PART II: PROPOSED RESEARCH AND CONTEXT

1 Background — Computer Algebra

The elementary functions are traditionally thought of as log, exp and the trigonometric and hyperbolic functions (and their inverses). This should include powering (to non-integral powers) and also the n-th root. These functions are built in, to a greater or lesser extent, into many computer algebra systems (not to mention other programming languages [14, 17]), and are heavily used. However, reasoning with them is more difficult than is usually acknowledged, and all algebra systems have one, sometimes both[1], of the following defects:

- they make mistakes, be it the traditional schoolchild one
$$1 = \sqrt{1} = \sqrt{(-1)^2} = -1 \tag{1}$$
or more subtle ones — see [18];
- they fail to perform obvious simplifications, leaving the user with an impossible mess when there "ought" to be a simpler answer. In fact, there are two possibilities here: maybe there is a simpler equivalent that the system has failed to find, but maybe there isn't, and the simplification that the user wants is not actually valid, or is only valid outside an exceptional set. In general, the user is not informed what the simplification might have been, nor what the exceptional set is: it may be a few points, some lines in the complex plane[2], or significant[3] regions of the complex plane.

It is often thought that these problems are limited to complex numbers, and that algebra with the reals (as opposed to the entirety of **C**) is safe from these difficulties. However, this is not true, for two reasons: one is that complex numbers can be introduced via operations on reals, as in equation (1) or $\arccos(2) \approx 1.317i$, and the other is that well-known "equations" may be invalid even on the reals, as in

$$\arctan x + \arctan y \overset{?}{=} \arctan\left(\frac{x+y}{1-xy}\right) \tag{2}$$

(the two sides differ by $\pm\pi$ if $xy \geq 1$).[4] However, the problems are more manifest on the complexes, and a proper solution needs complex analysis. For example,

$$\log \overline{z} \neq \overline{\log z} \tag{3}$$

on the branch cut for log: instead $\log \overline{z} = \overline{\log z} + 2\pi i$ on the cut. Similarly,

$$\log(\frac{1}{z}) \neq -\log z \tag{4}$$

on the branch cut: instead $\log(\frac{1}{z}) = -\log z + 2\pi i$ on the cut, and we even have lack of continuity:

$$\lim_{y\to 0^-} \log(x+iy) \neq \log(x). \tag{5}$$

Various solutions have been expressed to the problem of branch cuts and multi-valued functions.

1. [16] points out that the concept of a "signed zero" [13] (for clarity, we write the positive zero as 0^+ and the negative one as 0^-) can be used to solve the above problems, if we say that, for $x < 0$, $\log(x + 0^+ i) = \log(x) + \pi i$ whereas $\log(x + 0^- i) = \log(x) - \pi i$. Equation (5) then becomes an equality for all x, interpreting the x on the right as $x + 0^- i$. Similarly, (3) and (4) become equalities throughout. Attractive though this proposal is, it does not answer the fundamental question as far as the designer of a computer algebra system is concerned: what to do if the user types $\log(-1)$.

[1]For example, Maple's `simplify` without options has the second, whereas `simplify(...,symbolic)` has the first.

[2]As is $\log \frac{1}{x} = -\log x$ *except* on the negative real axis

[3]See the example in [16].

[4]This is a counterexample to the belief that many of those equalities you thought were true *are* actually true if you restrict your attention to the positive reals.

2. [2] and [8] point out that most "equalities" do not hold for the complex logarithm, e.g. $\log(z^2) \neq 2\log z$ (try $z=-1$), and its generalisation

$$\log(z_1 z_2) \neq \log(z_1) + \log(z_2). \tag{6}$$

The most fundamental of all non-equalities is $z = \log\exp z$, whose most obvious violation is at $z = 2\pi i$. The authors of [8] therefore propose to introduce the *unwinding number* $\mathcal{K}$, defined[5] by

$$\mathcal{K}(z) = \frac{z - \log\exp z}{2\pi i} = \left\lceil \frac{\Im z - \pi}{2\pi} \right\rceil \in \mathbf{Z} \tag{7}$$

(note that the apparently equivalent definition $\lfloor \frac{\Im z+\pi}{2\pi} \rfloor$ differs precisely on the branch cut for log as applied to $\exp z$). (6) can then be rescued as

$$\log(z_1 z_2) = \log(z_1) + \log(z_2) - 2\pi i \mathcal{K}(\log(z_1) + \log(z_2)). \tag{8}$$

Similarly (3) can be rescued as

$$\log \overline{z} = \overline{\log z} - 2\pi i \mathcal{K}(\overline{\log z}). \tag{9}$$

Note that, as part of the algebra of $\mathcal{K}$, $\mathcal{K}(\overline{\log z}) = \mathcal{K}(-\log z) \neq \mathcal{K}(\log \frac{1}{z})$. $\mathcal{K}(z)$ depends only on the imaginary part of z.

3. Although not formally proposed in the same way in the computational community, one possible solution, often found in texts in complex analysis, is to accept the multi-valued nature of these functions (we adopt the common convention of using capital letters, e.g. Ln, to denote the multi-valued function), defining, for example

$$\operatorname{Arcsin} z = \{y \mid \sin y = z\}.$$

This leads to $\operatorname{Sqrt} z^2 = \{\pm z\}$, which has the advantage that it is valid throughout $\mathbf{C}$. Equation (6) is then rewritten as

$$\operatorname{Ln}(z_1 z_2) = \operatorname{Ln}(z_1) + \operatorname{Ln}(z_2), \tag{10}$$

where addition is addition of sets ($A + B = \{a + b : a \in A, b \in B\}$) and equality is set equality[6].

However, it seems to lead in practice to very large and confusing formulae. More fundamentally, this approach does not say what will happen when the multi-valued functions are replaced by the single-valued ones of numerical programming languages.

A further problem that has not been stressed in the past is that this approach suffers from the same aliasing problem that naïve interval arithmetic does [9]. For example,

$$\operatorname{Ln}(z^2) = \operatorname{Ln}(z) + \operatorname{Ln}(z) \neq 2\operatorname{Ln}(z),$$

since $2\operatorname{Ln}(z) = \{2\ln(z) + 4k\pi i : k \in \mathbf{Z}\}$, but $\operatorname{Ln}(z) + \operatorname{Ln}(z) = \{2\ln(z) + 2k\pi i : k \in \mathbf{Z}\}$: indeed if $z=-1$, $\ln(z^2) \notin 2\operatorname{Ln}(z)$. Hence this method is unduly pessimistic: it may fail to prove some identities that are true.

2 Background — Cylindrical Algebraic Decomposition

Cylindrical Algebraic Decomposition [5] was invented as an effective method of solving algebraic problems of quantifier elimination over the reals. Given a statement such as $\forall x : x^5 + ax^4 + bx^3 + cx^2 + dx + e > 0$, cylindrical algebraic decomposition yields a semi-algebraic (that is, involving polynomials $\gtreqless 0$ with Boolean connectives) quantifier-free formula stating where the original is valid. A major problem of the cylindrical algebraic

[5]Note that the sign convention here is the opposite to that of [8], which defined $\mathcal{K}(z)$ as $\lfloor \frac{\pi - \Im z}{2\pi} \rfloor$: the authors of [8] recanted later to keep the number of -1s occurring in formulae to a minimum.

[6]"The equation merely states that the sum of one of the (infinitely many) logarithms of z_1 and one of the (infinitely many) logarithms of z_2 can be found among the (infinitely many) logarithms of $z_1 z_2$, and conversely every logarithm of $z_1 z_2$ can be represented as a sum of this kind (with a suitable choice of $\operatorname{Ln} z_1$ and $\operatorname{Ln} z_2$)." [4, pp. 259–260] (our notation).

decomposition method is that it is *too* powerful: given a quantified semi-algebraic formula, it produces a decomposition of $\mathbf{R}^n$ which answers, not merely the question asked, but also all other potential questions (with the same order of quantifiers) involving the same formulae. In particular, given $\forall \ldots (y^2 = 2 \wedge \ldots$, it will also analyse the case $y^2 < 2$ and the two cases ($y > \sqrt{2}$ and $y < -\sqrt{2}$) of $y^2 > 2$.

A further problem, more one of the algorithm than of the general methodology, is that in the process it constructs auxiliary formulae (such as $x = 0$ in the second example of the next section), and the decomposition relates to these as well as the original formulae. Clustering [1] is a retrospective technique for reducing the expression swell caused by this peculiarity of the algorithm.

3 Programme & Methodology

The fundamental aim of this project is to explore the approach of method (2) for handling analytic functions, as an *algebro-geometric* method of resolving many of these problems.

For definiteness, we will use the definitions in [7], though it is worth pointing out that one major application of the technology to be developed in this project would consist in providing the ability to explore the definitional variants, as in

$$\underbrace{\operatorname{arccot}}_{\text{Maple}} z = \frac{\pi}{2} - \overline{\underbrace{\arctan}_{\text{Derive}} \overline{z}}.$$

3.1 Theory

Since $\mathcal{K}$ is an integer-valued function of a complex variable, $\mathcal{K}(f(z))$ can only change when one of two things happens:

1. $\Im(f(z))$ crosses a multiple of 2π. This involves a (potentially infinite) number of lines in $f(z)$ space. For many f encountered in practice, this inverts to a family of algebraic lines in z space.
2. $f(z)$ itself is discontinuous, which involves recursion on the definition of f.

Examples explored by the investigators and their colleagues in practice (notably in the writing of [3, 6, 7]) show that one is forced to consider the geometry of $\mathbf{C}^k$, where k is the number of variables involved in the putative identity. We give two examples of this.

1. [7] proves, by hand by using the unwinding number formulation, that two alternative definitions of arccos are in fact equal.

   Theorem 1
   $$\frac{2}{i} \ln \left(\sqrt{\frac{1+z}{2}} + i\sqrt{\frac{1-z}{2}} \right) = -i \ln \left(z + i\sqrt{1-z^2} \right). \tag{11}$$

   A key step in the proof was proving the correct (and therefore containing unwinding numbers) version of $\sqrt{z_1 z_2} \overset{?}{=} \sqrt{z_1}\sqrt{z_2}$.

   Lemma 1
   $$\sqrt{z_1 z_2} = \sqrt{z_1}\sqrt{z_2}(-1)^{\mathcal{K}(\ln(z_1)+\ln(z_2))}. \tag{12}$$

   From this they deduced the following.

   Lemma 2 *Whatever the value of z,*
   $$\sqrt{1-z}\sqrt{1+z} = \sqrt{1-z^2}.$$

An alternative approach is given in [3], where the various branch cuts on which the two sides might differ are analysed. Let us reformulate the problem as that of proving that

$$2\ln\left(\sqrt{\frac{1+z}{2}}+i\sqrt{\frac{1-z}{2}}\right)-\ln\left(z+i\sqrt{1-z^2}\right) \tag{13}$$

is identically zero.

If we differentiate (13), we get

$$\frac{2\left(\frac{1}{\sqrt{2+2\,z}}-\frac{i}{\sqrt{2-2\,z}}\right)}{\left(\sqrt{2+2\,z}+i\sqrt{2-2\,z}\right)}-\frac{1-\frac{iz}{\sqrt{1-z^2}}}{z+i\sqrt{1-z^2}}, \tag{14}$$

which, with the help of Lemma 2, simplifies to zero. Hence (13) is a *local* constant, i.e. its value only changes at jump discontinuities, i.e. branch cuts. What are the branch cuts of (13)? It turns out that, writing $z = x + iy$, the branch cuts all have $y = 0$, so we shall omit this. For the first term, the branch cuts are at most $[\sqrt{2},\infty)$ for the logarithm itself, and $(-\infty,-1)$ and $(1,\infty)$ for the square roots. For the second term, we have $(-\infty,-1]$ for the logarithm, and the same cuts as before, though for different reasons, for the square root.

From the point of view of cylindrical algebraic decomposition, we have to decompose the plane by $y = 0$, $x = -1$, $x = 1$ and $x = \sqrt{2}$ (in the original formulation of [5],we would need $x^2 = 2$, but we ignore this here). This generates eight two-dimensional regions, ten one-dimensional regions and three zero-dimensional ones. Clustering [1] would not of itself make a significant improvement here, since the regions all differ in the signs of some of the polynomials. The important fact is that, for example, $x = \sqrt{2}$ is only *relevant* when $y = 0$. Taking this into account, a new theme in (semi-)algebraic decomposition, should reduce this decomposition to one two-dimensional region, three one-dimensional ones and three zero-dimensional ones: a significant gain.

2. If we consider, this time just for real x and y, the often-quoted identity

$$\arctan x + \arctan y \overset{?}{=} \arctan\left(\frac{x+y}{1-xy}\right), \tag{15}$$

we see (and this can be justified by branch-cut reasoning: arctan has a branch cut at infinity) that the case $xy = 1$ is critical, and divides the plane into three two-dimensional regions, to which the dividing curves (the two components of $xy - 1$) can be attached to give three semi-algebraic regions, on which the identity is respectively:

- false, needing $-\pi$ on the right-hand side to make it true;
- true;
- false, needing $+\pi$ on the right-hand side to make it true.

Application of cylindrical algebraic decomposition [5] to $xy - 1$ would divide the plane into seven regions: four two-dimensional and three one-dimensional. There would be three regions corresponding to $xy < 1$: $xy < 1 \wedge x < 0$, $x = 0$ and $xy < 1 \wedge x > 0$. Clustering techniques [1] should reduce this to the three two-dimensional and two one-dimensional regions that are intrinsic to the problem.

3.2 Implementation

The project lies on the borderline of (general-purpose, i.e. including calculus) computer algebra as classically perceived and quantifier elimination. It is a sad fact of life that no single piece of software supports both areas. While the state of the art in both is continually changing, we currently propose to base ourselves on Axiom [15] or Maple for the computer algebra simplification, and RedLog [11] for the quantifier elimination. While communication between software systems dealing in complicated mathematics is never trivial, the fact that both Axiom and Reduce (the underlying system of RedLog) support OpenMath [10] will help considerably.

We propose a two-level attack on the problem. Level 1 is largely at the computer algebra level, and consists in extracting the relevant branch cuts and their equations from proposed identities, pass them "naïvely" to quantifier elimination, and study the consequences for the proposed identity. These could be exploited at several levels, from an equivalent of Maple's `testeq` [12] probabilistic tester that would guarantee to test in every region, to a proof of identity, or a corrected identity, as worked by hand above.

Level 2 would be at the cylindrical decomposition level. It is clear from the examples worked by hand above that direct decomposition ignoring the fact that we are dealing with inequalities and regions is wasteful.

4 Relevance to beneficiaries

1. Users of mathematics, notably computer algebra systems, should no longer be faced with the choice between a very weak simplifier (e.g. Maple's `simplify(...)`) and a strong but often incorrect one (e.g. Maple's `simplify(...,symbolic)`).
2. Designers of computer algebra systems will have the powerful, but correct, algorithms on which to rely.
3. The tools and algorithms developed to analyse the geometry of $\mathbf{C}^n$ and $\mathbf{R}^n$ should be of wider relevance, since the existing cylindrical algebraic decomposition techniques are, as pointed out above, too powerful and general for many problems.

5 Relevance to MathFIT

- We intend to import ideas from theorem proving (quantifier elimination and the associated algorithms) to solving the mathematical problem of simplifying elementary functions.
- This rigorous theory of simplification can then be applied to computer algebra systems, and indeed back into theorem proving.

DISSEMINATION AND EXPLOITATION

The main route of dissemination will be the academic one: conference and journal papers. This project lies on the boundary of computer algebra (whose natural conference is the annual International Symposium on Symbolic and Algebraic Computation — ISSAC) and automated reasoning (whose main conference is the annual Conference on Automated Deduction — CADE). There is also a biannual conference devoted specifically to Artifical Intelligence and Symbolic Computation (AISC). The EU-funded Calculemus project organises an annual series of informal conferences in this area[7], which are co-located with the above. The informal nature of Calculemus would provide a particularly useful vehicle for disseminating the early results of the project, and getting feedback from the community.

RESOURCES

There is a substantial amount of highly mathematical development that is needed to turn the ideas behind the examples of section 3.1 into a practicable algorithm. This is a task that requires a post-doctoral researcher, working under the supervision of the two investigators.

The problem of expressing the geometry of $\mathbf{C}^n$ or $\mathbf{R}^n$ induced by the results of these algorithm is non-trivial. While cylindrical algebraic decomposition can solve it, in many cases this is a very wasteful approach. We are requesting a research student to work on this problem: designing and implementing a

[7]Defined in the Calculemus manifesto as "the design of a new generation of mathematical software systems and computer-aided verification tools based on the integration of the deduction and the computational power of Deduction Systems and Computer Algebra Systems respectively".

variant of cylindrical algebraic decomposition that is capable of handling these sort of constraints, e.g. “the negative real axis” directly.

In order that the research student has appropriate examples and material on which to work, and also the relevant interconnect software systems, we are requesting that the RA start three months before the research student. We are also requesting a standard PC for the RA and student, connected to the campus network, for which the University charges, and consumables. Travel is requested for the obvious conferences in the field.

References

[1] Cylindrical Algebraic Decomposition II: An Adjacency Algorithm for the Plane. *SIAM J. Comp.* **13** (1984) pp. 878–889.

[2] Bradford,R.J., Algebraic Simplification of Multiple-Valued Functions. Proc. DISCO ’92 (Springer Lecture Notes in Computer Science 721, ed. J.P. Fitch), Springer, 1993, pp. 13–21.

[3] Bradford,R.J., Corless,R.M., Davenport,J.H., Jeffrey,D.J. & Watt,S.M., Reasoning about the Elementary Functions of Complex Analysis. Submitted to Annals of Mathematics and Artificial Analysis.

[4] Carathéodory,C., *Theory of functions of a complex variable* (trans. F. Steinhardt), 2nd. ed., Chelsea Publ., New York, 1958.

[5] Collins,G.E., Quantifier Elimination for Real Closed Fields by Cylindrical Algebraic Decomposition. Proc. 2nd. GI Conference Automata Theory & Formal Languages (Springer Lecture Notes in Computer Science 33) pp. 134–183.

[6] Corless,R.M., Davenport,J.H., Jeffrey,D.J., Litt,G. & Watt,S.M., Reasoning about the Elementary Functions of Complex Analysis. Artificial Intelligence and Symbolic Computation (ed. John A. Campbell & Eugenio Roanes-Lozano), Springer Lecture Notes in Artificial Intelligence Vol. 1930, Springer-Verlag 2001, pp. 115–126.

[7] Corless,R.M., Davenport,J.H., Jeffrey,D.J. & Watt,S.M., “According to Abramowitz and Stegun”. SIGSAM Bulletin **34** (2000) 2, pp. 58–65.

[8] Corless,R.M. & Jeffrey,D.J., The Unwinding Number. SIGSAM Bulletin **30** (1996) 2, issue 116, pp. 28–35.

[9] Davenport,J.H. & Fischer,H.-C., Manipulation of Expressions. *Improving Floating-Point Programming* (ed. P.J.L. Wallis), Wiley, 1990, pp. 149–167.

[10] Dewar,M.C., OpenMath: An Overview. *ACM SIGSAM Bulletin* **30** (2000) 2 pp. 2–5.

[11] Dolzmann,A. & Sturm,Th., Redlog User Manual. Report MIP-9616, University of Passau, October 1996. `http://www.fmi.unipassau.de/~redlog`.

[12] Gonnet,G.H. Determining Equivalence of Expressions in Random Polynomial Time. Proc. 16th ACM Symp. Theory of Computing, 1984, pp. 334–341.

[13] IEEE Standard 754 for Binary Floating-Point Arithmetic. IEEE Inc., 1985.

[14] IEEE Standard Pascal Computer Programming Language. IEEE Inc., 1983.

[15] Jenks,R.D. & Sutor,R.S., *AXIOM: The Scientific Computation System.* Springer-Verlag, 1992.

[16] Kahan,W., Branch Cuts for Complex Elementary Functions. *The State of Art in Numerical Analysis* (ed. A. Iserles & M.J.D. Powell), Clarendon Press, Oxford, 1987, pp. 165–211.

[17] Steele,G.L.,Jr., Common LISP: The Language, 2nd. edition. Digital Press, 1990.

[18] Stoutemyer,D.R., Crimes and Misdemeanors in the Computer Algebra Trade. *Notices of the AMS* vol. 38, no. 7, September 1991, pp. 778–785.