Compensated Horner scheme in k-fold working precision

Ph. Langlois, **N. Louvet**

DALI-LP2A Laboratory,
University of Perpignan,
F-66860 Perpignan, France
[philippe.langlois, nicolas.louvet]@univ-perp.fr

The backward stability of the Horner scheme when evaluating a given polynomial p at the point x justifies its practical interest. Nevertheless, the computed result can be arbitrarily less accurate than the working precision $\mathbf{u}$ when the evaluation of $p(x)$ is ill-conditioned. This is the case for example in the neighborhood of multiple roots where all the digits or even the order of the computed value of $p(x)$ can be false. Several techniques and softwares intend to improve the accuracy of results computed in floating point arithmetic. When an IEEE-754 floating point arithmetic is available, "double-double" and "quad-double" software libraries are effective solutions to simulate respectively twice our four times the working precision [1].

In [2], we have already described a compensated Horner scheme: with this scheme, the result is of the same quality as if computed in doubled working precision. We present here another compensated algorithm that computes an approximate r of $p(x)$ of the same quality as if computed in k-fold working precision ($k \geq 2$). More exactly, this means that r satisfies

$$\frac{|r - p(x)|}{|p(x)|} \leq \mathbf{u} + (\alpha\,\mathbf{u})^k\,\mathrm{cond}(p, x), \tag{1}$$

where α is a moderate constant and $\mathrm{cond}(p, x)$ is the classical condition number that describes the sensitivity of the polynomial evaluation. This algorithm only requires an IEEE-754 like floating point arithmetic with rounding to the nearest. Our main tool to improve the accuracy of the computed result is an "error-free transformation" [3] (EFT) for the polynomial evaluation with the Horner scheme. We prove that recursive application of this EFT allows us to compute an approximate that satisfies (1).

Experimental results show that the time penalty due to the improvement of the accuracy is very reasonable for $k \leq 4$. In particular, our routine runs about 40% faster than the corresponding routine based on the quad-double library. This justifies the practical interest of the method when only a small increase of the working precision is needed.

References

[1] Y. Hida, X. S. Li, and D. H. Bailey, Algorithms for quad-double precision floating point arithmetic. In *Proc. 15th IEEE Symposium on Computer Arithmetic*, 155-162. IEEE Computer Society Press, Los Alamitos, CA, USA, 2001.
[2] S. Graillat, Ph. Langlois, N. Louvet, Compensated Horner Scheme. *Submitted for publication*, July 2005. Available at http://webdali.univ-perp.fr/RR/rr2005-04.pdf.
[3] T. Ogita, S. M. Rump, and S. Oishi, Accurate sum and dot product. *SIAM J. Sci. Comp.*, 26(6):1955-1988, 2005.