFR: A Multiple-Precision Binary Floating-Point Library With Correct Rounding*, l'Institut National de Recherche en Informatique et en Automatique INRIA, Technical Report RR-5753, November 2005, http://hal.inria.fr/inria-00000818

[2] N. Revol, *MPFI, a multiple precision interval arithmetic library*, http://perso.ens-lyon.fr/nathalie.revol/mpfi.html, 2001–04.

[3] S. M. Rump, *Kleine Fehlerschranken bei Matrixproblemen*, PhD thesis, Universität Karlsruhe, 1980.

[4] R. A. Trejo, J. Galloway, C. Sachar, V. Kreinovich, C. Baral, and L. C. Tuan, "From Planning to Searching for the Shortest Plan: An Optimal Transition", *International Journal of Uncertainty, Fuzziness, Knowledge-Based Systems (IJUFKS)*, 2001, Vol. 9, No. 6, pp. 827–838.

[5] J. van der Hoeven, "Computations with effective real numbers", *Theoretical Computer Science* (to appear).