

**OPTIMIZATION OF RATIONAL APPROXIMATIONS  
BY CONTINUED FRACTIONS\***

Frithjof Blomquist

ABSTRACT. To get guaranteed machine enclosures of a special function  $f(x)$ , an upper bound  $\varepsilon(f)$  of the relative error is needed, where  $\varepsilon(f)$  itself depends on the error bounds  $\varepsilon(\text{app})$ ,  $\varepsilon(\text{eval})$  of the approximation and evaluation error respectively. The approximation function  $g(x) \approx f(x)$  is a rational function (Remez algorithm), and with sufficiently high polynomial degrees  $\varepsilon(\text{app})$  becomes sufficiently small. Evaluating  $g(x)$  on the machine produces a rather great  $\varepsilon(\text{eval})$  because of the division of the two erroneous polynomials. However,  $\varepsilon(\text{eval})$  can distinctly be decreased, if the rational function  $g(x)$  is substituted by an appropriate continued fraction  $c(x)$  which in general needs less elementary operations than the original rational function  $g(x)$ . Numerical examples will illustrate this advantage.

**1. Introduction.** In general the exact sum  $a+b$  of two machine numbers  $a$ ,  $b$  is not a machine number itself and must therefore be rounded to the IEEE system S(2,53). If the rounding is realized to one of the neighboring machine numbers, denoted by  $a \oplus b$ , the relative error  $\varepsilon_{a+b}$  is defined by  $(a+b) - (a \oplus b) = \varepsilon_{a+b} \cdot (a+b)$  and it holds [2]

$$|\varepsilon_{a+b}| \leq \varepsilon^* := 2^{-52} = 2.220446 \dots \cdot 10^{-16}, \quad (\text{high accuracy}).$$

---

*ACM Computing Classification System* (1998): G.1, G.1.1, G.1.2, G.1.10, G.4.

*Key words:* C-XSC, continued fractions, error bounds, Special Functions.

\*The paper has been presented at the 12th International Conference on Applications of Computer Algebra, Varna, Bulgaria, June, 2006.

$\varepsilon^*$  is the error bound of the elementary operations in high accuracy.

If  $\tilde{f}(x)$  denotes the machine approximation of the exact function value  $f(x)$ , the relative error  $\varepsilon_f$  is defined by  $\tilde{f}(x) - f(x) = \varepsilon_f \cdot f(x)$  and for the error bound  $\varepsilon(f)$  it holds for all machine numbers  $x$  of the domain  $D_f$

$$|\varepsilon_f| \leq \varepsilon(f) \quad \forall x \in D_f \cap S(2, 53).$$

If  $f(x)$  is approximated by  $g(x) \approx f(x)$  and if  $\tilde{g}(x)$  denotes the machine value of  $g(x) \approx \tilde{g}(x)$  then  $\tilde{f}(x) = \tilde{g}(x)$  and with  $f(x) \approx g(x) \approx \tilde{g}(x)$ , together with the definitions

$$\begin{aligned} f(x) - g(x) &= \varepsilon_{\text{app}}(x) \cdot f(x), & |\varepsilon_{\text{app}}(x)| &\leq \varepsilon(\text{app}) \quad \forall x \in D_f, \\ g(x) - \tilde{g}(x) &= \varepsilon_{\text{eval}} \cdot g(x), & |\varepsilon_{\text{eval}}| &\leq \varepsilon(\text{eval}, g) \quad \forall x \in D_f \end{aligned}$$

the triangle inequality delivers

$$(1) \quad \varepsilon(f) = \varepsilon(\text{app}) + \varepsilon(\text{eval}, g) \cdot [1 + \varepsilon(\text{app})] \quad \forall x \in D_f \cap S(2, 53),$$

so guaranteed values of the upper bounds  $\varepsilon(\text{app})$  and  $\varepsilon(\text{eval}, g)$  are needed for calculating a guaranteed upper bound  $\varepsilon(f)$  of the relative error  $\varepsilon_f$ .

Let  $[a, b] \subset D_f$  be a given approximation interval, where  $f(x)$  is a positive and monotonic increasing function then a guaranteed enclosure of all function values  $f(x)$  is given by the following interval

$$f(x) \in \left[ \frac{\tilde{f}(a)}{1 + \varepsilon(f)}, \frac{\tilde{f}(b)}{1 - \varepsilon(f)} \right] \quad \forall x \in D_f \cap S(2, 53),$$

so  $\varepsilon(f)$ , depending on  $\varepsilon(\text{app})$  and  $\varepsilon(\text{eval}, g)$ , is essential for implementing interval functions, and for tight enclosures of  $f(x)$  small values of  $\varepsilon(\text{app})$  and  $\varepsilon(\text{eval}, g)$  are needed. The principles for calculating guaranteed upper bounds of  $\varepsilon(\text{app})$  and  $\varepsilon(\text{eval}, g)$  are given in [2], where a lot of examples can be found.

**2. Approximation Error.** In contrast to the elementary functions by special functions the approximation intervals  $[a, b]$  are widespread, not containing the origin in general. Therefore a rational function

$$g(x) := \frac{P_M(x - x_0)}{Q_N(x - x_0)} \approx f(x), \quad M, N \in \mathbb{N},$$

calculated with the Remez algorithm, delivers an optimal approximation of  $f(x)$ , i.e.,  $\varepsilon(\text{app})$  is for example much smaller than the appropriate error bound, calculated with the Padé algorithm using the same values of  $M, N$ . The typical

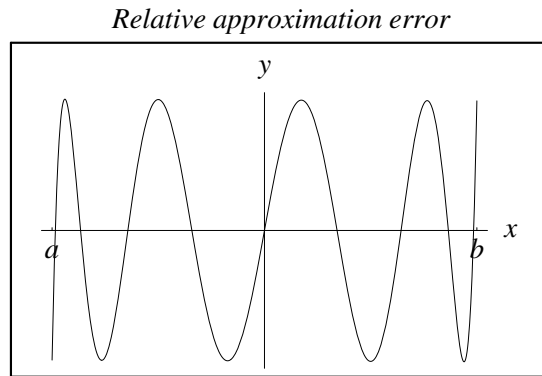


Fig. 1. Best approximation with  $g(x)$ :  $\varepsilon_{\text{app}}(x)$ ,  $M = N = 4$ .

behavior of the relative approximation error  $\varepsilon_{\text{app}}(x)$ , caused by best approximation with  $M = N = 4$ , is shown in Figure 1. There are  $M + N + 2 = 10$  equal values of the absolute extrema of  $\varepsilon_{\text{app}}(x)$ . The coefficients of the polynomials  $P_M, Q_N$  are calculated with the algebra system *Mathematica*, using a precision of 50 decimal digits, [5]. However, in practice these coefficients must be rounded to the IEEE system and hereby the maximum of  $|\varepsilon_{\text{app}}(x)|$  will be enlarged roughly by factor 2. The typical influence of this rounding is shown in Figure 2, where the rounding is done to 17, 16, 15 and 14 decimal digits. For the best approximation

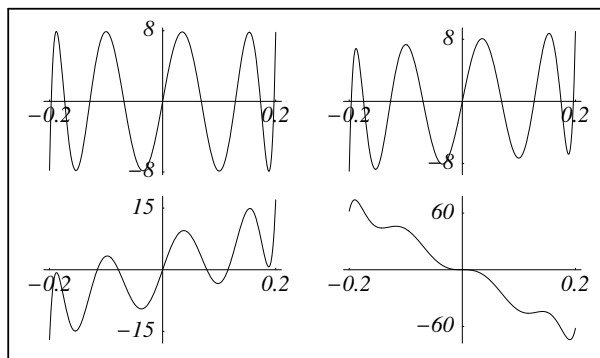


Fig. 2.  $\varepsilon_{\text{app}}(x)$ ; rounding the coefficients to 17, 16, 15, 14 decimal digits

with  $g(x)$  *Mathematica* delivers an approximation error, which however can only be used as an estimated value to realize the proper polynomial degrees  $M, N$ , because there is no information about the algorithm used. Furthermore, concerning the described rounding effects, the approximation error must be re-evaluated to get a reliable value of the upper bound  $\varepsilon(\text{app})$ , [4]. With a C-XSC program a guaranteed upper bound  $\varepsilon(\text{app})$  can be calculated, [2]. So, with

sufficiently high values of  $M$ ,  $N$ ,  $\varepsilon(\text{app})$  can be treated as a nearly optimal and reliable value.

**3. Evaluation Error.** In [2] a C-XSC library is described for calculating reliable upper bounds  $\varepsilon(\text{eval})$  of the evaluation errors concerning the elementary operations. Additionally C-XSC functions are implemented for calculating guaranteed upper bounds of the relative evaluation errors, if polynomials or rational functions are evaluated in the IEEE system.

In practice, for not too wide intervals  $[a, b]$  and with a proper point of expansion  $x_0 \in [a, b]$  for polynomials we have  $\varepsilon(\text{eval}) \sim 4 \cdot \varepsilon^*$ . Hence, for a rational approximation function  $g(x)$  it holds  $\varepsilon(\text{eval}, g) \sim (2 \cdot 4 + 1) \cdot \varepsilon^* = 9 \cdot \varepsilon^*$ . In comparison with the upper bound of the approximation error, which should be  $\varepsilon(\text{app}) \sim 0.1 \cdot \varepsilon^*$ , the upper bound  $\varepsilon(\text{eval}, g)$  is much too large and should ideally be reduced to  $\varepsilon(\text{eval}) \sim 1 \cdot \varepsilon^*$ .

To get such a small evaluation error  $\varepsilon(\text{eval}) \sim \varepsilon^*$ , the approximation function  $g(x) \approx f(x)$  should be a sum

$$(2) \quad S + s(x - x_0), \quad S, x, x_0 \in S(2, 53), \quad \text{with}$$

$$(3) \quad |S| \gg |s(x - x_0)|,$$

where  $S$  must be an error-free summand, which should be rather great in comparison to the erroneous second summand  $s(x - x_0)$ , [2].

**4. Approximation with Continued Fractions.** Starting from the rational function  $g(x) = P_M(x - x_0)/Q_N(x - x_0)$  the question is now, how to get a nearly equivalent sum  $S + s(x - x_0) \approx g(x)$ , fulfilling the condition (3).  $S + s(x - x_0)$  must be nearly equivalent to  $g(x)$  in order to keep the small approximation error  $\varepsilon(\text{app}) \sim 0.1 \cdot \varepsilon^*$ .

To achieve the sum (2) we perform simple polynomial divisions, where two strategies can be pursued

1. Discarding in  $P_M(x - x_0)$  the summand with the highest exponent  $M$ ,
2. Discarding in  $P_M(x - x_0)$  the summand with the exponent 0.

Strategy 1. is demonstrated with the following example, setting  $u = (x - x_0)$

$$(4) \quad g(x) = (2 + 4u - 2u^2) : (1 - u + u^2) = -2 + \frac{4 + 2u}{1 - u + u^2} \\ = -2 + \frac{1}{\frac{1 - u + u^2}{4 + 2u}}$$

$$\begin{aligned}
 &= -2 + \frac{1}{\frac{1}{2}u - \frac{3}{2} + \frac{7}{4+2u}} \\
 (5) \quad &= -2 + \frac{2}{u - 3 + \frac{7}{u+2}}
 \end{aligned}$$

In (4), with  $S = -2$ , we have the desired structure (2). However, the condition (3) is only fulfilled for  $|u| \rightarrow +\infty$ . With strategy 2. we get

$$(6) \quad g(x) = (2 + 4u - 2u^2) : (1 - u + u^2) = 2 + \frac{u}{\frac{1}{6} + \frac{u}{-18 + \frac{u}{\frac{-1}{42} + \frac{u}{14}}}}$$

now with  $S = +2$ , and condition (3) is fulfilled for  $|u| \rightarrow 0$ , i.e. for  $x \rightarrow x_0$ . To get a short runtime we should use (4), because here we only need 6 elementary operations instead of 8 operations in (6).

However, in (4) we should have  $|u| \rightarrow +\infty$  for  $x \rightarrow x_0$ . This problem is solved by using the transformation  $u = 1/v$  in the rational function  $g(x)$ , and a subsequent polynomial division with strategy 1. and  $v := 1/(x - x_0)$  delivers

$$(7) \quad g(x) = (-2 + 4v + 2v^2) : (1 - v + v^2) = 2 + \frac{6}{v - \frac{1}{3} + \frac{7/9}{v - \frac{2}{3}}} =: c(v).$$

In (7)  $x \neq x_0$  must be realized, and  $g(x_0) := 2$  is a continuous supplementation. Now we have the desired result:  $|v| \rightarrow +\infty$  for  $x \rightarrow x_0$ . Hence, with the continued fraction  $c(v)$  on the right-hand side in (7) we will find rather small evaluation errors using not too wide approximation intervals with a suitable point of expansion  $x_0 \in [a, b]$ .

Here still a closing remark for calculating the approximation error. Starting with  $c(v)$ , defined in (7), we first have to realize a rational interval function  $r(x)$  enclosing this finite continued fraction. The task can be done using the recurrence formula in [6, pp. 175–177], and with a C-XSC program a guaranteed upper bound  $\varepsilon(\text{app})$  of the relative approximation error can be calculated, [1, 2].

**5. Numerical Examples.** In this section the improvement of the evaluation error is demonstrated by using a continued fraction analogously to (7) instead of the rational function  $g(x) = P_M(x - x_0)/Q_N(x - x_0)$ . The special

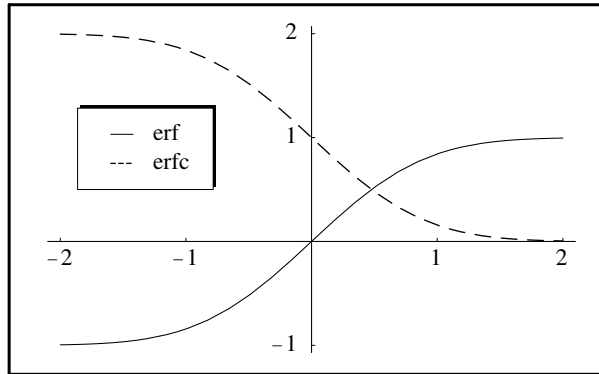


Fig. 3. erf(x) and erfc(x)

functions to be approximated are the error function erf(x) and the complementary error function erfc(x).

The two functions erf(x) and erfc(x) are defined by

$$\operatorname{erf}(x) := \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt, \quad \operatorname{erfc}(x) := 1 - \operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_x^\infty e^{-t^2} dt, \quad x \in \mathbb{R};$$

**5.1. Approximation of erfc(x),  $x \in [14, 26.5]$ .** In the wide approximation interval  $[a, b] = [14, 26.5]$  we consider the asymptotic expansion, [1, formulae 7.1.23, 7.1.24]

$$\operatorname{erfc}(x) = \frac{e^{-x^2}}{\sqrt{\pi} \cdot x} \left[ 1 - \frac{1}{(2x^2)^1} + \frac{1 \cdot 3}{(2x^2)^2} - + \dots + (-1)^N \cdot \frac{1 \cdot 3 \cdots (2N - 1)}{(2x^2)^N} + r \right]$$

$$|r| = |r(x, N)| \leq \frac{1 \cdot 3 \cdot 5 \cdots (2N + 1)}{(2x^2)^{N+1}}, \quad N = 1, 2, 3, \dots$$

With the factor  $e^{-x^2}$  in the above asymptotic expansion erfc(x) is a strongly decreasing function. Hence, the following approximation  $\operatorname{erfc}(x) \approx c_N(v)$  with a continued fraction  $c_N(v)$  of length  $N$  will fail, because  $c_N(v)$  itself is a strongly decreasing function, which consequently can not fulfill the condition (3) for a small evaluation error. However, in contrast to erfc(x), the quotient  $\operatorname{erfc}(x)/e^{-x^2}$  is a nearly constant function, which can successfully be approximated by a continued fraction  $c_4(v)$  of length 4;  $v := 1/(x - x_0)$ ,  $x_0 := 20.5$ ;

$$(8) \quad \frac{\operatorname{erfc}(x)}{e^{-x^2}} \approx c_4(v) := b_0 + \frac{a_1}{v + b_1 + \frac{a_2}{v + b_2 + \frac{a_3}{v + b_3 + \frac{a_4}{v + b_4}}}}, \quad x \neq x_0.$$

The calculation of the rational function  $P_4(x-x_0)/Q_4(x-x_0)$ , the transformation  $u := x - x_0 = 1/v$  and the computation of the  $a_k, b_k$  is done with the algebra system *Mathematica*. Approximations of the  $a_k, b_k \in S(2, 53)$  are listed in the following table:

$k$	$a_k := \text{nearest}(\cdot)$	$b_k := \text{nearest}(\cdot)$
0	$+0.000000000000000000 \cdot 10^{+0}$	$+2.748881515193487221 \cdot 10^{-2}$
1	$-1.337745866182817076 \cdot 10^{-3}$	$+4.860780872578862971 \cdot 10^{-2}$
2	$+2.771654901614425610 \cdot 10^{-6}$	$+4.826766715012656847 \cdot 10^{-2}$
3	$+5.428546251910422025 \cdot 10^{-6}$	$+4.793524916454342483 \cdot 10^{-2}$
4	$+7.982629192430865797 \cdot 10^{-6}$	$+4.740017176613045964 \cdot 10^{-2}$

Table 1. Approximations of the  $a_k, b_k \in S(2, 53)$  with 19 decimal digits.

Notice, that  $\text{erfc}(x_0)/e^{-x_0^2}$  will be approximated by  $b_0 = 2.74888151519 \dots \cdot 10^{-2}$ . As  $\text{erfc}(x)/e^{-x^2}$  and  $c_4(v)$  are nearly constant functions in  $[a, b]$ , the condition (3) for a small evaluation error of  $c_4(v)$  will be fairly well fulfilled, and a C-XSC program delivers the guaranteed upper bound  $\varepsilon(\text{eval}, c_4(v))$  of the evaluation error

$$(9) \quad \varepsilon(\text{eval}, c_4(v)) = 5.353163 \cdot 10^{-16} \approx 2.41 \cdot \varepsilon^*.$$

If the approximation is done by the rational function  $g(x)$

$$\frac{\text{erfc}(x)}{e^{-x^2}} \approx g(x) := \frac{P_4(x-x_0)}{Q_4(x-x_0)} \approx c_4(v), \quad x \in [a, b],$$

with the same point of expansion  $x_0 = 20.5$ , then with another C-XSC program we get the guaranteed upper bound  $\varepsilon(\text{eval}, g(x))$  of the evaluation error

$$(10) \quad \varepsilon(\text{eval}, g(x)) = 3.469925 \cdot 10^{-15} \approx 15.6 \cdot \varepsilon^*.$$

Comparing the results in (9) and (10) we get an improvement of the evaluation error by the factor 6.5 using the continued fraction  $c_4(v)$  instead of the rational function  $g(x)$ .

Up to now we have approximated only the quotient  $\text{erfc}(x)/e^{-x^2}$ . However, in practice  $\text{erfc}(x)$  is evaluated by  $h(x)$ :

$$\text{erfc}(x) \approx h(x) := e^{-x^2} \cdot c_4(v), \quad v = \frac{1}{x - x_0}.$$

The machine value  $\tilde{h}(x)$  is defined by

$$\tilde{h}(x) := \text{expm2}(x) \odot \tilde{c}_4(\tilde{v}(x)), \quad \tilde{v}(x) := 1 \oslash (x \ominus x_0),$$

where  $\text{expmx2}(x)$  is the C-XSC function for calculating  $e^{-x^2}$ , provided with the error bound  $\varepsilon(e^{-x^2}) = 4.618919 \cdot 10^{-16}$ .  $\{\oplus, \ominus, \odot, \oslash\}$  is the set of the erroneous floating-point operators. The upper bound  $\varepsilon(\text{eval}, h(x))$  of the relative evaluation error  $\varepsilon_{\text{eval}}$  is defined by

$$\tilde{h}(x) - h(x) = \varepsilon_{\text{eval}} \cdot h(x), \quad |\varepsilon_{\text{eval}}| \leq \varepsilon(\text{eval}, h(x)) = 1.233494 \cdot 10^{-15},$$

and calculated with a special C-XSC program.

Using the upper bound  $\varepsilon(\text{app}, \text{erfc}(x)) = 7.7344 \cdot 10^{-17}$  of the approximation error [2], together with (1) we finally get

$$\varepsilon_{\text{erfc}} := \frac{\text{erfc}(x) - \tilde{h}(x)}{\text{erfc}(x)}, \quad |\varepsilon_{\text{erfc}}| \leq \varepsilon(\text{erfc}(x)) = 1.3109 \cdot 10^{-15} \quad \forall x \in [a, b].$$

**5.2. Approximation of  $\text{erf}(x)$ ,  $x \in [4.75, 6]$ .** As can be seen in Figure 3,  $\text{erf}(x)$  is a nearly constant function for  $x \in [a, b] = [4.75, 6]$ . Hence, an approximation with the continued fraction  $c_5(v)$

$$(11) \quad \text{erf}(x) \approx c_5(v) := b_0 + \frac{a_1}{v + b_1 + \frac{a_2}{v + b_2 + \frac{a_3}{v + b_3 + \frac{a_4}{v + b_4 + \frac{a_5}{v + b_5}}}}}$$

$$v = \frac{1}{x - x_0}, \quad x \neq x_0, \quad x_0 = \frac{43}{8} = 5.375;$$

will lead to a rather small evaluation error of  $c_5(v)$ , because condition (3) is fulfilled now very well. As in section 5.1 the calculation of the rational function  $P_4(x-x_0)/Q_4(x-x_0)$ , the transformation  $u := x-x_0 = 1/v$  and the computation of the  $a_k, b_k$  is done with the algebra system *Mathematica*. Approximations of the  $a_k, b_k \in S(2, 53)$  are listed in the Table 2.

Notice that  $\text{erf}(x_0)$  will be approximated by  $b_0 = 9.99999999999970 \dots \cdot 10^{-1}$  without any evaluation error. With the erroneous machine value  $\tilde{c}_5(\tilde{v})$  the relative evaluation error  $\varepsilon_{\text{eval}}$  is defined by

$$\varepsilon_{\text{eval}} = \frac{\tilde{c}_5(\tilde{v}) - c_5(v)}{c_5(v)}, \quad \tilde{v} = 1 \oslash (x \ominus x_0).$$

A C-XSC program delivers for the upper bound  $\varepsilon(\text{eval}, c_5(v))$  the following result

$$(12) \quad |\varepsilon_{\text{eval}}| \leq \varepsilon(\text{eval}, c_5(v)) = 2.220447 \cdot 10^{-16} \approx 1 \cdot \varepsilon^*.$$



$k$	$a_k := \text{nearest}(\cdot)$	$b_k := \text{nearest}(\cdot)$
0	$+0.000000000000000000 \cdot 10^{+0}$	$+9.999999999999707074 \cdot 10^{-1}$
1	$+3.201486811957019238 \cdot 10^{-13}$	$+5.376690224467207768 \cdot 10^{+0}$
2	$+9.971477472292114810 \cdot 10^{+0}$	$-8.665555788956434789 \cdot 10^{-2}$
3	$+2.021756014259896991 \cdot 10^{+0}$	$-1.023626941358960172 \cdot 10^{-1}$
4	$+9.110335999780354109 \cdot 10^{-1}$	$-2.340999377105155262 \cdot 10^{-1}$
5	$+4.483072053115112668 \cdot 10^{-1}$	$-4.994571201677685505 \cdot 10^{-1}$

Table 2. Approximations of the  $a_k, b_k \in S(2, 53)$  with 19 decimal digits.

As we have already supposed, with the continued fraction  $c_5(v)$  we now get a rather small and optimal upper bound of the relative evaluation error.

If the approximation is done by the rational function  $g(x)$

$$\text{erf}(x) \approx g(x) := \frac{P_5(x - x_0)}{Q_5(x - x_0)} \approx c_5(v), \quad x \in [a, b],$$

with the point of expansion  $x_0 = 4.875$ , then with another C-XSC program we get the guaranteed upper bound  $\varepsilon(\text{eval}, g(x))$  of the evaluation error

$$(13) \quad \varepsilon(\text{eval}, g(x)) = 3.450345 \cdot 10^{-15} \approx 15.5 \cdot \varepsilon^*.$$

Comparing the results in (12) and (13) we get an improvement of the evaluation error by the factor 15.5 using the continued fraction  $c_5(v)$  instead of the rational function  $g(x)$ .

With the upper bound  $\varepsilon(\text{app}, \text{erf}(x)) = 2.0982 \cdot 10^{-17}$  of the approximation error [2], together with (1) we finally get a rather small error bound:

$$\varepsilon_{\text{erf}} := \frac{\text{erf}(x) - \tilde{c}_5(\tilde{v})}{\text{erf}(x)}, \quad |\varepsilon_{\text{erf}}| \leq \varepsilon(\text{erf}(x)) = 2.4303 \cdot 10^{-16} \quad \forall x \in [a, b].$$

Both examples demonstrate a drastic reduction of the evaluation error by using a continued fraction of structure (7) instead of an equivalent rational approximation function. Furthermore the runtime can be reduced, because the evaluation of  $c_5(v)$  in (11) needs only 17 elementary operations in contrast to the rational function  $P_5(x - x_0)/Q_5(x - x_0)$ , which requires 22 operations.

**6. Conclusions.** Starting the approximation of a given function  $f(x)$  in a special, wide interval  $[a, b]$  with a rational function  $g(x) \approx f(x)$ , the evaluation error can be significantly improved by using an appropriate continued fraction

$c_k(v)$  based on  $g(x) \approx c_k(v)$ , with  $v := 1/(x - x_0)$ . Furthermore the runtime can be reduced, provided that multiplication and division have nearly the same runtime. The discussed functions  $c_k(v)$  must not be confounded with the well-known continued fractions of some special functions valid for  $x \rightarrow 0$  or  $|x| \rightarrow \infty$ , [6].

## REFERENCES

- [1] ABRAMOWITZ M., I. A. STEGUN. Handbook of Mathematical Functions, with Formulas, Graphs, and Mathematical Tables. Dover Publications, Inc., New York, 1972.
- [2] BLOMQUIST F. Entwicklung von Werkzeugen zur automatischen Berechnung von a priori Fehlerschranken und deren Anwendung bei der Realisierung von hochgenauen mathematischen Funktionen in Rechenanlagen. Fachbereich 7, Mathematik, Bergische Universität Wuppertal,] 2006. [www.math.uni-wuppertal.de/org/WRST/index\\_de.html](http://www.math.uni-wuppertal.de/org/WRST/index_de.html)
- [3] HOFSCHESTER W., KRÄMER WALTER. C-XSC 2.0: A C++ Library for Extended Scientific Computing. In: Numerical Software with Result Verification (Eds Alt René., A. Frommer, Kearfott R. Baker, W. Luther). Lect. Notes Comput. Sci. Vol. **2991/2004**, Springer-Verlag, Heidelberg, 2004, 15–35.
- [4] KRÄMER W. Sichere und genaue Abschätzung des Approximationsfehlers bei rationalen Approximationen. Report of the Institut für Angewandte Mathematik, Universität Karlsruhe, Germany, 1996.
- [5] Wolfram Research *Mathematica*, Computer algebra system.
- [6] PRESS W. H. et al. Numerical Recipes in C, The Art of Scientific Computing, second edition. Cambridge University Press, 1992.

University of Wuppertal  
 Scientific Computing/Software Technology  
 Gaußstr. 20  
 D-42097 Wuppertal, Germany

Received Desember 25, 2006  
 Final Accepted October 18, 2007