Programming semantics in the presence of complex numbers, logarithms etc.

James Davenport
University of Bath
J.H.Davenport@bath.ac.uk

23 May 2012

Conventional Wisdom

Programming errors in numerical programs come in three distinct flavours

blunder This is the sort of error traditionally addressed in “program verification”: are all array elements properly initialised before use, are array bounds always respected etc.

parallelism Issues of deadlocks or races occurring due to the parallelism of an otherwise correct sequential program.

numerical Do truncation and round-off errors, individually or combined, mean that the program computes approximations to the “true” answers which are out of tolerance. This is the area traditionally addressed in Numerical Analysis.

We note that [Cou05] contains 30 papers, of which only [Mar05] deals with strictly numerical issues, four with parallelism issues, and the rest (83%) with the first kind.

Numerical Analysis

Itself a very large and complicated subject

There are really two subquestions here:

- the rounding question, i.e. does $\mathbf{R}_{IEEE}$ approximate $\mathbf{R}$ sufficiently well,
- and the truncation error question, i.e. is h small enough that it is the mathematical ϵ.

Unfortunately the two interact!

Our thesis

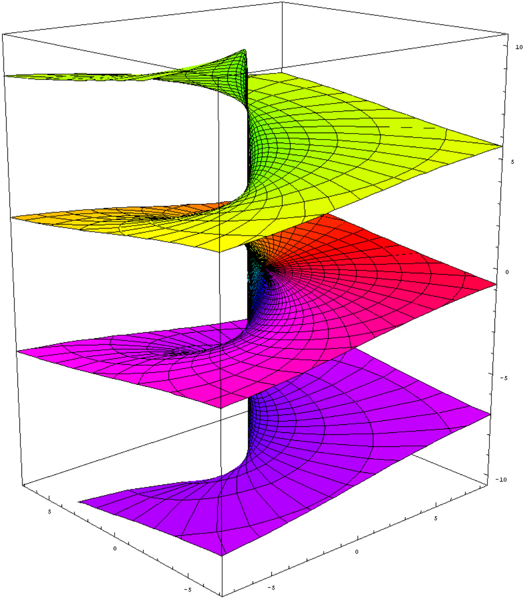
There is a fourth category

manipulation A piece of algebra, which is "obviously correct", turns out not to be.

Note: throughout this paper we take the standard definitions of the branch cuts of the elementary functions from [AS64, Nat10, as tightened in [CDJW00]]. Other definitions would have different, but not fewer, problems.
Initial capitals, such as $\mathrm{Log}\, z$ denote the multivalued functions, i.e.

$$\mathrm{Log}\, z = \{\log z + 2ni\pi | n \in \mathbf{Z}\}$$

An example: log



Generalities

The problems we are going to describe arise largely from complex numbers, and it is sometimes said “real programs don’t use complex numbers”.

- Many people, such as [Ter12, (5.2.5)], have been misled by

  $$\mathrm{Arctan}(x) + \mathrm{Arctan}(y) = \mathrm{Arctan}\left(\frac{x+y}{1-xy}\right)$$

  [AS64, (4.4.34)]

  into writing

  $$\arctan(x) + \arctan(y) = \arctan\left(\frac{x+y}{1-xy}\right),$$

  which is not universally valid (consider $x = y = 2$).
- Many problems arise in fluid dynamics, where two-dimensional real space $\mathbf{R}^2 = \{(x,y)\}$ is viewed as the complex plane $\mathbf{C} = \{z = x + iy\}$, then mapped analytically

a simple case

Before looking at genuine problems, consider a simple example

true $\sqrt{1-z}\sqrt{1+z} = \sqrt{1-z^2}$ (A)

and the r.h.s. is half the cost

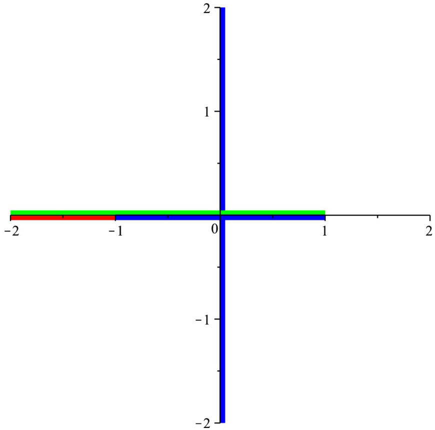
(and vectorises better)

false $\sqrt{z-1}\sqrt{z+1} \overset{?}{=} \sqrt{z^2-1}$ (B)

consider $z=-2$: $\sqrt{-3}\sqrt{-1} \overset{?}{=} \sqrt{3}$

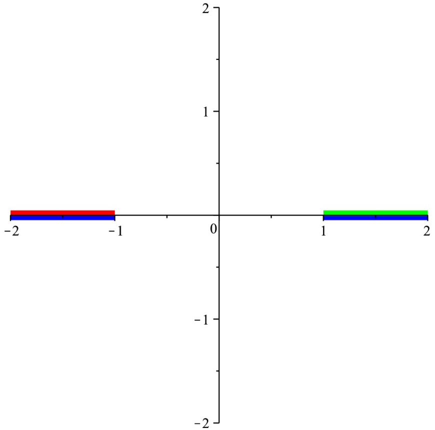
The difference is that the branch cuts in (B) divide the complex plane, essentially into a region of truth and a region of "well, it's true if the r.h.s. is the other square root"

$\sqrt{z-1}\sqrt{z+1} \overset{?}{=} \sqrt{z^2-1}$



The main region is “clearly” disconnected, so there *might* be generic areas of falsity, as well as problems on the branch cuts

$\sqrt{1-z}\sqrt{1+z}\overset{?}{=}\sqrt{1-z^2}$



The main region is “clearly” connected, so if there are problems, they are only on the cuts

Kahan's example: [Kah87, pp. 187–189]

$$w = g(z) := 2\,\mathrm{arccosh}\left(1+\frac{2z}{3}\right) - \mathrm{arccosh}\left(\frac{5z+12}{3(z+4)}\right) \qquad (1)$$

is only the same as the ostensibly more efficient

$$w \overset{?}{=} q(z) := 2\,\mathrm{arccosh}\left(2(z+3)\sqrt{\frac{z+3}{27(z+4)}}\right), \qquad (2)$$

if we avoid the negative real axis and the area

$$\left\{ z = x+iy : |y| \le \sqrt{\frac{(x+3)^2(-2x-9)}{2x+5}} \wedge -9/2 \le x \le -3 \right\} \quad (3)$$

Modern computer algebra systems will refuse to convert one into the other, but this does not constitute a proof of difference.

Challenge

Demonstrate automatically that g and q are not equal, by producing a z at which they give different results.

There is a solution, but

The first truly algorithmic approach is ten years old ([BCD$^+$02], refined in [BBDP07]), and has various difficulties.

- We use Cylindrical Algebraic Decomposition of $\mathbf{R}^N$ to find the connected components of $\mathbf{C}^{N/2} \setminus \{\text{branch cuts}\}$.
- The complexity of this is doubly exponential in N: $\leq d^{O(2^N)}$ [Hon91] and $\geq 2^{2^{(N-1)/3}}$ [BD07, DH88].
- Better algorithms are in principle known ([BRSEDS12] is $d^{O(N\sqrt{N})}$), we do not know of any accessible implementations.
- We are clearly limited to small values of N, at which point $O(\ldots)$ is of limited use. The cross-over point between $2^{(N-1)/3}$ and $N\sqrt{N}$ is at $N = 21$
- A more detailed comparison is given in [Hon91].

but (2)

- While the fundamental branch cut of log is simple enough, being $\{z = x + iy | y = 0 \wedge x < 0\}$, actual branch cuts are messier
- Part of the branch cut of (2) is

$$2x^3+21x^2+72x+2xy^2+5y^2+81 = 0 \wedge \text{other conditions}, \quad (4)$$

  whose solution accounts for the curious expression in (3).
- While there has been some progress in manipulating such images of half-lines (described in [PBD10, Phi11]), there is almost certainly more to be done

Note This would also be of interest for motion-planning

Injectivity

Consider the Joukowski map [Hen74, pp. 294–298]:

$$f : z \mapsto \frac{1}{2}\left(z + \frac{1}{z}\right). \tag{5}$$

Lemma

f is injective as a function from $D = \{z : |z| > 1\}$,

"Proof": If $z \mapsto \zeta$ then $1/z \mapsto \zeta$, and there are no other pre-images of ζ. If $|z| > 1$, then $|1/z| < 1$, so z is unique in D. In fact f is a bijection from D to $\mathbf{C} \setminus [-1, 1]$, and hence has an inverse.

More formally

(5) is the conformal map $\mathbf{C} \to \mathbf{C}$ that equates to the map

$$f_R : (x, y) \mapsto \left(\frac{1}{2} x + \frac{1}{2} \frac{x}{x^2+y^2}, \frac{1}{2} y - \frac{1}{2} \frac{y}{x^2+y^2} \right) \qquad (6)$$

$\mathbf{R}^2 \to \mathbf{R}^2$. However, it is not obvious from (6) alone that f_R is a bijection, i.e. that

$$\forall x_1 x_2 y_1 y_2 \quad \left(x_1^2 + y_1^2 > 1 \wedge x_2^2 + y_2^2 > 1 \wedge \ x_1 + \tfrac{x_1}{x_1^2+y_1^2} = x_2 + \tfrac{x_2}{x_2^2+y_2^2} \wedge \right.$$
$$\left. y_1 - \tfrac{y_1}{x_1^2+y_1^2} = y_2 - \tfrac{y_2}{x_2^2+y_2^2} \right) \Rightarrow (x_1 = x_2 \wedge y_1 = y_2). \qquad (7)$$

Challenge

Demonstrate automatically the truth of (7).

Towards this challenge

We have been unable to do this with either the QEPCAD [Bro03] of Partial Cylindrical Algebraic Decomposition [CH91] or the Maple implementation of Cylindrical Algebraic Decomposition via triangular decomposition [CMMXY09].

However, Brown [Bro12] has been able to reformulate the problem to make it amenable to QEPCAD, and indeed solved it in under 12 seconds.

Challenge

Automate these techniques and transforms.

Why so difficult?

The lemma seems to be about complex functions of one variable, so why do we need to handle (or fail to handle) statements about four real variables to prove them?

1) The statements require the $|\cdot|$ function which is not complex analytic. Hence some recourse to real analysis (and therefore twice as many variables) seems inevitable, though it would be nice to have a more formal statement and proof of this.
2) Equation (7) is the direct translation of the basic definition of injectivity. In practice, certainly if we were looking at functions $\mathbf{R} \to \mathbf{R}$, we would want to use the fact that the function concerned was continuous.

Challenge

Find a better formulation of injectivity questions $\mathbf{R}^N \to \mathbf{R}^N$, making use of the properties of the functions concerned (certainly continuity, possibly rationality).

Why so difficult (2)?

3) While (7) is from the existential theory of the reals, and so the theoretically more efficient algorithms quoted in [Hon91] are in principle applicable, the more modern developments described in [PJ09] do not seem to be directly applicable. However, we can transform then into a disjunction of statements to each of which the Weak Positivstellensatz [PJ09, Theorem 1] is applicable.

Challenge

Solve these problems using the techniques of [PJ09],

A bijection has an inverse, right!

So what is it?

Formally: if $\zeta = \frac{1}{2}\left(z + \frac{1}{z}\right)$, then $2z\zeta = z^2 + 1$ and $z = \zeta \pm \sqrt{\zeta^2 - 1}$, and the only challenge is: which ± to choose?

The answer is "neither", or at least "neither, uniformly".

For f a bijection from $\{z : |z| > 1\}$ to $\mathbf{C} \setminus [-1, 1]$, its inverse is

$$f_1(\zeta) = \zeta \begin{cases} +\sqrt{\zeta^2 - 1} & \Im(\zeta) > 0 \\ -\sqrt{\zeta^2 - 1} & \Im\zeta) < 0 \\ +\sqrt{\zeta^2 - 1} & \Im(\zeta) = 0 \wedge \Re(\zeta) > 1 \\ -\sqrt{\zeta^2 - 1} & \Im(\zeta) = 0 \wedge \Re(\zeta) < -1 \end{cases} \tag{8}$$

In fact, a better (at least, free of case distinctions) definition is

$$f_2(\zeta) = \zeta + \sqrt{\zeta - 1}\sqrt{\zeta + 1}. \tag{9}$$

The techniques of [BBDP07] are able to **verify** (9)

Challenge

Derive automatically, and prove, either (8) or (9).

Worse

Lemma

f is injective as a function from $H = \{z : \Im z > 0\}$.

"proof" is the same, and we don't have a formal proof, which needs $\mathbf{R}^4$ ($\Im$ is not complex analytic).
[Hen74, (5.1-13), p. 298] argues for the inverse

$$f_2(\zeta) = \zeta + \underbrace{\sqrt{\zeta - 1}}_{\arg\in(-\pi/2,\pi/2]} \underbrace{\sqrt{\zeta + 1}}_{\arg\in(0,\pi]}. \tag{10}$$

Challenge

Find a way to represent functions such as $\underbrace{\sqrt{\zeta + 1}}_{\arg\in(0,\pi]}$

Worse (2)

Fortunately such bizarre functions are expressible in this case, we can write $\underbrace{\sqrt{\zeta+1}}_{\arg\in(0,\pi]} = i \underbrace{\sqrt{-\zeta-1}}_{\arg\in(-\pi/2,\pi/2]}$, and the latter is the normal `sqrt` function of [AS64]. Hence we have an inverse function

$$f_3(\zeta) = \zeta + \sqrt{\zeta-1}\, i\sqrt{-\zeta-1}. \tag{11}$$

Challenge

Demonstrate automatically that this is an inverse to f on $\{z : \Im z > 0\}$.

A Higher-level challenge

Most of what we have been talking about is represented in "normal" texts such as [Hen74], and a programmer implementing this would write comments (we hope!).

How to turn these into any sort of machine-readable specification?

This may be related to the more general question of capture of side-conditions, see Ariane-V.

Bibliography

For the full bibliography, see `http://staff.bath.ac.uk/masjhd/Slides/JHD-Wessex-bib.pdf`.

M. Abramowitz and I. Stegun.

Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables, 9th printing.
US Government Printing Office, 1964.

J.C. Beaumont, R.J. Bradford, J.H. Davenport, and N. Phisanbut.

Testing Elementary Function Identities Using CAD.
AAECC, 18:513–543, 2007.

R.J. Bradford, R.M. Corless, J.H. Davenport, D.J. Jeffrey, and S.M. Watt.

Reasoning about the Elementary Functions of Complex Analysis.
Annals of Mathematics and Artificial Intelligence, 36:303–318, 2002.

C.W. Brown and J.H. Davenport.

The Complexity of Quantifier Elimination and Cylindrical Algebraic Decomposition.
In C.W. Brown, editor, *Proceedings ISSAC 2007*, pages 54–60, 2007.

C.W. Brown.

QEPCAD B: a program for computing with semi-algebraic sets using CADs.
ACM SIGSAM Bulletin 4, 37:97–108, 2003.

C.W. Brown.

Re: Query about QEPCAD.
Personal Commnication to David Wilson, 2012.

S. Basu, M.-F. Roy, M. Safey El Din, and É. Schost.