Mathematics: Complex functions

Marcel Leutenegger

École Polytechnique Fédérale de Lausanne
Laboratoire d'Optique Biomédicale
1015 Lausanne, Switzerland

January 2, 2005

The efficient computation of transcendental functions for complex floating point numbers is an important issue for any number crunching software as MATLAB for example. Efficient code is based on the analytical simplification of the function definition together with a reuse of intermediate values.

The limited precision of the representation of floating point values can lead to significant errors. In particular, the evaluation of complex values frequently involves additions and subtractions. Their round-off error potentially degrades the accuracy of the final result. This issue is briefly discussed within this document.

Contents

1 Introduction

Complex values z are represented as two floating point numbers representing the real and the imaginary part respectively. Hence

$$z = \Re(z) + i\Im(z) = a + ib \qquad \text{where } i = \sqrt{-1} \text{ is the complex unit.}$$

The following abbreviations are used for function arguments and results:

f is a finite real value $\in (-\infty, +\infty)$,
n is a finite, negative real value $\in (-\infty, -0]$,
p is a finite, positive real value $\in [+0, +\infty)$,
r is a real value $\in [-\infty, +\infty]$,
u is an undefined value (NAN) and
a is an affine value $\in \{\pm\infty, \pm\text{NAN}\}$.

The result of an undefined argument (NAN) is in general undefined. Only the rare exceptions are listed hereoff.

Although, the sign does not matter, the MATLAB representation of +NAN is effectively the IEEE representation of -NAN.

2 Basic functions

2.1 Exponential

$$\exp(z) = e^a \cos(b) + ie^a \sin(b) \tag{1}$$

See also the comment on π in section 3 Transcendental functions.

Argument	Result
$-\infty$	0
f	r
$+\infty$	$+\infty$
$f+if$	$r+ir$
$f+ia$	$u+iu$
$-\infty+if$	0
$-\infty+ia$	0
$+\infty+if$	$a+ia$
$+\infty\pm i\infty$	$u+iu$

2.2 Natural logarithm

$$\log(z) = \frac{1}{2}\log(a^2 + b^2) + i\,\mathrm{atan}\,\frac{b}{a} \tag{2}$$

The logarithm "compresses" the dynamic range of the input value rendering it sensible to a misproportion between $|a|$ and $|b|$. Such a misproportion leads to a round-off error in the subexpression $a^2 + b^2$ affecting the result. Over the full complex plane $\mathbb{R} \times i\mathbb{R}$, the logarithm is computed as

$$\log(z) = \log|a| + \frac{1}{2}\log\left(1 + \frac{b^2}{a^2}\right) + i\,\mathrm{atan}\,\frac{b}{a} \qquad \forall\ a^2 > 2^9 b^2 \tag{3}$$

$$\log(z) = \log|b| + \frac{1}{2}\log\left(1 + \frac{a^2}{b^2}\right) + i\,\mathrm{atan}\,\frac{b}{a} \qquad \forall\ b^2 > 2^9 a^2 \tag{4}$$

The ratio is 2 to the power of the mantissa length difference (64bit - 53bit) less 2 rounding bits, hence $2^9 = 512$.

The logarithm involved in the inverse transcendental functions is always computed with equation (2). Due to the three to four addends involved, split-up leads to a manifold of different expressions. Conditional branches to these expressions are unpredictable and degrade the overall performance significantly.

Argument	Result
$-\infty$	$+\infty+i\pi$
n	$r+i\pi$
0	$-\infty$
p	r
$+\infty$	$+\infty$
$r+ir$	$r+if$

2.3 Square root

In principle, the square root could be computed as

$$\operatorname{sqrt}(z) = \sqrt{\frac{|z|+a}{2}} \pm i\sqrt{\frac{|z|-a}{2}} \qquad \text{with} \qquad |z| = \sqrt{a^2+b^2} \tag{5}$$

But for $|a| \gg |b|$, the imaginary part would be truncated to zero. To maintain precision over the entire complex plane $\mathbb{R} \times i\mathbb{R}$, the square root is implemented as

$$\operatorname{sqrt}(z) = \sqrt{\frac{|z|+a}{2}} + i\frac{b}{2}\sqrt{\frac{2}{|z|+a}} \qquad \forall\, a > 0 \tag{6}$$

$$\operatorname{sqrt}(z) = \frac{|b|}{2}\sqrt{\frac{2}{|z|-a}} \pm i\sqrt{\frac{|z|-a}{2}} \qquad \forall\, a < 0 \tag{7}$$

The sign in equation (6) is automatically correct. In equation (7), it is adjusted accordingly.

Argument	**Result**
$-\infty$	$+i\infty$
n	ip
p	p
$+\infty$	$+\infty$
$f+if$	$f+if$
$f\pm i\infty$	$+\infty\pm i\infty$
$-\infty+if$	$+i\infty$
$+\infty+if$	$+\infty$
$\pm\infty\pm i\infty$	$u+iu$

3 Transcendental functions

The results of the transcendentals and their inverse functions will not be given explicitely hereoff. For affine arguments, they reply either by an undefined real or complex value or by

$$\text{function}\,(\infty + i\infty) = \lim_{z\to\infty+i\infty} \text{function}\,(z)$$

Comment on π

The transcendental FPU operations *fcos*, *fsin*, *fsincos* and *fptan* reduce their arguments by computing the 2π-reminder first. They use an internal representation of π with a 66bit mantissa regardless of the precision setting. Normal floating point operations never exceed the 64bit mantissa of the *temporary real* format.

To minimize round-off error, the arguments for the transcendental FPU functions are explicitly reduced by taking the 2π-reminder represented with a 53bit mantissa as *double* value. In combination with a prescaling by a *temporary real* factor, the round-off error is reduced to less than 5.5E-20. As a drawback, the performance is lowered by about 10% to 15%.

3.1 Cosine

$$\cos(z) = \frac{e^{iz} + e^{-iz}}{2} = \frac{1}{2}\left(e^{-b} + e^{b}\right)\cos(a) + \frac{i}{2}\left(e^{-b} - e^{b}\right)\sin(a) \tag{8}$$

3.2 Hyperbolic cosine

$$\cosh(z) = \cos(b - ia) = \frac{1}{2}\left(e^{a} + e^{-a}\right)\cos(b) + \frac{i}{2}\left(e^{a} - e^{-a}\right)\sin(b) \tag{9}$$

3.3 Sine

$$\sin(z) = \frac{e^{iz} - e^{-iz}}{2i} = \frac{1}{2}\left(e^{b} + e^{-b}\right)\sin(a) + \frac{i}{2}\left(e^{b} - e^{-b}\right)\cos(a) \tag{10}$$

3.4 Hyperbolic sine

$$\sinh(z) = i\sin(b - ia) = \frac{1}{2}\left(e^{a} - e^{-a}\right)\cos(b) + \frac{i}{2}\left(e^{a} + e^{-a}\right)\sin(b) \tag{11}$$

3.5 Tangent

$$\tan(z) = \frac{\sin(z)}{\cos(z)} = \frac{4e^{2b}\cos(a)\sin(a) + i\left(e^{2b} + 1\right)\left(e^{2b} - 1\right)}{\left(\left(e^{2b} + 1\right)\cos(a)\right)^2 + \left(\left(e^{2b} - 1\right)\sin(a)\right)^2} \tag{12}$$

The tangent maintains precision up to an imaginary part $|b| \lesssim 373$. A larger $|b|$ causes either an overflow of the subexpression e^{2b} to infinity or a truncation of the real part of the result to zero when saving as *double*. For a *single* (*extended*) result, the limit would be $|b| \lesssim 53$ (5678).

MATLAB maintains precision for $|b| \lesssim 19$.

3.6 Hyperbolic tangent

$$\tanh(z) = \frac{\sinh(z)}{\cosh(z)} = \frac{\left(e^{2a}+1\right)\left(e^{2a}-1\right) + i4e^{2a}\cos(b)\sin(b)}{\left((e^{2a}+1)\cos(b)\right)^2 + \left((e^{2a}-1)\sin(b)\right)^2} \tag{13}$$

The hyperbolic tangent maintains precision up to a real part $|a| \lesssim 373$. A larger $|a|$ causes either an overflow of the subexpression e^{2a} to infinity or a truncation of the imaginary part of the result to zero when saving as *double*. For a *single* (*extended*) result, the limit would be $|b| \lesssim 53$ (5678).

MATLAB maintains precision for $|a| \lesssim 20$.

4 Inverse transcendental functions

4.1 Inverse cosine

$$\operatorname{acos}(z) = -i \log\left(a + ib + \sqrt{a^2 - b^2 - 1 + 2iab}\right) \tag{14}$$

To maximise precision, the function computes equation (14) always with the absolute values $|a|$ and $|b|$ respectively. It adjusts then the real part of the result regarding the signs of the components in z. Therefore, the result will always have a negative imaginary part.

4.2 Inverse hyperbolic cosine

$$\operatorname{acosh}(z) = \log\left(a + ib + \sqrt{a^2 - b^2 - 1 + 2iab}\right) \tag{15}$$

To maximise precision, the function computes equation (15) always with the absolute values $|a|$ and $|b|$ respectively. It adjusts then the imaginary part of the result regarding the signs of the components in z. Therefore, the result will always have a positive real part.

4.3 Inverse sine

$$\operatorname{asin}(z) = -i \log\left(ia - b + \sqrt{1 - a^2 + b^2 - 2iab}\right) \tag{16}$$

To maximise precision, the function computes equation (16) with the absolute values $|a|$ and $|b|$ respectively. It adjusts then the result accordingly.

4.4 Inverse hyperbolic sine

$$\operatorname{asinh}(z) = \log\left(a + ib + \sqrt{1 + a^2 - b^2 + 2iab}\right) \tag{17}$$

To maximise precision, the function computes equation (17) with the absolute values $|a|$ and $|b|$ respectively. It adjusts then the result accordingly.

4.5 Inverse tangent

$$\operatorname{atan}(z) = -\frac{i}{2} \log \frac{1 - a^2 - b^2 + 2ia}{1 + a^2 + b^2 + 2b} \tag{18}$$

The function improves precision by computing equation (18) with the absolute values $|a|$ and $|b|$. The signs of the real and imaginary parts of the result are then set to the signs of the input components.

4.6 Inverse hyperbolic tangent

$$\operatorname{atanh}(z) = -\frac{1}{2} \log \frac{1 - a^2 - b^2 - 2ib}{1 + a^2 + b^2 + 2a} \tag{19}$$

The function improves precision by computing equation (19) with the absolute values $|a|$ and $|b|$. The signs of the real and imaginary parts of the result are then set explicitly.