Interval Arithmetic

?) was one of the early developers of the techniques Interval Analysis.

While it is not possible to implement true interval arithmetic on computers (any more than it is possible to implement true scalar arithmetic), it is possible to produce an implementation which guarantees an enclosure of all arithmetic operations.

We shall implement our interval arithmetic calculations using the BIAS/PROFIL (Basic Interval Arithmetic Subroutines / Programmer's Runtime Optimized Fast Interval Library) package ??) in C++. The BIAS software is modelled after the Basic Linear Algebra Subroutines (BLAS) software. Specifically, careful thought and testing has gone into the devlopment of the software to achieve optimal speed and accuracy in the computed results.

While BIAS/PROFIL contains C++ classes for complex numbers and operations, we have found it necessary to add an implementation of a class for interval complex arithmetic.

The extension from complex arithmetic to interval complex arithmetic is not quite as intuitive as the extension from real arithmetic in interval real arithmetic. One possible extension is via the use of disks in the complex plane, i.e. by the pair (z, r) where $z \in C$ is the center of the disk and r is its radius. See ?) for a brief discussion.

More common is the representation of an interval complex number as a rectangle. Such a rectangle can be represented by a pair of complex numbers denoting the lower left and upper right corners as in ?), or as a pair of interverals (r, i) where r and i are in the real and imaginary coordinate directions. These two representations of a complex interval rectangle are equivalent, differing only in the software data structures used to support the implementation. We have chosen to use the (r, i) representation, which seems to be natural to use with the BIAS/PROFIL package.

Given this representation, arithmetic operations on interval complex numbers can be implemented quite easily. However, how to achieve the best implementation is not transparent. This results from the unfortunate characteristic that for rectangular complex interval numbers z_1 and z_2,

$$S = \{z_1/z_2 | z_1 \in z_1, z_2 \in z_2\}$$

is generally not a rectangular region.

?) presented an algorithm for obtaining a small rectangular enclosure of S. The scalar complex arithmetic algorithm in PROFIL, which calculates $r = d/c$ and then

$$\frac{a + bi}{c + di} = \frac{(a + br) + (b - ar)i}{c + dr}$$

could also be used in hopes of achieving a fast answer. In the context where we use interval complex arithmetic, the intervals are degenerate (or nearly so, given rounding). This may explain why we have found the usual definition of complex division, implemented with interval arguments, to give the best results.

Complex Error Function

The error function of a complex number z is defined as

$$erf(z) = \frac{2}{\sqrt{\pi}} \int_0^z e^{-t^2} dt$$

and the complementary error function is given as $erfc(z) = 1 - erf(z)$. A widely-used related function is the plasma dispersion function

$$w(z) = e^{-z^2} erfc(-iz) \tag{0.1}$$

.

Test: The Faddeeva function 0.1 is OK.

Let $\overline{z}$ denote the complex conjugate of z and i denote the square root of -1. Using the relations

$$w(-z) = 2e^{-z^2} - w(z)$$

$$w(\overline{z}) = \overline{w(-z)}$$

it is possible to obtain $w(z)$ for values of z over the entire complex plan using the values of $w(z)$ for z in the first quadrant.

Numerous techniques for computation of the complex error function have been proposed. See ?) for several references.

To successfully enclose the result of a computation, it is necessary to use techniques which offer strict bounds on truncation errors. Fortunately, such bounds exist for many complex continued fractions.

Let

$$f(z) = \frac{\sqrt{\pi} e^{z^2}}{z} erfc(z) = \frac{2e^{z^2}}{z} \int_z^\infty e^{-t^2} dt$$

.

From results in ?), it is possible to represent $f(z)$ for all z with $Re(z) > 0$ by the continued fraction

$$f(z) = \frac{1}{z} \left(\frac{1}{z+} \frac{1/2}{z+} \frac{1}{z+} \frac{3/2}{z+} \frac{2}{z+} \frac{5/2}{z+} \cdots \right)$$

.

Let $f_n(z), n = 1, 2, 3, \ldots$ represent the convergents of the continued fraction and let $f(z)$ be the limit of this sequence. ?) show that the continued fraction satisfies certain necessary conditions so that

$$|f(z) - f_n(z)| \leq \begin{cases} |f_n(z) - f_{n-1}(z)| & , |\arg z| \leq \pi/2 \\ \sec(|\arg(z)| - \pi/2)|f_n(z) - f_{n-1}(z)| & , \pi/2 < |\arg z| < \pi \end{cases}$$

.

Since we are considering values of z in the first quadrant, once our algorithm stops at the n^{th} iteration, the box $f_n(z)$ can be expanded by the amount $r = |f_n(z) - f_{n-1}(z)|$ in both the positive and negative directions of the real and imaginary axes to obtain a guaranteed enclosure for $f(z)$.

Implementation Details

For complex numbers $a_1, a_2, \ldots, b_1, b_2, \ldots$ we use the notation

$$\frac{a_1}{b_1+}\frac{a_2}{b_2+}\frac{a_3}{b_3+}\cdots$$

as a shorthand for the continued fraction

$$\cfrac{a_1}{b_1 + \cfrac{a_2}{b_2 + \cfrac{a_3}{b_3 + \cdots}}}$$

When the continued fraction is of finite length n, its value is denoted by f_n, the n^{th} convergent.

One method of evaluating continued fractions in the forward direction is through the use of the system of second-order linear difference equations

$$A_{-1} = 1, A_0 = 0, B_{-1} = 0, B_0 = 1,$$
$$A_n = b_n A_{n-1} + a_n A_{n-2}, n = 1, 2, 3, \ldots$$
$$B_n = b_n B_{n-1} + a_n B_{n-2}, n = 1, 2, 3, \ldots$$

From these, the convergents may be calculated as $f_n = A_n/B_n$. Forward evaluation of continued fractions has the advantage that the stopping value of n need not be determined in advance. The backward recurrence algorithm (not defined here) requires determination of n before calculations proceed. According to ?), the backwards recurrence algorithm may be preferred for computational purposes since the rounding error in calculating f_n is either bounded or else grows very slowly.

Not as good as forward. Contraction. Rescaling.

Future work

?) descibe a method which improves upon that of ?) to a suggested relative accuracy of 14 digits.

Bibliography