Analysis versus Algebra in Symbolic Computation

Robert M. Corless
University of Western Ontario, Canada

30th Applications of Computer Algebra - ACA 2025

The search for formulaic answers via algebra has an ancient history. That such formulæ may have lacunæ where they do not apply was perhaps first noted explicitly by Cauchy in his 1821 *Cours d'Analyse* (English translation presented at the MacTutor site):

> We must even note that they suggest that algebraic formulas have an unlimited generality, whereas in fact the majority of these formulas are valid only under certain conditions and for certain values of the quantities they contain. By determining these conditions and these values, and by fixing precisely the sense of all the notations I use, I make all uncertainty disappear.

But battle was joined anew when digital computers arrived on the scene. The early generations of software performed pretty well all transformations taking an "algebraic" approach and not considering "analytic" issues[1]. This had consequences: the tension between algebra and analysis continues to this day and many current algebra systems will still sometimes give incomplete, misleading, or flatly incorrect answers to various questions. In this talk I will describe some of the history of how this battle has unfolded in the symbolic computation community. Some good progress has been made, and some of today's algorithms and implementations are genuinely better than most of those of thirty years ago.

[1] I'm not being precise, here. Roughly what I intend to convey by "algebra" versus "analysis" is that algebraic models of computation typically have a different notion of continuity than do analytic models of computation, if they consider continuity at all.