Journal of Universal Computer Science, vol. 4, no. 1 (1998), 25-33
submitted: 25/9/97, accepted: 1/11/97, appeared: 28/1/98 © Springer Pub. Co.

Reliable Computation of Elliptic Functions

Wolfram Luther
(Gerhard–Mercator–Universität –GH Duisburg, Fachbereich Mathematik
Luther@informatik.uni-duisburg.de)

Werner Otten
(Gerhard–Mercator–Universität –GH Duisburg, Fachbereich Mathematik
Otten@informatik.uni-duisburg.de)

Abstract: In this note we present rapidly convergent algorithms depending on the method of arithmetic-geometric means (AGM) for the computation of Jacobian elliptic functions and Jacobi's Theta-function. In particular, we derive explicit a priori bounds for the error accumulation of the corresponding Landen transform.

AMS Classification: 65D20, 65G05

Key Words: Elliptic functions, Landen transform, AGM-method, a priori rounding error estimation

1 Introduction and recent results

In 1994 [LuOt94] we have published rigorous a priori estimates for the real AGM-method and the ascending Landen transform by considering errors inherent in the floating-point representation as well as round-off errors in the arithmetic to calculate the square root-, logarithm- and arctan-function and their inverses. The special interest in the AGM-method arises form quadratic convergence of these algorithms, so that fast and reliable calculations are possible. Later [LuOt96] we have extended the method to calculate the corresponding complex- and matrix-valued functions.

In his thesis [W96] Werner developed a cancellation-free algorithm to evaluate the inverse Weierstraß-function

$$\mathcal{P}^{-1}(u; e_1, e_2, e_3) := \int_u^\infty \frac{dx}{\sqrt{(x-e_1)(x-e_2)(x-e_3)}},$$
$$e_3 < e_2 < e_1 \leq u,\ e_1 + e_2 + e_3 = 0.$$

He also analyses the ascending Landen transform to calculate the Jacobian elliptic functions $sn(u, m) = \sin\varphi$, $cn(u, m)$ and $dn(u, m)$, where

$$u = F(\varphi, k) = \int_0^\varphi \left(1 - k^2 \sin^2 \vartheta\right)^{-1/2} d\vartheta, m = k^2.$$

Remark that

$$\frac{\sqrt{e_1 - e_3}}{2} \mathcal{P}^{-1}(u; e_1, e_2, e_3) = F\left(\arccos\sqrt{\frac{u-e_1}{u-e_3}}, \sqrt{\frac{e_2-e_3}{e_1-e_3}}\right).$$

Thus we have

$$m = \frac{e_2 - e_3}{e_1 - e_3},\ \cos^2\varphi = \frac{u - e_1}{u - e_3}.$$

However, this transformation holds only for $\sqrt{\varepsilon_\ell} \le \varphi \le \pi/2 - \sqrt{\varepsilon_\ell}$, where ε_ℓ denotes the screening-ε. The complete results read as follows

Theorem 1. *With*

$$\begin{aligned}
a_0 &:= \sqrt{e_1 - e_3},\ b_0 := \sqrt{e_1 - e_2},\ a_{i+1} := (a_i + b_i)/2,\\
b_{i+1} &:= \sqrt{a_i \cdot b_i},\ d_0 := u_0 - e_1,\\
d_{i+1} &:= \frac{1}{2} d_i \left(1 + \frac{a_i^2 + b_i^2 + d_i}{\sqrt{(a_i^2 + d_i)(b_i^2 + d_i)} + a_i b_i}\right),\\
n &:= \lceil \mathrm{ld}(2 + \mathrm{ld}\mathcal{B} \cdot (\ell - 1)/2) \rceil + \lceil \mathrm{ld}(\mathrm{ld}\mathcal{B} \cdot (\ell - 1) - 1) \rceil,\\
\varepsilon_\ell &:= \mathcal{B}^{1-\ell}, \mathcal{B} := 2^\beta, \varepsilon_\ell \le 2^{-52},
\end{aligned}$$

it holds that

$$\begin{aligned}
\mathcal{P}^{-1}(u_0; e_1, e_2, e_3) &= 2/_\ell B_n \mathrm{Arctan}(B_n/_\ell \mathrm{Sqrt}_\ell D_n) \cdot (1 + \delta\varepsilon_\ell),\\
|\delta| &\le 8.3 \cdot 3.00001^n + 4.0001n + 8.
\end{aligned}$$

Here u_0, e_1, e_2, e_3 are machine numbers and A_n, B_n, D_n, Arctan machine approximations of a_n, b_n, d_n, arctan and $/_\ell$ the machine division.

In the above theorem $\mathrm{ld}(\cdot) = \log_2(\cdot)$ denotes the dual logarithm. In the sequel we denote the machine approximations by capital letters. Putting

$$\begin{aligned}
a_0 &:= 1,\ b_0 := k_0',\ c_0 := k_0, \quad n := 2\lceil \mathrm{ld}(\mathrm{ld}\mathcal{B} \cdot (\ell - 1)) \rceil - 1,\\
t_n &:= \frac{a_n + b_n}{2\sin(u \cdot b_n)},\\
c_{i+1} &:= \frac{c_i^2}{4a_{i+1}} = \frac{a_i - b_i}{2} = a_i - a_{i+1},\ i = 0, 1, 2, 3, ...,\\
t_i &:= t_{i+1} + \frac{c_i^2}{4t_{i+1}},\ i := n - 1, ..., 0,
\end{aligned}$$

Werner shows that

$$sn(u, m) = T_0^{-1} \cdot (1 + \delta_\ell \varepsilon_\ell).$$

The value of δ_ℓ depends on the choice of the base $\mathcal{B}$ and the exponent ℓ. They are given in a precalculated table, e.g. $\delta_\ell \le 2^{13}$ for the IEEE double format and $\delta_\ell \le 2^{15}$ for the quad-format with 128 Bits.

In two other notes we have considered the descending Landen transform to complete our studies of the AGM-method and elliptic functions. First we have developed a new algorithm for the evaluation of

$$F(\varphi, k),\ \varepsilon_\ell < \varphi \le \pi/2 - \varepsilon_\ell,\ 2\varepsilon_\ell \le k^2 \le 1 - 2\varepsilon_\ell,\ \varphi, k^2 \in S',$$

which avoids cancellation [LuOt97]. The same method was utilized to derive bounds for the absolute error of each term in the series representation of Jacobi's Zeta function

$$\widetilde{Z}(\varphi,k) := E(\varphi,k) - (E/K)F(\varphi,k) = \sum\nolimits_{i\geq 1} c_i \sin\varphi_i$$

$$E := E\left(\frac{\pi}{2},k\right) = \int\limits_0^{\pi/2} \sqrt{1-k^2\sin^2\vartheta}\,d\vartheta,\ K := F\left(\frac{\pi}{2},k\right)$$

and the relative error of the product representation of Jacobi's Theta function $\widetilde{\Theta}(\varphi,k)$[LuOt97-2].

We complete our definitions and put as above

$$\begin{aligned} k_0 &:= k,\ k_0' = \sqrt{1-k^2},\ \varphi_0 := \varphi,\\ k_i' &:= \frac{b_i}{a_i},\ k_i := \frac{c_i}{a_i},\\ \tan\varphi_{i+1} &:= \frac{(1+k_i')\tan\varphi_i}{1-k_i'\tan^2\varphi_i}. \end{aligned} \tag{1}$$

The descending Landen transform states that

$$\frac{1}{2^i a_i}F(\varphi_i,k_i) = \frac{1}{2^{i+1}a_{i+1}}F(\varphi_{i+1},k_{i+1}),\ i = 0,1,2,3,....$$

The sequence $\{a_i\}$ and $\{b_i\}$ tend to the limit agm, $\varphi_i/2^i$ to ξ, $\varphi_i/(b_i 2^i)$ decreases to $u = F(\varphi,k) = \xi/agm$ as well as $\varphi_i/(a_i 2^i)$ increases to ξ/agm as i tends to infinity. For the approximation error it was proved:

Theorem 2. *Choose $n \in \mathbf{N}$ such that $1 - k_n' < \varepsilon_\ell$. Then it holds that*

$$\frac{\varphi_n}{2^n a_n} = \frac{\xi}{agm}(1+\delta\varepsilon_\ell), |\delta| \leq 1.$$

We have shown in [LuOt97] that applying the AGM-method with $\mathrm{ld}\varepsilon_\ell \geq -2^{n/2}$ after n iteration steps we have $1 - k_n' < \varepsilon_\ell$.

2 Basic error analysis

Now we start with two machine numbers $\varphi_0 = \varphi \in (0,\pi/2)$ and $k^2 \in (0,1)$ belonging to the floating-point screen $S' := S(\mathcal{B},\ell',em',eM')$ with its even base $\mathcal{B}$, mantissa length ℓ' and $[em',eM']$ smallest and largest allowable exponent, respectively. Computations require guard digits and are made in a finer screen $S := S(\mathcal{B},\ell,em,eM), \ell' < \ell \leq \ell' + const.,\ em \leq em', eM \geq eM'$.

The relative error for all elementary operations $\times$ with machine numbers x and y is assumed to be bounded by

$$\frac{|x \times_l y - x \times y|}{|x \times y|} < \varepsilon_\ell.$$

We assume $\varepsilon_\ell < 10^{-4}$ and mention some basic error estimations [LuOt94].

Given two numbers a, b and their corresponding machine approximations A, B with $|A - a| \le |a|\,\varepsilon_a$ and $|B - b| \le |b|\,\varepsilon_b$, then it holds

$$\begin{aligned}\frac{|(A +_\ell B) - (a+b)|}{|a+b|} &\le \varepsilon_\ell + \left|\frac{1+\varepsilon_\ell}{a+b}\right| \{|a|\,\varepsilon_a + |b|\,\varepsilon_b\}\,,\\ \frac{|(A \cdot_\ell B) - (a \cdot b)|}{|a \cdot b|} &\le \varepsilon_\ell + (1+\varepsilon_\ell)\{\varepsilon_a + \varepsilon_b + \varepsilon_a \cdot \varepsilon_b\}\,,\\ \frac{|(A/_\ell B) - (a/b)|}{|a/b|} &\le \varepsilon_\ell + (1+\varepsilon_\ell)\left\{\frac{\varepsilon_a + \varepsilon_b}{1-\varepsilon_b}\right\}.\end{aligned} \tag{2}$$

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

Werner [W96-2] obtained a complete error analysis for the evaluation of the square root by using Newton's method. His result reads as follows:

Starting from an initial value

$$y_0 = (1+x)/2 \text{ where } x \in [0.5, 2] \cap S^0,\ \mathcal{B} = 2^\beta,\ \varepsilon_\ell \le 2^{-52},$$

the relative error of the square root $\sqrt{x}$ calculated by Newton's method

$$y_n = y_{n-1} - \frac{(y_{n-1}^2 - x)}{2y_{n-1}},\ n \ge \lceil \mathrm{ld}(\beta(\ell-1)-2)\rceil + 2$$

is bounded by $1.50001\varepsilon_\ell$.

Under the assumption $(i+1)^2\varepsilon_\ell < 10^{-8}$ it was proved in [LuOt94] and [W96-2] that starting from machine numbers

$$A_0 = 1;\ B_0 = k_0'(1 + \delta_0' \cdot 2.001\varepsilon_\ell),\ |\delta_0'| \le 1,$$

and applying AGM-iteration we find after i steps a relative error of order

$$\begin{aligned}A_i &= a_i(1 + \delta_1' \cdot 2.001 \cdot (i+1)\varepsilon_\ell),\\ B_i &= b_i(1 + \delta_1' \cdot 2.001 \cdot (i+1)\varepsilon_\ell),\ |\delta_1'| \le 1.\end{aligned}$$

Assuming a sharper restriction $\varepsilon_\ell \le 2^{-52}$, we see that $i^4\varepsilon_\ell \le 10^{-3}$, if $i \le 1000$, and we can derive a bound $\gamma_i \le 2.021^{i+3}\varepsilon_\ell$ for the relative error γ_i of the sequence $\{C_i\}$,

$$C_i = c_i(1 + \delta_i' \cdot \gamma_i),\ |\delta_i'| \le 1,$$

involved in the calculation of Jacobi's Zeta-function by using

$$\begin{aligned}C_1 &= \frac{k^2}{4a_1}(1 + \delta_1' \cdot 6.21\varepsilon_\ell),\ c_{i+1} = \frac{c_i^2}{4a_{i+1}},\\ \gamma_{i+1} &\le 2.001\gamma_i + 2.01(i+3.5)\varepsilon_\ell.\end{aligned}$$

This result shows that the absolute error in the representation of c_i by the machine number C_i is roughly speaking bounded by $8\varepsilon_\ell$.

Furthermore, it holds

$$\begin{aligned}K_1' &= k_1'(1 + \delta_2' \cdot 6.78\varepsilon_\ell), |\delta_2'| \le 1,\\ K_i' &= k_i'(1 + \delta_3'(1.01 + 4.2 \cdot (i+1))\varepsilon_\ell), |\delta_3'| \le 1, i > 1.\end{aligned} \tag{3}$$

3 Jacobi's Theta-function

Now we give an algorithm to compute Jacobi's Theta-function

$$\Theta(u,m) = \widetilde{\Theta}(\varphi,k) = \left(\frac{2k'K}{\pi}\right)^{1/2} \prod\nolimits_{i\geq 0} \left(1 - k_i^2 \sin^2 \varphi_i\right)^{-1/2^{i+2}}$$

without cancellation including a bound for the relative error.

Algorithm 1. (Theta-Function)

1. Take the values $agm, k_i',\ \cot\psi_i, \cot\varphi_i$ previously computed.
2. Initialize
$$\Theta_n := \frac{1}{k_n'}\sqrt{\frac{1+\cot^2\varphi_n}{1+\cot^2\psi_n}}.$$
3. Loop:
For $i := n-1$ downto 0 do
$$\Theta_i := \frac{1}{k_i'}\sqrt{\Theta_{i+1}\cdot\frac{1+\cot^2\varphi_i}{1+\cot^2\psi_i}}$$
4. End:
$$\Theta(u,m) := \sqrt{\Theta_0 \cdot \frac{k_0'}{agm}}.$$

Algorithm 2.

1. Initialization:
   a) We enter the argument $u, 0 < u \leq K(1-\varepsilon_\ell)$ and the second argument k^2 fulfilling $2\varepsilon_\ell \leq k^2 \leq 1 - 2\varepsilon_\ell,\ u, k^2 \in S'$.
   b) We put $k_0 := k,\ a_0 := 1,\ b_0 := k_0'$.
2. Iteration:
   a) We calculate successively $a_{i+1}, b_{i+1}, k_{i+1}', \sqrt{k_{i+1}'}, 2^{i+1}, i = 0, ..., n-1$.
   b) If $1 - k_n' < \varepsilon_\ell$ (i.e. $n \geq 2\mathrm{ld}(\mathrm{ld}(1/\varepsilon_\ell))$) we put
   $$agm := a_n,\ \varphi_n := agm\cdot 2^n\cdot u,\ j_n := \lfloor 2\varphi_n/\pi\rfloor, \cot\psi_n := \cot\varphi_n/k_n'.$$
   Then we compute successively for $i := n, ..., 1$,
   $$j_{i-1} := \lfloor j_i/2 \rfloor,$$
   $$\cot\psi_{i-1} := \begin{cases} \left(\cot\psi_i + \sqrt{1+\cot^2\psi_i}\right)/\sqrt{k_{i-1}'}, & \text{if } j_{i-1} \text{ even},\\ -1/\left(\left(\cot\psi_i + \sqrt{1+\cot^2\psi_i}\right)\cdot\sqrt{k_{i-1}'}\right), & \text{if } j_{i-1} \text{ odd}.\end{cases}$$
3. End:
We take the values n, $agm, k_i', \cot\psi_i, \cot\varphi_i$ and compute $\Theta(u,m)$ as pointed out in Algorithm 1.

Now we want to estimate the relative error of our machine approximation $\Theta(u,m)$. Starting from

$$X = x(1+\delta_4'\varepsilon_\ell),\ |\delta_4'| \le 3(n+3)^2\varepsilon_\ell,$$

we derive

$$X +_\ell \mathrm{Sqrt}_\ell(1 +_\ell X \cdot_\ell X) = (x+\sqrt{1+x^2})(1+\delta_4'\varepsilon_\ell + 3.53\varepsilon_\ell).$$

We first consider algorithm 2. By (2) and an accurate cotangent-evaluation, we derive

$$\mathrm{Cot}\psi_n = \cot\psi_n(1+\delta_n\varepsilon_\ell), |\delta_n| \le 2.1(n+4),$$

and by induction for $i = n-1$ to 0

$$\mathrm{Cot}\psi_i = \cot\psi_i(1+\delta_i\varepsilon_\ell),\ |\delta_i| \le 2.1\cdot(n+1-i)(n+4).$$

The same estimation is valid for Cot φ_i. Defining

$$R_i := (1 +_\ell \mathrm{Cot}^2\ \varphi_i)/_\ell(1 +_\ell \mathrm{Cot}^2\ \psi_i),$$

in an analogous way we infer

$$R_i = \frac{1+\cot^2\ \varphi_i}{1+\cot^2\ \psi_i}\left(1+\delta_5'\left(5.05+8.4(n+4)\left(n+1-i\right)\right)\varepsilon_\ell\right), |\delta_5'| \le 1.$$

Using (3) and starting in algorithm 1, step 2, we have an error bound for Θ_n with $|\delta_6'| \le 1$:

$$\mathrm{Sqrt}_\ell R_n/_\ell K_n' = \frac{1}{k_n'}\sqrt{\frac{1+\cot^2\ \varphi_n}{1+\cot^2\ \psi_n}}\left(1+\delta_6'\left(4.2(n+4)+5.05\right)\varepsilon_\ell\right).$$

By induction we derive the following bound for the relative error ζ_i of our machine approximation Θ_i :

$$|\zeta_i| \le (8.4(n+4)(n+1-i)+5.05)\,\varepsilon_\ell,\ i = n-1,...,0.$$

The last term k_0'/agm can by calculated with a relative error bounded by $2.1(n+1)\varepsilon_\ell + 3.03\varepsilon_\ell$ and after a multiplication and root extraction the one of Θ is bounded by

$$(4.2(n+4.25)(n+1)+6.1)\,\varepsilon_\ell$$

Thus we have proved

Theorem 3. *Calculating* $\Theta(u,m)$,

$$0 < u \le K(1-\varepsilon_\ell),\ 2\varepsilon_\ell \le k^2 \le 1-2\varepsilon_\ell,\quad u,k^2 \in S',\ \varepsilon_\ell \le 2^{-52},$$

as indicated in Algorithms 1 and 2, the relative error is bounded by

$$(6.1+4.2(n+4.25)(n+1))\,\varepsilon_\ell.$$

Remark: In the same way we find

$$\mathrm{Sin}\varphi_i = \sin\varphi_i(1+\sigma_i\varepsilon_\ell), |\sigma_i| \le 2.1\cdot(n+1-i)(n+4)+3.2.$$

For $i=0$ we have a relative error bound for the machine approximation of

$$sn(u,m) = \sin\varphi_0 = \sqrt{\frac{1}{1+\cot^2\varphi_0}}$$

of order $(2.1\cdot(n+1)(n+4)+3.2)\,\varepsilon_{\ell.}$ An analogous estimation holds for

$$dn(u,m) = \sqrt{\frac{1+\tan^2\psi_0}{1+\tan^2\varphi_0}}$$

with a relative error bounded by $(4.2\cdot(n+1)(n+4)+4.6)\,\varepsilon_{\ell.}$

By the way we have found a error estimation for the machine approximation of Jacobi's Zeta-function

$$Z(u,m) = \sum\nolimits_{i\ge1} c_i \sin\varphi_i$$

introducing

$$\mathrm{Sin}\varphi_i = \sin\varphi_i(1+\sigma_i\cdot 4.2(n+4)(n+1-i)\varepsilon_\ell + 3.2), |\sigma_i| \le 1,$$

and $C_i = c_i(1+\delta_i'\cdot 2.021^{i+3}),\ |\delta_i'| \le 1.$

There is another definition of Jacobi's theta-function as a Fourier series

$$\vartheta_4\left(\frac{\pi u}{2K}\right) = \Theta(u,m) = 1+2\sum_{i=1}^{\infty}(-1)^i \exp\left(-\frac{\pi K'}{K}i^2\right)\cos\left(i\frac{\pi u}{K}\right).$$

Remark that

$$K(k) = \frac{\pi}{2agm},\ K' := K(k') = \frac{\pi}{2agm'},$$
$$agm' = \lim_{i\to\infty} a_i',\ a_0' := 1, b_0' = k_0.$$

We prefer our method for large ℓ because the series converges slowly for large K. If $k = 1-\varepsilon_\ell = 1-2^{1-\ell}$ we have the asymptotic relation [LuOt96]

$$\left|\frac{\pi K(k)}{K'(k)} - \ln\frac{16}{k'^2}\right| \le \frac{k'^2}{2\,(1-5k'^2/4)},\ k'\to 0,$$

$$q := \exp\left(-\frac{\pi K'}{K}\right) \sim \exp\left(-\frac{\pi^2}{(\ell+2)\ln 2}\right) \tag{4}$$
$$k := 4\sqrt{q}\prod\nolimits_{i\ge1}\left(\frac{1+q^{2i}}{1+q^{2i-1}}\right)^4.$$

and $q^{(i^2)}$ stays nearby one for small i.

The product-representation can be used to find a first approximation for k as a function of q, when we calculate the inverse function to $q(k)$ with the aid of Newton's method. We have [BoBo84]

$$\begin{aligned} q(k) &= \exp\left(-\pi \frac{agm}{agm'}\right), \ \frac{dq}{dk} = -\pi q \frac{d}{dk}\frac{agm}{agm'}, \\ \frac{d\,agm'}{dk} &= \lim_{i\to\infty} \widetilde{a}_i, \ \widetilde{a}_0 := 0, \ \widetilde{b}_0 := 1, \\ & \widetilde{a}_{i+1} := (\widetilde{a}_i + \widetilde{b}_i)/2, \ \widetilde{b}_{i+1} := \left(\widetilde{a}_i \sqrt{\frac{b_i'}{a_i'}} + \widetilde{b}_i \sqrt{\frac{a_i'}{b_i'}}\right)/2, \end{aligned}$$

and there is a similar relation for

$$\frac{d\,agm}{dk} = \frac{d\,agm}{dk'}\frac{-k}{\sqrt{1-k^2}}.$$

Thus we have developed a quadratic convergent algorithm to calculate $\vartheta_4(v)$:

Algorithm 3.

1. Initialization:
   Find a first approximation $\widetilde{k}$ of $k = k(q)$ using the product-representation of k in (4).
2. Iteration:
   Calculate by Newton's and AGM iteration $k = k(q)$.
3. End:
   Compute $\Theta(2Kv/\pi, k^2)$ by our algorithms 1 and 2.

We will apply our results to solve a partial differential equation: Sugihara and Fujino [SF96] discuss Burgers' equation

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \nu\frac{\partial^2 u}{\partial x^2}, \ 0 \le x \le 1, \ u(x,0) = u_0(x), \ u(0,t) = u(1,t) = 0,$$

with large Reynolds–number $1/\nu$. They derive a representation of the exact solution including integrations of ϑ_3,

$$\begin{aligned} u(x,t) &= \frac{\int_{-1}^{+1} u_{0,odd}(\eta) w(\eta)\vartheta_3\left(0.5(x-\eta), \exp(-\pi^2\nu t)\right) d\eta}{\int_{-1}^{+1} w(\eta)\vartheta_3\left(0.5(x-\eta), \exp(-\pi^2\nu t)\right) d\eta}, \\ u_{0,odd}(-x) &:= -u_0(x), w(\eta) := \exp\left(-\frac{1}{2\nu}\int_0^\eta u_{0,odd}(\xi)d\xi\right), \end{aligned}$$

and consider

$$\vartheta_3\left(\frac{\pi u}{2K}\right) = \vartheta_4\left(\frac{\pi(K-u)}{2K}\right) \text{ for arguments } \nu := \frac{K'}{\pi K} \le 0.02$$

and u near K. It holds $q = \exp(-\pi^2 \nu t) \approx 1$, and the infinite series converges very slowly. If we assume $t = 1$, we see from (4) that a precision of $1/(\nu \cdot \ln 2)$ binary digits is necessary to apply our algorithms 1 and 2 for calculating $\vartheta_3\left(0.5(x-\eta), \exp(-\pi^2 \nu t)\right)$ and to achieve correct results including error bounds.

At the moment we implement a function library including all elementary and elliptic functions utilizing the C++–platform BIAS and arbitrary floating point screens.

However, in a recent talk on the SCAN–97 conference at Lyon Sugihara and Fujino proposed together with Hoshino another numerical method for the exact solution of Burgers' equation using the Jacobian Imaginary Transform

$$\vartheta_3(u,q) = \frac{1}{\sqrt{\pi t \nu}} \exp\left(-\frac{u^2}{\nu t}\right) \left\{1 + 2\sum_{n=1}^{\infty} \exp\left(-\frac{n^2}{\nu t}\right) \cosh \frac{2nu}{\nu t}\right\}.$$

When $\nu \leq 10^{-3}$, they cannot apply this representation because of the range limitation $[3.4 \cdot 10^{-4932}, 1.1 \cdot 10^{4932}]$ of the double extended IEEE-format. Thus, they redefine arithmetics to deal with large numbers about $10^{2.77 \cdot 10^{18}}$ in order to handle Burgers' equation with Reynolds–numbers up to 10^8.

References

J. M. Borwein, P. B. Borwein: The arithmetic-geometric mean and fast computation of elementary functions. SIAM Reviews 26 (1984), pp. 351-366

W. Luther, W. Otten: Computation of Standard Interval Functions in Multiple-Precision Interval Arithmetic. Interval Computations 4 (1994), pp. 78-99

W. Luther, W. Otten: The complex arithmetic-geometric mean and multiple-precision matrix functions. Scientific Computing and Validated Numerics. G. Alefeld, A. Frommer, B. Lang eds. Akad. Verlag Berlin 1996, pp. 52-58

W. Luther, W. Otten: Approximation Error and Error Accumulation for the Landen Transform. Reliable Computing, No. 3, 1997, pp. 249-258

W. Luther, W. Otten: Accurate Computation of Elliptic Functions. Submitted to ZAMM

M. Sugihara, S. Fujino: Numerical solutions of Burgers' Equation with a large Reynolds number. Reliable Computing, No. 2, 1996, pp. 173-179

M. Sugihara, S. Fujino, F. Hoshino: A numerical method for the exact solution of Burgers' equation using overload function of C++. SCAN-97, Lyon, France, September 10-12, 1997

K. Werner: Calculation of the Inverse Weierstraß-Function in an Arbitrary Machine Arithmetic. Scientific Computing and Validated Numerics. G. Alefeld, A. Frommer, B. Lang eds. Akad. Verlag Berlin 1996, pp. 72-78

K. Werner: Verifizierte Berechnung der inversen Weierstraß-Funktion und der elliptischen Funktionen von Jacobi in beliebigen Maschinenarithmetiken. PhD-Thesis, Duisburg 1996 Addresses:

Wolfram Luther; Werner Otten. Gerhard-Mercator-Universität - GH Duisburg. Fachbereich Mathematik / Informatik II, Lotharstrasse 65, D-47048 Duisburg