Solving Triangular Systems More Accurately and Efficiently

Ph. Langlois, N. Louvet

DALI-LP2A Laboratory. Université de Perpignan.
philippe.langlois,nicolas.louvet@univ-perp.fr

Abstract: The aim of the proposed talk is to present a new algorithm that solves linear triangular systems accurately and efficiently. By accurately, we mean that this algorithm should yield a solution as accurate as if it is computed in twice the working precision. By efficiently, we mean that its implementation should run faster that the existing algorithms with the same output accuracy.

Keywords: IEEE-754 floating point arithmetic, error-free transformations, extended precision, XBLAS, triangular linear system, substitution algorithm.

1 How to improve the result accuracy ?

When we perform computations with finite-precision arithmetic, *i.e*, with floating point arithmetic, the computed values of the intermediate variables often suffer from the rounding errors introduced by each arithmetic operator $+, -, \times, /, \sqrt{}$. These rounding errors contribute to the inaccuracy of the results computed with numerical algorithms.

With multiprecision libraries: To improve this accuracy, we sometines need to increase the working precision which is often the IEEE-754 double precision format. Fixed-length multiprecision librairies provide both efficiency and precision to be interesting in scientific computing. Bailey's double-double algorithms [6] are used by the authors of [4] to implement efficiently the XBLAS library. These extended basic linear algebra subroutines provide the same set of routines as the well known BLAS but allow intermediate computation in extended precision. Double-double numbers implement this extended precision that improves the accuracy and the convergence of some BLAS and LAPACK subroutines [1].

With targeted algorithms: Improving the computed result accuracy can be designed for a given algorithm. A classic example of such targeted accuracy improvement is the summation of n floating point numbers : Knuth-Kahan compensated summation, Kahan-Priest double compensated summation... (see [2, chap.4] for entries). These highly accurate algorithms compute correcting terms which take into account the rounding errors accumulated during the calculus. Since the propagation of these elementary rounding errors is tedious to describe for large numerical algorithms, the CENA method provides an automatic linear correction of the global rounding error together with a bound of the corrected accuracy [3]. This correction relies on well known results about the elementary rounding errors in the arithmetic operators named *error free transformations* by Ogita *et al.* in [5].

Experimental results exhibit that the accuracy of a corrected result with the CENA method is of the order of the condition number times the square of the working precision. Very recent results from Ogita *et al.* propose the first proof of this kind of "twice the working precision" behavior for the summation and the dot product algorithms [5].

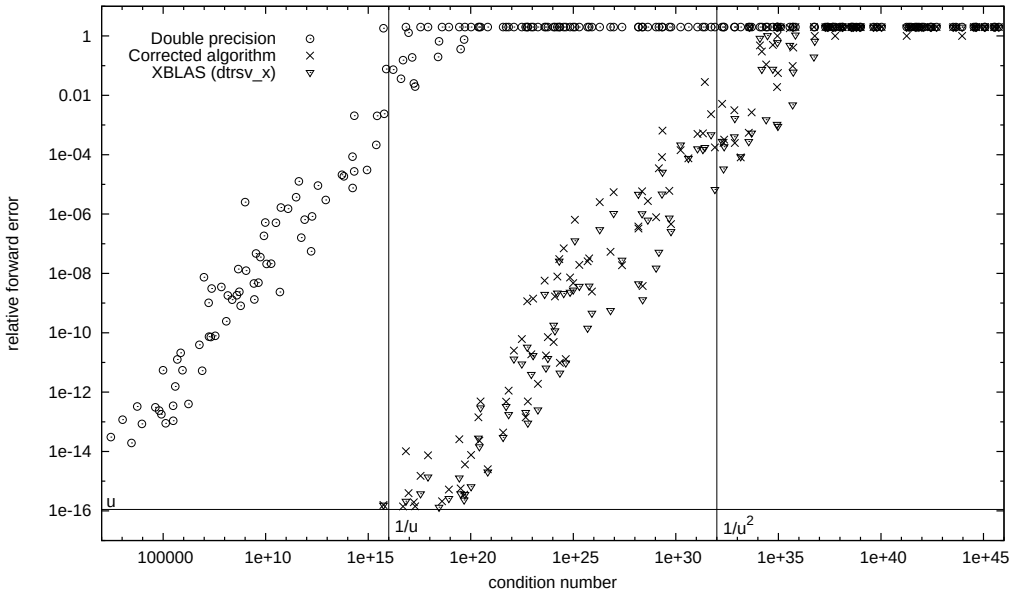
2 An efficient improvement of the accuracy for triangular systems

The algorithm we propose here is an optimised instantiation of the CENA correction applied to the substitution algorithm for triangular systems.

In this talk, we first explicit the correcting term the CENA method computes dynamically. This term represents the global forward error generated by the substitution algorithm when solving a triangular system $Tx = b$. Then we introduce a fast version of this corrected substitution algorithm that computes the expected accurate solution in $0(n^2)$ floating point operations, *i.e.*, in the same theoretical complexity as the not corrected substitution algorithm.

Modern floating point units and associated compilers are such that the actual overhead of the error free transformation algorithms is measured to be about 2.5 times faster than its theoretical floating point complexity [5]. We compare the actual computing times of the proposed corrected algorithm with the reference implementation of the XBLAS dtrsv_x routine. We experiment that *the proposed algorithm is twice faster than the reference time given by the XBLAS current implementation*.

We also experiment both the classic substitution algorithm performed in double precision, the XBLAS dtrsv_x and our corrected routine for a wide range of carefully generated very ill-conditioned systems: the Skeel condition numbers vary from 10^3 to 10^{45} (these huge condition numbers have a sense since here both T and b have been designed to be exact floating point numbers). Next figure presents the relative accuracy $|\widehat{x} - x_d|/|x_d|$ of the computed solution $\widehat{x}$ compared to the condition number range. We observe that both the XBLAS and our corrected substitution algorithms exhibit the expected behavior: *the relative accuracy is proportional to the square of the double precision* $\mathbf{u}$. The full precision solution is computed as long as the condition number is smaller than $1/\mathbf{u}$. Then the computed solution has an accuracy of the order cond $\times \mathbf{u}^2$ for systems with a condition number (cond) smaller than $1/\mathbf{u}^2$. At last, no computed digit remains exact for condition number up to $1/\mathbf{u}^2$.



References

[1] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen. *LAPACK Users' Guide*. Society for Industrial and Applied Mathematics, Philadelphia, PA, third edition, 1999.

[2] Nicholas J. Higham. *Accuracy and Stability of Numerical Algorithms*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, second edition, 2002.

[3] Philippe Langlois. Automatic linear correction of rounding errors. *BIT*, 41(3):515–539, 2001.

[4] Xiaoye S. Li, James W. Demmel, David H. Bailey, Greg Henry, Yozo Hida, Jummy Iskandar, William Kahan, Suh Y. Kang, Anil Kapur, Michael C. Martin, Brandon J. Thompson, Teresa Tung, and Daniel J. Yoo. Design, implementation and testing of extended and mixed precision BLAS. *ACM Transactions on Mathematical Software*, 28(2):152–205, June 2002.

[5] Takeshi Ogita, Siegfried M. Rump, and Shin'ichi Oishi. Accurate sum and dot product. *SIAM J. Sci. Comput.*, 2005. (to appear).

[6] Xiaoye S. Li Yozo Hida and David H. Bailey. Algorithms for quad-double precision floating point arithmetic. *15th IEEE Symposium on Computer Arithmetic*, pages 155–162, 2001.