Geometry of Branch Cuts

Nalina Phisanbut, Russell J. Bradford & James H. Davenport
Department of Computer Science, University of Bath, UK
{cspnp, rjb, jhd}@cs.bath.ac.uk

Introduction

'Simplification' is a key concept in Computer Algebra. But many simplifications rules, such as $\sqrt{x}\sqrt{y} \to \sqrt{xy}$, are not universally valid, due to the fact that many elementary functions are multi-valued. Hence a key question is *"Is this simplification correct?"*, which involves algorithmic analysis of the branch cuts involved. Here we look at variable ordering and pre-conditioning as supporting technologies for this analysis.

Algorithm

Our verification system to analyse formulae in elementary functions works as follows:

- Calculate all the branch cuts of the proposed identity.
- Decompose $\mathbb{C}$ (or $\mathbb{C}^n$), viewed as $\mathbb{R}^2$ (or $\mathbb{R}^{2n}$), with respect to the branch cuts and find a sample point in each region in $\mathbb{R}^2$ (or $\mathbb{R}^{2n}$) defined by the branch cuts.
- Evaluate the identity on each connected component using the obtained sample point, thereby conclude whether the identity is true or not on that entire region by the *Monodromy theorem*.

The decomposition step is achieved by means of Cylindrical Algebraic Decomposition (CAD), which in this case is the new Maple 14's CAD as opposed to QEPCAD used in our earlier papers.

Variable Order

There are $(2n)!$ possible variable orders and number of cell decomposition depends on which of these orders is used. Bigger problem in more dimensions.

MUNICH2010
35th International Symposium on Symbolic and Algebraic Computation
Munich, 25–28 July 2010

Branch Cuts

Example 1:

$$\sqrt{z-1}\sqrt{z+1} \stackrel{?}{=} \sqrt{z^2-1}, \qquad (1)$$

is *false* for some $z \in \mathbb{C}$.

Example 2:

$$\sqrt{1-z}\sqrt{1+z} \stackrel{?}{=} \sqrt{1-z^2}, \qquad (2)$$

is *true* for all $z \in \mathbb{C}$.

The branch cut for $\sqrt{z}$ is conventionally $\{z \mid \Re(z) < 0 \wedge \Im(z) = 0\}$. Illustrating these examples geometrically shows that although they are similar algebraically, the are very different geometrically.

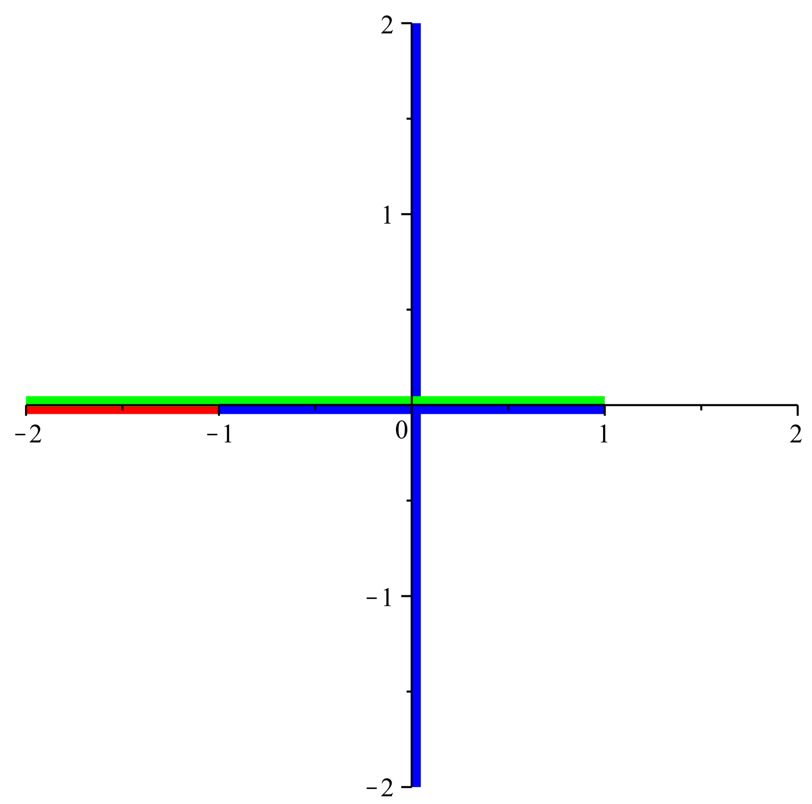

Figure 1: Branch cuts of (1)

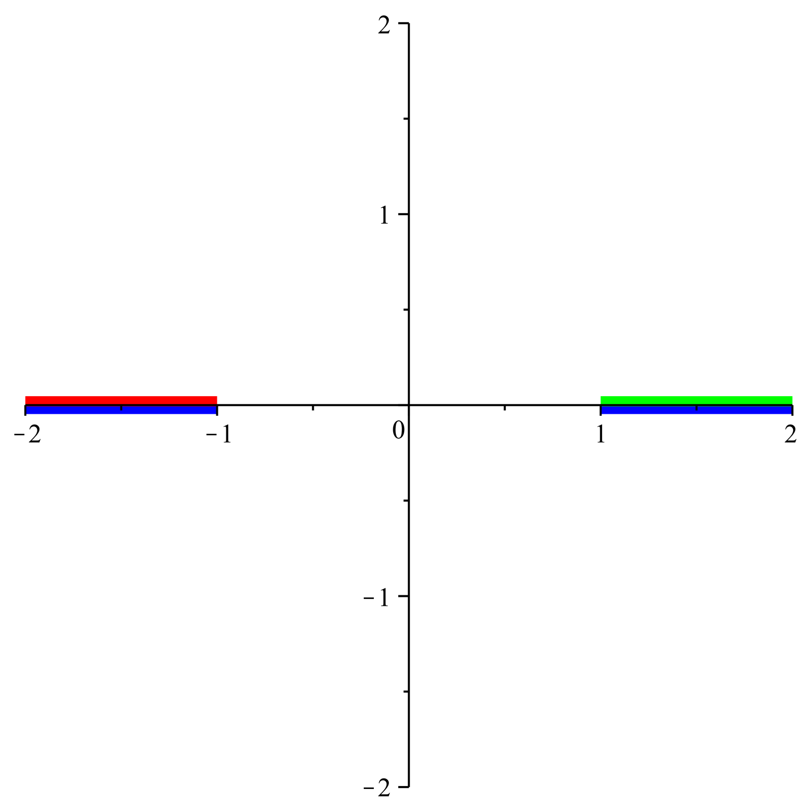

Figure 2: Branch cuts of (2)

Pre-conditioning

Example 1:
QEPCAD Input: Prenex formula

$$[[x-1<0 \wedge y=0] \vee [x+1<0 \wedge y=0] \vee [x^2-y^2-1<0 \wedge xy=0]]$$

Maple 14 Input: A set of polynomials

$$[x-1, y, x+1, y, x^2-y^2-1, xy]$$

`Note:` Redundant y can be removed without altering the result.

Problem: Maple loses information about the branch cuts.
Improvement aim: Allow some linkages between pairs of inequalities and equalities in Maple's CAD sense.
Method: Pseudo-division, either to eliminate x or y.
Result: The two - - - curves are removed.

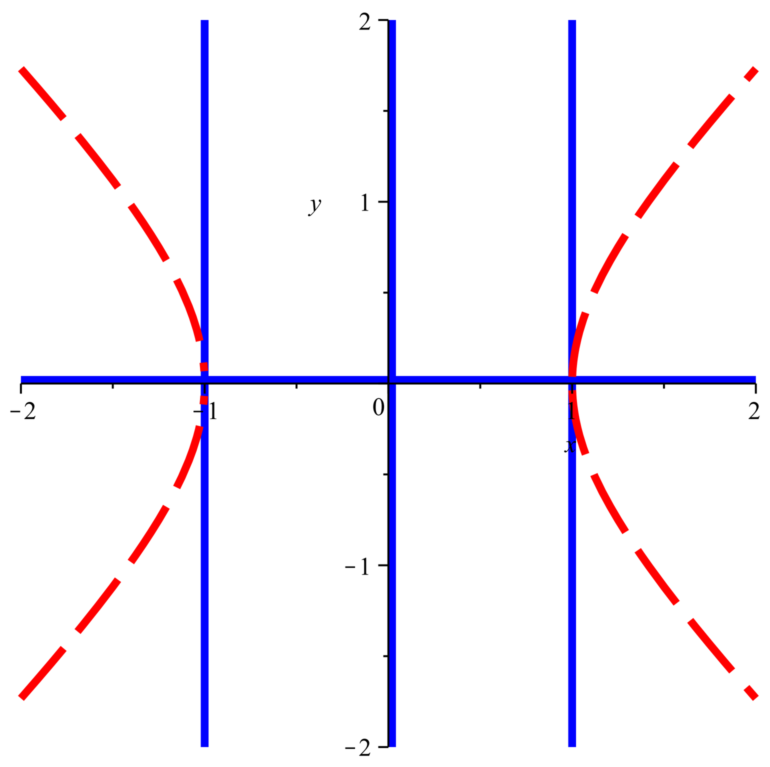

Figure 3: Branch cuts of (1) as viewed by Maple

Table

Example 3:

$$\log(z^3) \stackrel{?}{=} 3\log(z). \qquad (3)$$

Example	Maple		QEPCAD	
	x,y	y,x	x,y	y,x
1 (No elimination)	29	29	36	32
1 (Eliminating x)	21	21	28	24
1 (Eliminating y)	21	21	22	24
2 (No elimination)	29	29	36	32
2 (Eliminating x)	21	21	28	10
2 (Eliminating y)	21	21	13	24
3 (No elimination)	25	25	28	28
3 (Eliminating x)	17	17	20	17
3 (Eliminating y)	25	25	28	28

Table 1: Number of cell decomposition

`Note:` Pre-conditioning to eliminate y in (3) does not have any effect on the set of input polynomials.

Preliminary Results

- CAD via Triangular Decomposition, despite starting from a weaker formulation, is still very competitive with QEPCAD.
- Pre-conditioning the branch cuts often helps in reducing the number of cells produced by CAD. Even QEPCAD can benefit from it.
- Variable order matters, both in elimination and in projection (QEPCAD)/triangularization (Maple), and the interaction is significant and subtle.
- Unlike QEPCAD which is able to exploit the symmetry of the variables, Maple's CAD cannot.
- The minimal *cylindrical* algebraic decomposition may be larger than the optimal algebraic decomposition (the true branch cuts).