Interval Computations
No 4, 1994

Computation of Standard Interval Functions in Multiple-Precision Interval Arithmetic

Wolfram Luther and Werner Otten

We present quadratic convergent algorithms for the computation of standard interval functions like $\sqrt{\ }$, ln, arctan in a multiple-precision interval arithmetic. These algorithms depend on elliptic integrals of the first and second kind and the method of arithmetic and geometric means.

Вычисление стандартных интервальных функций в интервальной арифметике многократной разрядности

В. Лутер, В. Оттен

Представлены квадратично сходящиеся алгоритмы для вычисления стандартных интервальных функций, таких как $\sqrt{\ }$, ln, arctg в интервальной арифметике многократной разрядности. Эти алгоритмы построены с использованием эллиптических интегралов первого и второго рода и метода арифметического и геометрического средних.

1 Introduction

Recently, W. Krämer [10] has proposed the iteration method of arithmetic-geometric means (AGM) to calculate guaranteed bounds for the elliptic integrals of first and second kind. To obtain these bounds, it is necessary to use a module which makes the elementary operations available in an interval arithmetic with arbitrary precision (such as implemented in *TPX* for *Turbo-Pascal* [8], *C-XSC*, or *PASCAL–XSC* [11]). Moreover, one needs constants like π and the functions $\sqrt{\ }$, sin, tan, arctan in full precision.

The aim of our paper is to discuss the suitability of well-known algorithms based on AGM- and *Newton*-methods (cf. [2, 3, 5, 12]) to calculate enclosures of the elliptic functions and $\sqrt{\ }$, ln, arctan and their inverse functions. For that purpose it is necessary to give error estimations with explicit constants for the AGM-methods introduced by Brent [5]. Rigorous a priori error estimates have been made by considering errors inherent in the floating-point representation as well as round-off errors in arithmetic operations and the approximation error. To speed up the algorithms it is possible to carry out the calculations with a multiple-precision point arithmetic and guarantee the results by using the a priori error bounds.

The special interest in the AGM-method arises from the quadratic convergence of these algorithms, so that fast calculations with a multiple-precision arithmetic is practicable. The known algorithms (for example *Taylor* approximations) exhibits only linear convergence and need a complicated argument reduction. Thus, the complexity to calculate the standard functions is of order $O\big(pM(p)\big)$ where $M(p)$ denotes the cost to multiply or divide two numbers with p mantissa digits. The AGM-method requires only $O(\operatorname{ld} p)$ steps and reduces the time for obtaining standard and elliptic functions to the order $O\big(\operatorname{ld} p \cdot M(p)\big)$.

The proposed algorithms have been implemented and tested using the precompiler *TPX* [8] and PASCAL–XSC together with the long interval module *mpi_ari* [11]. They are formulated in a *Pascal*-like pseudo-code.

To accelerate the convergence of our algorithms it is possible to use results of existing double arithmetic implementations of the standard functions as starting points.

2 Elliptic integrals

First, we will give some basic formulas for elliptic integrals which are needed in the later paragraphs.

The (in)complete elliptic integrals of first and second kind are defined as:

$$\begin{aligned}
K\big(\sin(\alpha),\psi\big) &= \int_0^{\psi} \frac{d\phi}{\sqrt{1-\sin^2\alpha\sin^2\phi}}, \qquad 0\le\psi\le\tfrac{\pi}{2},\ 0\le\alpha<\tfrac{\pi}{2},\\
E\big(\sin(\alpha),\psi\big) &= \int_0^{\psi} \sqrt{1-\sin^2\alpha\sin^2\phi}\,d\phi, \quad 0\le\psi\le\tfrac{\pi}{2},\ 0\le\alpha<\tfrac{\pi}{2},\\
K\big(\sin(\alpha)\big) &:= K\left(\sin(\alpha),\tfrac{\pi}{2}\right),\\
E\big(\sin(\alpha)\big) &:= E\left(\sin(\alpha),\tfrac{\pi}{2}\right).
\end{aligned}$$

Then with $k := \sin\alpha$, $k' := \sqrt{1-k^2}$ and $K'(k) := K(k')$, $E'(k) := E(k')$ the *Legendre* relation

$$E(k)K'(k) + K(k)E'(k) - K(k)K'(k) = \pi/2, \quad 0<k<1 \tag{1}$$

[2, p. 24] holds.

Lemma 1. *Let $0\le\psi<\pi/2$. Then the following inequalities are true:*

$$\psi \;\le\; K(\sin\alpha,\psi) \;\le\; \psi(1+\sin^2\alpha\,\psi^2/3) \quad \psi\le\pi/4\,, \tag{2}$$

$$\ln\tan\left(\frac{\pi}{4}+\frac{\psi}{2}\right) \;\ge\; K(\sin\alpha,\psi) \ge \ln\tan\left(\frac{\pi}{4}+\frac{\psi}{2}\right)\left(1-\frac{\cos^2\alpha}{2}\tan^2\psi\right) \tag{3}$$

The proof of (2) depends on the inequality $1/\sqrt{1-x}\le 1+x$, $0\le x<1/2$.

To prove (3) we use $K(1,\psi)=\ln\tan(\pi/4+\psi/2)$, the inequality

$$\begin{aligned}
0 \;&\le\; \frac{1}{\sqrt{1-\sin^2\phi}} - \frac{1}{\sqrt{1-\sin^2\alpha\sin^2\phi}} \le \frac{\cos^2\alpha}{2}\tan^2\phi\frac{1}{\sqrt{1-\sin^2\phi}},\\
0 \;&\le\; \phi<\frac{\pi}{2}
\end{aligned}$$

and an estimation of the difference $K(\sin\alpha,\ \psi)-K(1,\ \psi)$, $0\le\psi<\pi/2$. Furthermore, the following lemmata are valid, some of their proofs can be found in [2, 3, 5, 12].

Lemma 2. *(cf.* [2, *p.* 15], [5]*) Given the so-called arithmetic-geometric mean (AGM) iteration sequences are*

$$\begin{array}{rclcrclcrcl} a_0 & = & 1, & & b_0 & = & \cos\alpha = k' & & c_0 & = & k, \\ a_{i+1} & := & \dfrac{a_i + b_i}{2}, & & b_{i+1} & := & \sqrt{a_i \cdot b_i}, & & c_{i+1} & := & a_i - a_{i+1}. \end{array}$$

Then $\{b_i,\ a_i\}$ *creates a nested sequence of intervals with limit* ξ *and it holds:*

$$K(k) = \frac{\pi}{2\xi}\,, \tag{4}$$

$$E(k) = K(k) \cdot \left(1 - \sum_{i=0}^{\infty} 2^{i-1} c_i^2\right). \tag{5}$$

Further, with $s_i := b_i/a_i$, $b_0 = k$, *it follows:*

$$\lim_{i\to\infty} \prod_{j=0}^{i} \frac{1+s_j}{2} = \xi = \frac{\pi}{2K(k')}.$$

If we now use the formulas (4) and (5) together with the *Legendre* relation (1) and $k = 1/\sqrt{2}$ we get an algorithm for computing π (cf. [5, p. 246]).

Lemma 3. *Landen-transformation.* ([2, *p.* 12; , 5, *p.* 245; 12, *p.* 78])

Let $\{\alpha_i\}$, $\{\psi_i\}$ *be two given sequences with* $0 < \alpha_i < \alpha_{i+1} < \pi/2$, $0 \le \psi_{i+1} < \psi_i \le \pi/2$, *satisfying the equations* $1 + \cos\alpha_{i+1} = 2/(1+\sin\alpha_i)$ *and* $\sin(2\psi_{i+1} - \psi_i) = \sin\alpha_i \sin\psi_i$. *Then the following holds for the incomplete elliptic integral of first kind:*

$$K(\sin\alpha_{i+1},\ \psi_{i+1}) = \frac{1+\sin\alpha_i}{2} K(\sin\alpha_i,\ \psi_i).$$

The formula is also valid for $\psi_i = \pi$, $\psi_{i+1} = \pi/2$.

Note: To calculate the sequences $\{\alpha_i\}$, $\{\psi_i\}$ satisfying the conditions of *Landen*-transformation it is not necessary to solve the equations of Lemma 3. Using some conditions for trigonometric functions one finds the following procedure only based on rational expressions and square root evaluations to calculate α_i and ψ_i respectively.

Let $s_i := \sin\alpha_i$ and $v_i := \tan\psi_i/2$. Then we obtain:

$$\begin{aligned} s_{i+1} &= 2\sqrt{s_i}/(1+s_i), \\ \sin\psi_i &= 2v_i/(1+v_i^2), \\ w_1 &:= \sin(2\psi_{i+1}-\psi_i) = s_i\sin\psi_i, \\ w_2 &:= w_1/\left(1+\sqrt{1-w_1^2}\right), \\ w_3 &:= \tan\psi_{i+1} = \tan(\psi_{i+1}-\psi_i/2+\psi_i/2) = (w_2+v_i)/(1-v_iw_2), \\ v_{i+1} &= w_3/\left(1+\sqrt{1+w_3^2}\right). \end{aligned}$$

Lemma 4. ([1, *p.* 356]) *For* $k \in (0,1]$ *holds:*

$$\left| \ln\left(\frac{4}{k}\right) - K(k') \right| \le 4k^2\left(8+\ln\frac{1}{k}\right). \tag{6}$$

If we use the method of arithmetic and geometric means together with interval arithmetic we have to bear in mind the following two points.

We look at the endpoints of the intervals $A = [a_l, a_u]$, $B = [b_l, b_u]$ and calculate the difference of two succeeding sequence terms $\tilde{A} = [\tilde{a}_l, \tilde{a}_u]$, $\tilde{B} = [\tilde{b}_l, \tilde{b}_u]$:

$$\begin{aligned} \tilde{a}_l &= \frac{a_l+b_l}{2}, \quad \tilde{b}_l = \sqrt{a_l\,b_l}, \quad \tilde{a}_u = \frac{a_u+b_u}{2}, \quad \tilde{b}_u = \sqrt{a_u\,b_u}, \\ |\tilde{A}| &= \frac{|A|+|B|}{2} = \frac{|A|}{a_l}\,\tilde{a}_l\,\frac{a_l}{a_l+b_l} + \frac{|B|}{b_l}\,\tilde{a}_l\,\frac{b_l}{a_l+b_l}, \\ |\tilde{B}| &\sim \frac{|A|}{2a_l}\,\tilde{b}_l\,\frac{b_ua_l}{\tilde{b}_l^2} + \frac{|B|}{2b_l}\,\tilde{b}_l. \end{aligned}$$

1) In each step the relative interval diameters $|\tilde{A}|/\tilde{a}_l$, $|\tilde{B}|/\tilde{b}_l$ are weighted means of both predecessor diameters $|A|/a$, $|B|/b$, but we have to consider the additional error introduced by the square root evaluation ($\sim 1.5\cdot 2^{1-p}$).

2) If we use a precision of 2^{1-p}, a relative rounding error of $2\cdot 2^{1-p}$ appears in each step.

An estimation of the interval diameters yields the result:

$$|\tilde{a}_u - \tilde{b}_l| \le \frac{1}{8\tilde{b}_l}|a_u-b_l|^2 + \frac{|A|+|B|}{2}.$$

Therefore, the quadratic convergence is disturbed by a linear term. This phenomenon can be interpreted as follows: For a required precision of p binary digits we have to execute $k = const \cdot \operatorname{ld} p$ iteration steps. By accumulation of the square root error of w binary digits we get a total error of $\operatorname{ld} k \cdot w =: \kappa$ binary digits and after a few initialisation steps the convergence is quadratic until the precision reaches $p - \kappa$ binary digits. For the calculation of all constants and functions we use results coming from other interval functions, for example in the algorithm for π we need the interval square root, in the algorithm for $\ln 2$ we need π etc. An additional error accumulation occurs from further operations such as squaring, division and the joining of intervals. For this reason we have a loss of 7 binary digits in the calculation of π with $p = 320$ binary digits precision.

3 Basic error analysis

In this section we will give some basic formulas for error analysis. We have used them to calculate the a priori error estimates for our proposed algorithms given below.

Our considerations concern the floating-point screen

$$S^0 := S(\mathcal{B}, l^0, em^0, eM^0)$$

with its even base $\mathcal{B}$ (e.g. $\mathcal{B} = 2^{16}$ in *TPX* or $\mathcal{B} = 2^{32}$ in *PASCAL–XSC*), mantissa length l^0 and $[em^0, eM^0]$ smallest and largest allowable exponent, respectively. Computations require guard digits and are made in a finer screen

$$S := S(\mathcal{B}, l, em, eM),\ l \le l^0 + k,\ em \le em^0,\ eM \ge eM^0.$$

Directed roundings from the screen S to S^0 are necessary. The relative error for all elementary operations with machine numbers x, y is assumed to be bounded by

$$\frac{|x \times_l y - x \times y|}{|x \times y|} \le \epsilon < \mathcal{B}^{1-l}\ .$$

Setting $\epsilon(l) := \mathcal{B}^{1-l}$, which is referred to as screen epsilon, we assume $\epsilon(l) < 10^{-4}$, utilize the same notations as in [4, 9] and give some basic error estimations.

Define $x = \epsilon_l \Leftrightarrow x = \delta\epsilon(l)$, $|\delta| \le 1$. Assuming $0 < n^2\epsilon(l) < 5 \cdot 10^{-3}$, we have

$$(1+\epsilon_l)^{\pm n} \le 1 + (n + 0.01)\epsilon(l).$$

Furthermore,

$$\begin{aligned} \left|1 - \sqrt{1+\epsilon_l}\right| &\le \frac{\epsilon(l)}{2} + \frac{\epsilon(l)^2}{8} 1.0002, \\ \frac{1}{1+a\epsilon(l)} &= 1 - a\epsilon(l) + \frac{a^2}{1+a\epsilon(l)}\epsilon(2l-1), \\ |\alpha\epsilon_n \pm \beta\epsilon_k| &\le \big(|\alpha| + |\beta|\epsilon(k-n+1)\big)\epsilon(n). \end{aligned}$$

The following error bounds hold for rounded operations $+_l$, $/_l$, floating-point numbers a, b and their corresponding machine approximations $\tilde{a}$, $\tilde{b}$ with $|a - \tilde{a}| \le |a|\epsilon_a$ and $|b - \tilde{b}| \le |b|\epsilon_b$:

$$\begin{aligned} \frac{|a+b-(\tilde{a}+_l\tilde{b})|}{|a+b|} &\le \epsilon(l) + \left|\frac{1+\epsilon(l)}{a+b}\right| \{|a|\cdot\epsilon_a + |b|\cdot\epsilon_b\}, \\ \frac{|a/b-(\tilde{a}/_l\tilde{b})|}{|a/b|} &\le 1.01\{\epsilon_a + \epsilon_b + \epsilon(l)\}, \quad \epsilon_a, \epsilon_b, \epsilon(l) < 5 \cdot 10^{-3}. \end{aligned}$$

Cancellation occurs when the operants have different signs and are nearby quantities. The second formula also holds for multiplication.

Finally, assuming $\epsilon_a, \epsilon_b < 2 \cdot 10^{-3}$, $\epsilon := \max\{\epsilon_a,\ \epsilon_b,\ \epsilon(l)\}$, we have

$$\left|\frac{|\tilde{a}/_l\tilde{b}|}{|a/b|} - 1\right| \le \epsilon_a + \epsilon_b + \epsilon(l) + 4.01\epsilon^2.$$

Using these formulas we have estimated the rounding errors in our algorithms, so that we can give a priori error bounds for all functions calculated in the following paragraphs.

With these preparations we are now able to formulate algorithms for the verified inclusion of the constants π, $\ln 2$ and the functions $\sqrt{\ }$, ln, arctan.

4 The square root and π

First, we will consider the algorithm for the square root.

We denote all numbers and functions of the type *long_ interval* by capital letters and the double-precision numbers by small letters. The type conversion is immediately made by the assignment. With a diameter of $2 \cdot 2^{-50}$ the start interval is adjusted to the double-precision format. The multiplication by powers of two should be realized by manipulations of the exponent in the internal representation of the numbers.

To calculate the square root we use the following interval version of the *Newton*-iteration. This iteration is based on the formula $x_{n+1} = x_n - 0.5 * (x_n^2 - x)/x_n$ and not on the classical iteration $x_{n+1} = 0.5 * (x_n + x/x_n)$. The iteration used in our implementation needs some more operations but is of higher precision.

$\mathrm{Sqrt}\,(X)$:

If $X.\inf < 0$ *then Error else*

$\{st := \lfloor \mathrm{ld}\, p \rfloor + 1;$ $(*$ *p binary digits* $*)$

$X_0 := \left[\mathrm{sqrt}\left(x.\inf *(1 - 2^{-50})\right), \mathrm{sqrt}\left(x.\sup *(1 + 2^{-50})\right)\right]$;

while $st > 0$ *do*

$$\left\{X_0 := \left(\mathrm{mid}\ X_0 - \frac{\mathrm{Sqr}\,(\mathrm{mid}\ X_0) - X}{2X_0}\right) \cap X_0\ ; \quad \mathrm{dec}\,(st)\right\};$$

$\mathrm{Sqrt}\,(X) := X_0\}$.

It is also possible to use another stopping criterion based on the difference between the midpoints of two consecutive approximation intervals. Please note that if the argument X of the above algorithm leaves the range of the double precision format, the calculation of the startvalue X_0 does not work. In this case we split the argument into a power of $\mathcal{B}$ (e.g. $\mathcal{B} = 2$) with an even exponent and a factor within the double precision range to calculate the startvalue X_0 described as above.

Now we estimate the rounding error of the interval *Newton* square root algorithm for an argument x belonging to S.

Let ϵ_n, satisfying $10^{-4} > |\epsilon_n| > 0$, be the approximation error and $2 \cdot \delta(l)$ with $|\delta(l)| < 10^{-4}$ representing the interval diameter. Then we put $Y_n = \sqrt{x}\big(1+\epsilon_n \pm \delta(l)\big)$, mid $Y_n = \sqrt{x}(1+\epsilon_n)$, and perform the next step of the *Newton* iteration. Using formulas of paragraph 3 we get the new interval

$$Y_{n+1} = \sqrt{x}\big(1 + 0.51\epsilon_n^2 \pm w\epsilon(l) \pm 1.01\epsilon_n\delta(l)\big), \quad w < 1.6.$$

The asymptotic approximation error is of order $0.5\epsilon_n^2$, the rounding error $1.5\epsilon(l)$. Thus, two guard digits are sufficient in most cases.

If we inject a small interval $X = (1 \pm \rho)$ mid X instead of the argument x, we have an additional error term 0.51ρ. However we prefer to evaluate a square root with an interval argument in the form $\sqrt{[x_l, x_u]} \subseteq [(\sqrt{x_l})_l, (\sqrt{x_u})_u]$. In the case of the classical *Newton* iteration we get $w < 2.6$.

With the a priori error estimation for the square root algorithm we are now able to discuss the error accumulation for the arithmetic-geometric mean iteration (cf. Lemma 2). With a starting argument $X = x\big(1 \pm \epsilon(l)\big)$ for the AGM iteration after n_1 steps we find a relative error of order $(2.1 \cdot n_1 + 1)\epsilon(l)$ for a_{n_1} and b_{n_1}, respectively. Assuming overlapping limit intervals we finally deduce a relative error bound (cf. [13])

$$\Delta_{AGM} \leq (4.2 \cdot n_1 + 2)\epsilon(l), \quad \Xi \subseteq \xi(1 \pm \Delta_{AGM}).$$

In some further algorithms for standard functions we need the constant Π. So we first formulate an algorithm for verified inclusion of Π in multiple-precision arithmetic.

Using the formulas (4) and (5) together with the *Legendre* relation (1) and $k = 1/\sqrt{2}$ [5, p. 246]) we are able to construct the following algorithm:

Π :
$st := \lfloor \mathrm{ld}\, p \rfloor + 1$; (* *p binary digits* *)
$A := 1$; $B := 1/\mathrm{Sqrt}\,(2)$; $T := 0$; $x := 1$;
while $st > 0$ *do*
{$Y := A$; $A := (A+B)/2$; $B := \mathrm{Sqrt}\,(B * Y)$;
$T := T - x * \mathrm{Sqr}\,(A - Y)$; $x := 2 * x$; $\mathrm{dec}\,(st)$};

$$T := T - x * \left(\frac{A-B}{2}\right)^2 + \frac{1}{4} - \left[0, \sup\left(4 * x * \left(\frac{A-B}{4}\right)^4\right)\right] \ ;$$

$$\Pi := [B^2.\inf,\ A^2.\sup]/T\ .$$

Notes:

1) With $t_i := 1/4 - \sum_{j=1}^{i} 2^{j-1}c_j^2$ it holds:

$$\begin{aligned} 0 \ \le\ t_i - t_\infty \ &=\ \sum_{j=i+1}^{\infty} 2^{j-1}(a_j - a_{j-1})^2 \ \le\ 2^{i+1}\left(\frac{a_i - b_i}{2}\right)^2 \\ &\le\ 2^{i+1}\frac{(a_{i-1} - b_{i-1})^4}{256 \cdot a_{i+1}^2} \ \le\ 4 \cdot 2^i \left(\frac{a_{i-1} - b_{i-1}}{4}\right)^4 . \end{aligned}$$

2) It is also practicable to use $\Pi = [(A_{i+1}^2/T_i).\inf,\ (A_i^2/T_i).\sup]$, [5, p. 246].

To estimate the relative error we have to discuss the utilized terms A^2, B^2, T. With n_1 the number of steps of the AGM iteration and the assumptions $2^{n_1+2}(2.1 \cdot n_1 + 1)^2\epsilon(l) < 1$ we get $A^2 \subseteq a^2(1 \pm \Delta_{A^2})$, $B^2 \subseteq b^2(1 \pm \Delta_{B^2})$, $T \subseteq t(1 \pm \Delta_T)$ with the relative error bounds (cf. [13]):

$$\Delta_{A^2,B^2} \le (4.2 \cdot n_1 + 4)\epsilon(l), \quad \Delta_T \le (0.1 \cdot n_1 + 13.33)\epsilon(l).$$

If we now execute the last divison, we finally obtain:

$$\Delta_\Pi \le (8.6 \cdot n_1 + 22.6)\epsilon(l), \qquad \Pi \subseteq \pi(1 \pm \Delta_\Pi).$$

Because of the quadratic convergence of the AGM-method as a number of iteration steps the choice of $n_1 = \lfloor \operatorname{ld}(l/\log_{\mathcal{B}} 2)\rfloor + 1$ is sufficient.

5 The natural logarithm and $\operatorname{Ln} 2$

We will now discuss the function $\operatorname{Ln} x$. To compute inclusions for the logarithm we need the constants Π and $\operatorname{Ln} 2$ as well as the function $K'(4/X)$. The calculation of $\operatorname{Ln} 2$ is possible with an algorithm similar to the one given below. If we assume that $\operatorname{Ln} 2$ is precalculated, we can give the algorithm for $\operatorname{Ln} X$, using the following theorem.

Theorem 1. *Let $p \ge 30$ be the number of valid binary digits and $m := \lfloor p/2 - \ln y/\ln 2 + \ln p + 3\rfloor$. Then with $k = 2^{2-m}/y$ it follows: $|\ln(y \cdot 2^m) - K'(k)| \le 2^{-p}$.*

Proof. Using $4k^2(8+|\ln k|) \le 8\cdot k^2|\ln k|$ and by virtue of Lemma 4 we find

$$|\ln(y\cdot 2^m) - K'(k)| \le 2^{-p}2^{7-2\ln p-4}(p/2+\ln p+1)\ln 2 \le 2^{-p}.$$

The choice of $p \ge 30$ in the assumptions of Theorems 1 and 2 corresponds to nine decimal places and so the results include the classical real and double types.

$\mathrm{Ln}\,X$: *p binary digits*, $\mathrm{Ln}\,2$ *known*
If $X.\inf \le 0$ *then Error* ;
If $X.\sup \le 1$ *then* $\mathrm{Ln}\,X = -\mathrm{Ln}\,(1/X)$;
If $X.\inf < 1 \ \wedge\ X.\sup > 1$ *then* $\mathrm{Ln}\,X = \mathrm{Ln}\,[X.\inf, 1] \cup \mathrm{Ln}\,[1, X.\sup]$;
If $X.\inf < 1 + 2^{-p/(2\mu)} \ \wedge\ X.\sup > 1 + 2^{-p/(2\mu)}$ *then*
$\mathrm{Ln}\,X = \mathrm{Ln}\,[X.\inf, 1 + 2^{-p/(2\mu)}] \cup \mathrm{Ln}\,[1 + 2^{-p/(2\mu)}, X.\sup]$;
If $X = 1 + Y \ \wedge\ Y.\inf \ge 0 \ \wedge\ Y.\sup < 2^{-p/(2\mu)}$ *then*

$$\mathrm{Ln}\,X = \frac{2Y}{2+Y}\sum_{k=0}^{\mu-1}\frac{2}{2k+1}\left(\frac{Y}{2+Y}\right)^{2k+1} + [0, Y.sup * 2^{-p-2\mu}] \ ;$$

If $X.\inf \ge 1 + 2^{-p/(2\mu)}$ *then*
$\{st := \lfloor 2 * \mathrm{ld}\,p \rfloor$; $(*$ p = *number of binary digits* $*)$

$$m := \left\lfloor \frac{p}{2} - \frac{\ln x.\inf}{\ln 2} + \ln p + 3 \right\rfloor \ ;$$

$X := X * 2^m$; $\quad A := 1$; $\quad B := 4/X$;
while $st > 0$ *do*
$\{Y := A$; $\quad A := (A+B)/2$; $\quad B := \mathrm{Sqrt}\,(B * Y)$; $\quad \mathrm{dec}\,(st)\}$;

$$\mathrm{Ln}\,X := \left[\left(\frac{\Pi}{2 * [B.\inf, A.\sup]}\right).\inf -2^{-p},\right.$$
$$\left.\left(\frac{\Pi}{2 * [B.\inf, A.\sup]}\right).\sup +2^{-p}\right] - m\cdot \mathrm{Ln}\,2\}.$$

In the last case ($2^{-p/(2\mu)} < Y.sup < 1/2$) it is necessary to increase the mantissa length by $(50/\mu)\%$ to get results of the same quality as in the other cases (cp. (7)).

To compute $\operatorname{Ln} 2$ it is preferable to use an argument 2^m which belongs to the screen S. As a relative error bound we get in this case (cf. [13]):

$$\Delta_{\operatorname{Ln} 2} \leq (8.8 \cdot n_1 + 4.3 \cdot n_2 + 29.2)\epsilon(l), \qquad \operatorname{Ln} 2 \subseteq \ln 2(1 \pm \Delta_{\operatorname{Ln} 2})$$

where $n_2 = \lfloor 2 \operatorname{ld}(l/\log_{\mathcal{B}} 2) \rfloor$ is the number of iteration steps necessary to calculate the AGM sequences and n_1 the number of steps in the calculation of Π.

For the function $\operatorname{Ln} x$ we only consider the case x point-interval, $x \geq 1$, and have to take into account the absolute errors stemming from the computation of $\operatorname{Ln}(2^m \cdot x)$ as well as $\operatorname{Ln} 2^m$. Additionally, for an argument x nearby 1 a cancellation occurs in the difference $\ln(2^m \cdot x) - m \ln 2$ depending on the number of vanishing digits of $x - 1$ after the radix point and the mantissa length l. Roughly speaking $const \cdot \left([\log_{\mathcal{B}} 1/(x-1)] + 1\right)$ guard digits are sufficient. More precisely, we find (cf. [13]):

$$\begin{aligned} \Delta_{\operatorname{Ln} x} &\leq \left(1 + 1.01\left(1 + \frac{2m \ln 2}{\ln x}\right)(8.8 \cdot n_1 + 4.3 \cdot n_2 + 30.2)\right)\epsilon(l)\ , \\ m &= 0.62 \cdot l \cdot \operatorname{ld} \mathcal{B} + 3 \qquad (7) \\ \operatorname{Ln} x &\subseteq \ln x(1 \pm \Delta_{\operatorname{Ln} x}). \end{aligned}$$

As described in the square root case it is also possible to calculate the logarithm of an interval argument by taking advantage of its monotonicity.

6 The inverse tangent $\operatorname{Arctan} X$

Now we will develop an algorithm for the calculation of the inverse tangent function. First we note down a corollary of the *Landen*-transformation (Lemma 3). It holds:

$$K(\sin \alpha_i, \psi_i) = \prod_{j=0}^{i-1} \frac{1 + s_j}{2} K(\sin \alpha_0, \psi_0). \qquad (8)$$

With $a_0 = 1$, $b_0 = k$, $s_i = b_i/a_i$ the sequence $\{a_i\}$ decreases strictly to the limit

$$\lim_{i \to \infty} a_i = a_\infty = \prod_{i=0}^{\infty} \frac{1 + s_i}{2} = \frac{\pi}{2K'(k)}, \quad s_{i+1} = \frac{2\sqrt{s_i}}{1 + s_i}.$$

Hence the estimation $a_\infty \le a_i \le a_\infty + a_i - b_i \le a_\infty + (1 - s_i)$ follows. To develop an a priori estimation of ψ_i we assume that $\psi_0 \le \pi/4$, $s_0 = 2^{-p/2}$, $1 - s_i \le 1/(100 \cdot p)$ is valid. Then (3) shows:

$$s_i \ln \tan\left(\frac{\pi}{4} + \frac{\psi_i}{2}\right) \le \left(a_\infty + (1 - s_i)\right) \psi_0 \left(1 + \frac{2^{-p}}{3}\right).$$

From Lemma 4 follows

$$a_\infty = \frac{\pi}{2 \cdot \left(\ln(2^{2+p/2}) + \theta\, 2^{2-p}(8 + \ln 2^{p/2})\right)}, \quad -1 < \theta < 1$$

so that we finally get

$$\frac{\pi}{(p + 4.001)\ln 2}\,\psi_0 \;\le\; \ln \tan\left(\frac{\pi}{4} + \frac{\psi_i}{2}\right) \;\le\; \frac{\pi}{p \ln 2}\,\psi_0 + (1 - s_i), \quad p \ge 30.$$

With the note following Lemma 3, $x = \tan \psi_0$ and $0 < \theta_\nu < 1$, we obtain:

$$\begin{aligned}\arctan x \left(1 + \theta_1 \frac{2^{-p}}{3}\right) &= \ln \frac{1 + v_i}{1 - v_i} \cdot \left(1 - \theta_2 (1 - s_i) \tan^2 \psi_i\right) \prod_{j=0}^{i-1} \frac{2}{1 + s_j} \\ &= \ln \frac{1 + v_i}{1 - v_i} \prod_{j=0}^{i-1} \frac{2}{1 + s_j} - \theta_3 (1 - s_i) \tan^2 \psi_i \ln \frac{1 + v_i}{1 - v_i} \cdot \frac{2}{\pi} (\ln 2^{p/2+2} + \ln 2)\end{aligned}$$

so that an application of the a priori estimation of ψ_i leads to the following theorem.

Theorem 2. *With $p \ge 30$, $1 - s_i \le 1/(100 \cdot p)$, $s_0 = 2^{-p/2}$, $0 \le x = \tan \psi_0 \le 1$ and s_i, v_i defined as in the note after Lemma 3 the following estimation is valid:*

$$\ln \frac{1 + v_i}{1 - v_i} \prod_{j=0}^{i-1} \frac{2}{1 + s_j} \;\ge\; \arctan x \;\ge\; \ln \frac{1 + v_i}{1 - v_i} \prod_{j=0}^{i-1} \frac{2}{1 + s_j} - \frac{2^{-p}}{3} - (1 - s_i)\frac{13}{p^2}.$$

Arctan (X): *p binary digits*
If $X.\sup \le 0$ *then* $\text{Arctan}\,(X) = -\text{Arctan}\,(-X)$;
If $X.\inf \le 0 \;\wedge\; X.\sup \ge 0$ *then* $\text{Arctan}\,(X)$
$\quad = \text{Arctan}\,[X.\inf, 0] \cup \text{Arctan}\,[0, X.\sup]$;

$$
\begin{aligned}
&\textit{If } X.\inf \geq 1 \textit{ then } \operatorname{Arctan}(X) = \Pi/2 - \operatorname{Arctan}(1/X)\ ;\\
&\textit{If } X.\inf < 1 \ \wedge\ X.\sup > 1 \textit{ then } \operatorname{Arctan}(X)\\
&\qquad = \operatorname{Arctan}[X.\inf, 1] \cup \operatorname{Arctan}[1, X.\sup]\ ;\\
&\textit{If } X.\inf \geq 0 \ \wedge\ X.\sup \leq 1 \textit{ then}\\
&\Big\{ \textit{If } X.\inf < 0.5 \ \wedge\ X.\sup > 0.5 \textit{ then}\\
&\operatorname{Arctan}(X) = \operatorname{Arctan}[X.\inf, 0.5] \cup \operatorname{Arctan}[0.5, X.\sup] \textit{ else}\\
&\quad \textit{If } X.\sup < 0.5 \textit{ then } \operatorname{Arctan}(X) = \Pi/4 - \operatorname{Arctan}(1-X)/(1+X) \textit{ else}\\
&\quad \big\{\ st := \lfloor 2 * \operatorname{ld} p \rfloor + 1\ ;\ (*\ p \textit{ binary digits } *)\\
&\quad\ S := 2^{-p/2}\ ;\quad V := X/\Big(1 + \operatorname{Sqrt}\big(1 + \operatorname{Sqr}(X)\big)\Big)\ ;\quad Q := 1\ ;\\
&\quad\ \textit{while } st > 0 \textit{ do}\\
&\quad\ \big\{\ Q := 2 * Q/(1+S)\ ;\quad W := 2 * S * V/\big(1 + \operatorname{Sqr}(V)\big)\ ;\\
&\quad\ \textit{If } W.\sup > 1 \textit{ then } W.\sup := 1\ ;\\
&\quad\ W := W/\Big(1 + \operatorname{Sqrt}\big(1 - \operatorname{Sqr}(W)\big)\Big)\ ;\\
&\quad\ W := (V+W)/(1 - V * W)\ ;\\
&\quad\ V := W/\Big(1 + \operatorname{Sqrt}\big(1 + \operatorname{Sqr}(W)\big)\Big)\ ;\\
&\quad\ \textit{If } V.\inf < 0 \textit{ then } V.\inf := 0\ ;\\
&\quad\ S := 2 * \operatorname{Sqrt}(S)/(1+S)\ ;\\
&\quad\ \textit{If } S.\sup > 1 \textit{ then } S.\sup := 1\ ;\quad \operatorname{dec}(st)\big\}\Big\};\\
&\operatorname{Arctan}(X) := Q * \operatorname{Ln}\frac{1+V}{1-V} - \big[0, 2^{-p-1} + \big(13 * (1-S)/\operatorname{Sqr}(p)\big).\sup\big]\Big\}.
\end{aligned}
$$

Using the algorithm described above we have a loss of precision because the argument of the logarithm is close to 1. In this case the asymptotic relation $(1+v_i)/(1-v_i) = 1 + O(1/p)$ shows that we lose $\operatorname{ld} p$ binary digits.

For the derivation of a relative a priori error bound in the sequel we will consider the final error terms of s, q, v, $\ln(1+v)/(1-v)$ and assume

$$\epsilon(l) \leq 10^{-4}, \quad \delta(l)^2 \leq 10^{-4}\epsilon(l), \quad (5n_3+1)^2\epsilon(l) < 10^{-2}.$$

Looking at the transformation for the sequence $\{s_j\}$, we get the relative errors (cf. [13]):

$$\Delta_{S_1} = 2^{-p/2}\epsilon(l), \quad \Delta_{S_j} = s_j\big(1 + 4.7 \cdot (j-1)\big)\epsilon(l), \quad S_j \subseteq s_j(1 \pm \Delta_{S_j}).$$

The sequence $\{s_j\}$ increases to 1 with quadratic convergence, so that $n_3 = \lfloor 2\,\mathrm{ld}\,(l/\log_{\mathcal{B}} 2)\rfloor + 1$ steps are sufficient. Then after n_3 steps we obtain for Q the relative error

$$\Delta_Q \le 1.2 \cdot n_3 \cdot (n_3+2)\epsilon(l), \quad Q \subseteq q(1 \pm \Delta_Q).$$

Simultaneously, we have to calculate the value V. The following lemma proved in [13] shows the error estimations for the terms necessary.

Lemma 5.

i) Let $X = x\big(1 \pm \delta(l)\big)$, $0 < x \le a$, $a \le 0.99$ *in the "-" case. Then it follows*

$$\begin{aligned} X/_l \left(1 +_l \sqrt{1 \pm_l X^2}\right) & \\ \subseteq \; & \frac{x}{1+\sqrt{1 \pm x^2}}\Bigg(1 \pm 2.03\epsilon(l) \pm 1.01\delta(l) \pm 1.01 c_{sgn} \times \\ & \times \left(\frac{a^2}{1 \pm a^2}\{1.005\delta(l) + 0.505\epsilon(l)\} + 2.11\epsilon(l)\right)\Bigg). \\ c_+ \; := \; & \frac{\sqrt{1+a^2}}{1+\sqrt{1+a^2}}, \\ c_- \; := \; & 0.5. \end{aligned}$$

ii) Let $0 < v \le a \le 1$, $V = v\big(1 \pm \delta(l)\big)$, $0 < s \le 1$, $S = s\big(1 \pm \rho(l)\big)$. *Then it holds*

$$\begin{aligned} & 2 \cdot_l S \cdot_l V/_l(1 +_l V^2) \subseteq (2sv)/(1+v^2) \cdot \Bigg\{1 \pm 1.01 \times \\ & \times \left(\rho(l) + \epsilon(l) + 1.01\left[\left(1 + \frac{2.01a^2}{1+a^2}\right)\delta(l) + \left(\frac{1.01a^2}{1+a^2} + 3.01\right)\epsilon(l)\right]\right)\Bigg\}. \end{aligned}$$

iii) From $0 < v \le a \le 1/\sqrt{2}$, $0 < w \le b \le 1/\sqrt{2}$, $V = v\big(1 \pm \delta(l)\big)$, $W = w\big(1 \pm \rho(l)\big)$, *it follows*

$$(V +_l W)/_l(1 -_l V \cdot_l W) \subseteq (v+w)/(1-vw) \cdot \Bigg\{1 \pm 1.01 \times$$

$$\times \left[\left(3.02 + \frac{1.01ab}{1-ab} \right) \epsilon(l) + \left(\frac{1.01ab}{1-ab} + \frac{v}{v+w} \right) \delta(l) + \right.$$
$$\left. + \left(\frac{1.01ab}{1-ab} + \frac{w}{v+w} \right) \rho(l) \right] \Bigg\}.$$

Using the results from the above lemma we estimate the relative error of the term v. We consider the first two loops and then the general step $v \to w_1 \to w_2 \to w_3 \to v$ of the algorithm. We get the following results.

First step:	Second step:
$V_0 \subseteq v_0\Big(1 \pm 4.74\epsilon(l)\Big)$,	$V_1 \subseteq v_1\Big(1 \pm 11.8\epsilon(l)\Big)$, $v_1 \le 0.207116$,
$W_1 \subseteq w_1\Big(1 \pm 11.5\epsilon(l)\Big)$,	$W_1 \subseteq w_1\Big(1 \pm 22.92\epsilon(l)\Big)$, $w_1 \le 0.0046$,
$W_2 \subseteq w_2\Big(1 \pm 14.72\epsilon(l)\Big)$,	$W_2 \subseteq w_2\Big(1 \pm 26.25\epsilon(l)\Big)$, $w_2 \le 0.0023$,
$W_3 \subseteq w_3\Big(1 \pm 7.84\epsilon(l)\Big)$.	$W_3 \subseteq w_3\Big(1 \pm 15.6\epsilon(l)\Big)$, $w_3 \le 0.21$,
	$V_2 \subseteq v_2\Big(1 \pm 19.22\epsilon(l)\Big)$, $v_2 \le 0.105$.

General loop:

$$V_j \subseteq v_j\Big(1 \pm \delta(l)\Big),\ v_j \le 0.105,\ S_j \subseteq s_j(1 \pm \Big(1 + 4.7(j-1)\Big)\epsilon(l)),\ s_j \le 1,$$
$$W_1 \subseteq w_1\left(1 \pm 1.043\delta(l) \pm \Big(5.11 + 4.747(j-1)\Big)\epsilon(l)\right),\ w_1 \le 0.21,$$
$$W_2 \subseteq w_2\left(1 \pm 1.078\delta(l) \pm \Big(8.39 + 4.91(j-1)\Big)\epsilon(l)\right),\ w_2 \le v_j,\ \frac{w_2}{v_j + w_2} \le 0.5,$$
$$W_3 \subseteq w_3\left(1 \pm 1.0731\delta(l) \pm \Big(7.4 + 2.54(j-1)\Big)\epsilon(l)\right),\ w_3 \le 0.213,$$
$$V_{j+1} \subseteq v_{j+1}\left(1 \pm 1.11\delta(l) \pm \Big(10.76 + 2.63(j-1)\Big)\epsilon(l)\right),\ j \ge 2.$$

Starting from $X = x\big(1 \pm \epsilon(l)\big)$ and $V_0 = v_0\big(1 \pm 4.74\epsilon(l)\big)$, we have an estimate after n_3 steps:

$$\Delta_V \le (319.92 \cdot 1.11^{n_3} - 23.9n_3 - 315.17)\epsilon(l), \quad V \subseteq v(1 \pm \Delta_V) \qquad (9)$$

or $\Delta_V \le (356 \cdot p^{0.302} - 47.8 \operatorname{ld} p - 315) \cdot \epsilon(l)$ when $\mathcal{B} = 2$ and $n_3 = \lfloor 2 \operatorname{ld} p \rfloor + 1$. The inequality

$$\frac{\pi}{(p+5) \cdot 2 \ln 2} \psi_0 \le v \le \frac{\pi}{p \cdot 2 \ln 2} \psi_0, \quad p \ge 30$$

yields

$$\Delta_{\frac{1+V}{1-V}} \leq 3.03\epsilon(l) + \frac{3.6}{p}\Delta_V.$$

Now we have to evaluate the relative error of the logarithm $\mathrm{Ln}\,\frac{1+V}{1-V}$ by (7). This leads to a further factor which predominates the error in (9) in most cases

$$1+\big(1+\{1.01+0.4(1.24\cdot l\cdot \mathrm{ld}\,\mathcal{B}+6)\cdot(l\cdot \mathrm{ld}\,\mathcal{B}+4)\}(8.8\cdot n_1+4.3\cdot n_2+30.2)\big)\epsilon(l). \tag{10}$$

Numerical example. Now we give a numerical example for our Arctan-routine. We have calculated the value $\mathrm{Arctan}\,(0.5)$ using the *Turbo-Pascal* extension *TPX* [8] of the Institut Informatik III at TU Hamburg-Harburg (Prof. *Rump*). The calculations are executed with a mantissa length of 320 binary digits.

$$\begin{aligned}
\mathrm{Arctan}\,(0.5) &= 0.463647^{6099}_{5926} \\
&= 0.463647609000^{8224}_{5076} \\
&= 0.4636476090008061162 14^{2617}_{1572} \\
&= 0.463647609000806116214256231461214402028 5^{3765}_{2616} \\
&= 0.46364760900080611621425623146121440202 85 \\
&\qquad 37054286120263810933088720197864165741^{7125}_{5737}
\end{aligned}$$

In the next iteration step we get 7 more valid digits. Using the formulas derived above we get a relative error bound of $\Delta = 10^7 \cdot 2^{-304}$. All algorithms are also implemented and tested with *PASCAL–XSC* [11] and *INTPAK* for *MAPLE* [6] with ϵ-inflated arguments for point intervals. The results are similar to the results of *TPX*.

7 Further functions

The given algorithms together with the self-correcting interval *Newton*-method allow us to calculate the functions Tan , Exp with a cost of $O\big(\mathrm{ld}\,pM(p)\big)$.

As a higher function we will now discuss the inverse *Weierstraß*-function and show that it is possible to calculate inclusions for this function with

a quadratic convergent algorithm ([3]). The inverse *Weierstraß*-function is defined as

$$I(u,e_1,e_2,e_3) := \int_u^\infty \frac{dx}{\sqrt{(x-e_1)(x-e_2)(x-e_3)}}\,, \qquad e_3 < e_2 < e_1, \quad e_1+e_2+e_3 \;=\; 0. \tag{11}$$

We now define

$$a = a_0 := \sqrt{e_1-e_3}\,, \quad b = b_0 := \sqrt{e_1-e_2}\,, \quad c = c_0 := \sqrt{e_2-e_3}\,, \quad u_0 = u,$$

$$a_{j+1} \;:=\; \frac{a_j+b_j}{2}, \quad b_{j+1} \;:=\; \sqrt{a_j\cdot b_j},$$

$$e_1^{(j)} \;:=\; \frac{a_j^2+b_j^2}{3}, \quad e_2^{(j)} \;:=\; \frac{a_j^2-2b_j^2}{3}, \quad e_3^{(j)} \;:=\; \frac{b_j^2-2a_j^2}{3}.$$

With

$$u_{j-1} = u_j + \frac{\left(e_1^{(j)}-e_3^{(j)}\right)\left(e_2^{(j)}-e_3^{(j)}\right)}{u_j-e_3^{(j)}}$$

we get

$$u_j = \frac{u_{j-1}+e_3^{(j)}}{2} + \sqrt{\frac{\left(u_{j-1}-e_3^{(j)}\right)^2}{4} - a_j^2(a_j^2-b_j^2)}$$

and for $j\to\infty$ holds:

$$a_j \downarrow AGM(a,b)\,,\; b_j \uparrow AGM(a,b)\,,$$

$$e_1^{(j)} \to \frac{2AGM^2(a,b)}{3}\,,\; e_2^{(j)} \downarrow -\frac{AGM^2(a,b)}{3}\,,\; e_3^{(j)} \uparrow -\frac{AGM^2(a,b)}{3}\,.$$

With the substitution $x = \phi^2 + 2AGM^2(a,b)/3$ we finally find

$$\begin{aligned} I(u,e_1,e_2,e_3) \;&=\; \int_{u_\infty}^\infty \frac{dx}{\sqrt{\left(x-e_1^{(\infty)}\right)\left(x-e_2^{(\infty)}\right)\left(x-e_3^{(\infty)}\right)}} \\ &=\; 2\int_{\sqrt{u_\infty-2AGM^2/3}}^\infty \frac{d\phi}{\phi^2+AGM^2} \\ &=\; \frac{2}{AGM}\left(\frac{\pi}{2} - \arctan\frac{\sqrt{u_\infty-2AGM^2/3}}{AGM}\right). \end{aligned}$$

In the special case of $u = e_1$ we have

$$I(e_1, e_1, e_2, e_3) = \frac{\pi}{AGM(a,b)}.$$

Therefore, the following algorithm is valid.

$I(u, e_1, e_2, -e_1 - e_2) :\ e_2 < e_1 \le u,\ e_1 > 0,\ p$ *binary digits*

$U := u\ ; \quad E_1 := e_1\ ; \quad E_2 := e_2\ ; \quad E_3 := -E_1 - E_2\ ;$

If $U.\inf < E_1.\sup \vee E_1.\inf \le E_2.\sup \vee E_2.\inf \le E_3.\sup$ *then Error else*

$\{\ A := \mathrm{Sqrt}\,(E_1 - E_3)\ ; \quad B := \mathrm{Sqrt}\,(E_1 - E_2)\ ;$

$st := \lfloor \mathrm{ld}\, p \rfloor + 1 + \lfloor \mathrm{ld}\, k \rfloor\ ;$ $(*\ p$ *binary digits precision* $*)$

while $st > 0$ *do* $\qquad (*\ k$ *number of digits of* $Int\big((e_1 - e_3)/(e_1 - e_2)\big)\ *)$

$\{\ T := A\ ; \quad A := (A + B)/2\ ; \quad B := \mathrm{Sqrt}\,(B * T)\ ;$

$A2 := A * A\ ; \quad B2 := B * B\ ; \quad E3 := (B2 - 2 * A2)/3\ ;$

$U := (U + E3)/2 + \mathrm{Sqrt}\,\big(\mathrm{Sqr}\,(U - E3)/4 - A2 * (A2 - B2)\big)\ ;$

$\mathrm{dec}\,(st)\};$

$T1 := U - 2 * \mathrm{Sqr}\,(A)/3\ ; \quad$ *If* $T1.\inf < 0$ *then* $T1.\inf := 0\ ;$

$T2 := U - 2 * \mathrm{Sqr}\,(B)/3\ ; \quad$ *If* $T2.\inf < 0$ *then* $T2.\inf := 0\ ;$

$$I(u, e_1, e_2, -e_1 - e_2) \in$$

$$\left[\frac{2}{A.\sup}\left\{\frac{\Pi}{2} - \mathrm{Arctan}\,\left(\frac{\mathrm{Sqrt}\,T2}{B}.\sup\right)\right\}.\inf,\right.$$

$$\left.\frac{2}{B.\inf}\left\{\frac{\Pi}{2} - \mathrm{Arctan}\,\left(\frac{\mathrm{Sqrt}\,T1}{A}.\inf\right)\right\}.\sup\right]\Bigg\}$$

If the parameter u is near e_1 it is possible that up to $p/2$ binary digits are inaccurate.

Numerical example.

$$I(4, 2, 1, -3) \ = \ 1.0439152198309301320850224250197643005923227766097547853947331705225497501115586355663^{3223}_{2516}$$

The inclusion needs 7 iteration steps with a precision of 320 binary digits.

8 Execution time

In the above chapters we have given algorithms for elementary functions in high precision arithmetic based on AGM-iteration. Now we will discuss the execution time of our implementations in comparison with the one of the implementation in the PASCAL–XSC modules *mp_ari* and *mpi_ari* [11] which works with a priori estimations and *Taylor* series expansions [4, 9].

Therefore we have implemented all described algorithms using PASCAL–XSC and the multiple-precision arithmetic of the modules *mp_ari* and *mpi_ari*. Here we have used the algorithms with multiple-precision point arithmetic and the enclosures for point arguments are computed using outward rounding of the results and a sufficiently large number of guard digits. The quantity of these guard digits is given by the a priori estimations (7), (9) and (10). For interval arguments we utilize the monotonicity of the functions in connection with point evaluations for the lower and upper bounds of the interval argument. In this case the algorithms are equal to the given PASCAL pseudo code where all interval operations are changed to point operations and the last inclusions are omitted.

To compare the execution time T_{AGM} of our implementation with the time T_{PXSC} of the PASCAL–XSC functions in *mpi_ari* we have calculated for example the values arctan(0.5) and ln(3.0) with various precisions. The needed constants Π and $\operatorname{Ln} 2$ were precalculated. We get the following results.

	ln(3.0)		arctan(0.5)	
precision	T_{PXSC}/T_{AGM}	guard digits	T_{PXSC}/T_{AGM}	guard digits
1600 bits	2.6	51	1.3	72
6400 bits	5.2	53	4.0	77
9600 bits	6.5	53	6.1	78

Both implementations of the square root lead to the same evaluation time. For the precision of a quad data-type (128 bits) we find that our implementation of logarithms needs the same execution time as the PASCAL–XSC function and the inverse tangent needs approximately twice as much time.

The precision of the results was the same as in the PASCAL–XSC implementation. So we see that the advantage of quadratic convergent AGM procedures grows with increasing precision.

9 Conclusion

We have presented quadratic convergent algorithms for the verified inclusion of elementary functions like arctan, ln and higher functions like the *Weierstraß* function. The quadratic convergence provides faster procedures than *Taylor* series expansions if we need high precision. Another advantage is that it is not necessary to reduce the function arguments to a specific interval. The presented algorithms in connection with a priori error estimates lead to inclusions by using only multiple-precision point arithmetic with a sufficiently large number of guard digits.

References

[1] Borwein, J. M. and Borwein, P. B. *The arithmetic-geometric mean and fast computation of elementary functions.* SIAM Review **26** (1984), pp. 351–366.

[2] Borwein, J. M. and Borwein, P. B. *π and the A.G.M.* Wiley, 1987.

[3] Bost, J. B. and Mestre, J. F. *Moyenne arithmético-géométrique et périodes de courbes de genre 1 et 2.* Gazette des Mathématiciens **38** (1988), pp. 36–64.

[4] Braune, K. *Standard functions for real and complex point and interval arguments with dynamic accuracy.* Computing Suppl. **6** (1988), pp. 227–244.

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

[6] Connell, A. E. and Corless, R. M. *An experimental interval arithmetic package in Maple.* INTPAK description from MAPLE share library, ETH Zürich.

[7] Gal, S. and Bachelis, B. *An accurate elementary mathematical library for the IEEE Floating Point Standard.* ACM Transactions on Math. Software **17** (1991), pp. 26–45.

[8] Husung, D. *TPX Version 1.1US.* TU Hamburg-Harburg, 1992.

[9] Krämer, W. *Inverse standard functions for real and complex point and interval arguments with dynamic accuracy.* Computing Supplement **6** (1988), pp. 185–212.

[10] Krämer, W. *Computation of interval bounds for elliptic integrals.* In: Atanassova, L. and Herzberger, J. (eds) "Computer Arithmetic and Enclosure Methods", North-Holland, 1992, pp. 289–298.

[11] Krämer, W. *Eine portable Langzahl- und Langzahlintervallarithmetik mit Anwendungen.* ZAMM **73** (7/8) (1993), T849–853.

[12] Lawden, D. F. *Elliptic functions and applications.* Springer, 1989.

[13] Luther, W. and Otten, W. *Computation of standard interval functions in multiple-precision interval arithmetic.* Schriftenreihe des Fachbereichs Mathematik der Universität-GH Duisburg, SM–DU–233, 1993.

Received: November 11, 1993
Revised version: July 13, 1994

Universität–GH Duisburg
Informatik II
Lotharstraße 65
D–47048 Duisburg
Germany
e-mail: `werner@marvin.uni-duisburg.de`