Fast Algorithms for High-Precision Computation of Elementary Functions (extended abstract)

Richard P. Brent
Australian National University
http://www.rpbrent.com

In many applications of real-number computation we need to evaluate elementary functions such as $\exp(x)$, $\ln(x)$, $\arctan(x)$ to high precision (see for example [1]). We shall survey some of the well-known (and not so well-known) techniques as well as mentioning some new ideas.

Let d be the number of binary digits required, so the computation should be accurate with relative (or, if appropriate, absolute) error $O(2^{-d})$. By "high-precision" we mean higher than can be obtained directly using IEEE 754 standard floating-point hardware, typically d several hundred up to millions.

We are interested both in "asymptotically fast" algorithms (the case $d \to +\infty$) and in algorithms that are competitive in some range of d. Let $M(d)$ denote the time (measured in word- or bit-operations) required to multiply d-bit numbers with d-bit accuracy (we are generally only interested in the upper half of the $2d$-bit product). Classically $M(d) = O(d^2)$ and the Schönhage-Strassen algorithm [12] shows that $M(d) = O(d \log d \log\log d)$. However, Schönhage-Strassen is only useful for large d, and there is a significant region $d_1 < d < d_2$ where A. Karatsuba's $O(d^{\lg 3})$ algorithm [8] is best ($\lg 3 = \log_2 3 \approx 1.58$). In the region where Karatsuba's algorithm is best for multiplication, the best algorithms for elementary functions need not be those that are asymptotically the fastest.

Sometimes the best algorithm depends on the ground rules: are certain constants such as π allowed to be precomputed, or does the cost of their computation have to be counted every time in the cost of the elementary function evaluation?

Techniques for high-precision elementary function evaluation include the following. Often several are used in combination, e.g. argument reduction is used before power series evaluation.

1. Argument reduction using identities such as

$$\exp x = (\exp(x/2))^2 \ , \ \ \arctan x = 2 \arctan\left(\frac{x}{1+\sqrt{1+x^2}}\right) \ , \ \ \text{etc.}$$

2. Use of power series such as

$$\exp x = \sum_{k \ge 0} \frac{x^k}{k!} \ , \ \ \ln(1+x) = \sum_{k \ge 0} \frac{(-1)^k x^{k+1}}{k+1} \ , \ \ \arctan x = \sum_{k \ge 0} \frac{(-1)^k x^{2k+1}}{2k+1} \ ,$$

perhaps evaluated using the technique of Smith [13], which applies more generally (for example to the evaluation of hypergeometric functions). Smith seems to be the first to apply his technique for real computation, but the idea was suggested by Paterson and Stockmeyer [10] and used in a different context by Brent and Kung [6] (but not in the author's multiple-precision package [5], because of the storage requirements).

By combining argument reduction and power series evaluation we get an $O(M(d)d^{1/2})$ algorithm for $\exp(x)$, and using Smith's technique this can be improved to $O(M(d)d^{1/3}) + O(d^{5/3} \log\log d \log\log\log d)$ (the second term is essentially $O(d^{5/3})$ in practice).

3. Use of the arithmetic-geometric mean (AGM) to compute $\ln x$ in time $O(M(d) \log d)$ (asymptotically the fastest known), see [2, 3, 4]. In particular we mention the algorithm of Sasaki and Kanada [11], based on the elegant formula
$$\ln x = \frac{\pi}{\mathrm{AGM}(\theta_2^2(1/x), \theta_3^2(1/x))} .$$
Because the theta functions have rapidly-converging series and this formula is exact, we can use it for smaller x than is possible with the usual "approximate" AGM-based formulae such as $\ln x = \pi/((2 + O(1/x^2))\mathrm{AGM}(1, 4/x))$.

4. Use of Newton's method to compute inverse functions, for example we can compute $\exp(x)$ from $\ln(x)$ and *vice versa*. The overhead introduced by Newton's method can be reduced to a factor $1 + o(1)$ as $d \to +\infty$ by using higher-order methods [3, §6-§9].

5. Use of complex arithmetic to compute a real result, for example
$$\arctan x = \frac{1}{2i} \ln\left(\frac{1 + ix}{1 - ix}\right) = \Im \ln(1 + ix) ,$$
where the complex log can be computed by the AGM; this gives the asymptotically fastest known algorithm for arctan (although the complex arithmetic is a significant overhead).

6. Use of *binary splitting* [2, p. 329] (or similarly E. Karatsuba's *FEE method* [9]) to sum series with rational arguments [7]. For real arguments, we may be able to use a good rational approximation and then apply a small correction. To illustrate this we shall describe some new ideas for arctan evaluation which, although not asymptotically the fastest, are competitive for a wide range of precisions d (this is joint work with Jim White).

References

[1] D. H. Bailey, High-precision floating-point arithmetic in scientific computation, *Computing in Science and Engineering*, May–June 2005, 54–61; also report LBNL–57487. Available from `http://crd.lbl.gov/~dhbailey/dhbpapers/` .

[2] J. M. Borwein and P. B. Borwein, *Pi and the AGM*, Monographies et Études de la Société Mathématique du Canada, John Wiley & Sons, Toronto, 1987.

[3] R. P. Brent, Multiple-precision zero-finding methods and the complexity of elementary function evaluation, in *Analytic Computational Complexity* (edited by J. F. Traub), Academic Press, New York, 1975, 151–176. Available from `http://wwwmaths.anu.edu.au/~brent/pub/pub028.html` .

[4] R. P. Brent, Fast multiple-precision evaluation of elementary functions, *J. ACM* **23** (1976), 242–251.

[5] R. P. Brent, Algorithm 524: MP, a Fortran multiple-precision arithmetic package, *ACM Trans. Math. Software* **4** (1978), 71–81.

[6] R. P. Brent and H. T. Kung, Fast algorithms for manipulating formal power series, *J. ACM* **25** (1978), 581–595.

[7] X. Gourdon and P. Sebah, Numbers, constants and computation: binary splitting method, `http://numbers.computation.free.fr/Constants/Algorithms/splitting.html` .

[8] A. Karatsuba and Y. Ofman, Multiplication of multidigit numbers on automata (in Russian), *Doklady Akad. Nauk SSSR* **145** (1962), 293–294. English translation in *Sov. Phys. Dokl.* **7** (1963), 595–596.

[9] E. A. Karatsuba, Fast evaluation of transcendental functions (in Russian), *Probl. Peredachi Inf.* **27**, 4 (1991), 87–110. English translation in *Problems of Information Transmission* **27** (1991), 339–360. See also `http://www.ccas.ru/personal/karatsuba/faqen.htm` .

[10] M. S. Paterson and L. J. Stockmeyer, On the number of nonscalar multiplications necessary to evaluate polynomials, *SIAM J. Computing* **2** (1973), 60–66.

[11] T. Sasaki and Y. Kanada, Practically fast multiple-precision evaluation of $\log(x)$, *J. Inf. Process.* **5** (1982), 247–250. See also [2, §7.2].

[12] A. Schönhage and V. Strassen, Schnelle Multiplikation Grosser Zahlen, *Computing* **7** (1971), 281–292.

[13] D. M. Smith, Efficient multiple-precision evaluation of elementary functions, *Math. Comp.* **52** (1989), 131–134.