A method of calculating faithful rounding of l_2-norm for n-vectors

Stef Graillat[1], Christoph Lauter[1], Ping Tak Peter Tang[2], Naoya Yamanka[3] and Shin'ichi Oishi[3]

[1] Sorbonne Universités
UPMC Univ Paris 06
UMR 7606, LIP6
4, place Jussieu
F - 75005 Paris
France

[2] Intel Corporation
2200 Mission College Blvd
Santa Clara, CA 95054
USA

[3] Faculty of Science
and Engineering
Waseda University
3-4-1 Okubo
Tokyo 169-8555
Japan

stef.graillat@lip6.fr, christoph.lauter@lip6.fr, peter.tang@intel.com, yamanaka@suou.waseda.jp, oishi@waseda.jp

Keywords: Floating-point arithmetic, error-free transformations, faithful rounding, 2-norm, underflow, overflow

In this paper, we present an efficient algorithm to compute the faithful rounding of the l_2-norm, $\sqrt{\sum_j^n x_j^2}$, of a floating-point vector $[x_1, x_2, \ldots, x_n]^T$. This means that the result is accurate to within one bit of the underlying floating-point type. The algorithm is also faithful in exception generations: an overflow or underflow exception is generated if and only if the input data calls for this event. This new algorithm is also well suited for parallel and vectorized implementations. In contrast to other algorithms, the expensive floating-point division operation is not used. We demonstrate our algorithm with an implementation that runs about 4.5 times faster then the netlib version [1].

There are three novel aspects to our algorithm for l_2-norms:

First, for an arbitrary real value σ, we establish an accuracy condition for a floating-point approximation S to σ that guarantees the correct rounding of the square root $\circ(\sqrt{S})$ to be a faithful rounding of $\sqrt{\sigma}$.

Second, we propose a way of computing an approximation S to the sum $\sigma = \sum_j x_j^2$ that satisfies the accuracy condition. This summation

algorithm makes use of error-free transformations [4] at crucial steps. Our error-free transformation is custom designed for l_2-norm computation and thus requires fewer renormalization steps than a more general error-free transformation needs. We show that the approximation S is accurate up to a relative error bound of $\Delta_\ell(3\varepsilon^2)$, where ε is the machine epsilon and $\Delta_\ell(\delta) = \ell\delta/(1-\ell\delta)$ bounds the accumulated error over ℓ summation steps [3] for an underlying addition operation with a relative error bound of δ. Our derivation of $\delta = 3\varepsilon^2$ is an enhancement; the standard bounds on δ in the literature are strictly greater than $3\varepsilon^2$.

Third, in order to avoid spurious overflow and underflow in the intermediate computations, our algorithm extends the previous work by Blue [2]: the input data x_j are appropriately scaled into "bins" such that computing and accumulating their squares x_j^2 are guaranteed exception free. While Blue uses three bins and the division operation, our algorithm uses only two and is division free. These properties economize registers usage and improve performance. The claim of faithful rounding and exception generation is supported by mathematical proofs. The proof of faithful overflow generation is relatively straightforward, but that for faithful underflow generation requires considerably greater care.

References:

[**1**] Anderson, Bai, Bischof, Blackford, Demmel, Dongarra, Croz, Hammarling, Greenbaum, McKenney, and Sorensen, *LAPACK Users' guide (third ed.)*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1999.

[**2**] Blue, A portable Fortran program to find the Euclidean norm of a vector, *ACM Trans. Math. Softw.*, 4 (1978), No. 1, pp. 15–23.

[**3**] Higham, *Accuracy and stability of numerical algorithms (second ed.)*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2002.

[**4**] Ogita, Rump, and Oishi, Accurate Sum And Dot Product, *SIAM J. Sci. Comput.*, 26 (2005), No. 6, pp. 1955–1988.