Multiprecision: solving and causing problems

Danielle Brake - University of Wisconsin - Eau Claire

Systems of polynomials are not uniformly challenging to solve. Some are quite easy, others quite hard; and, what a statement of difficulty means depends on context including the solution method and output requirements. Numerical methods of solving systems typically produce inexact solutions, in the form of approximations using floating point numbers and a method of refining to arbitrary accuracy. Numerical algebraic geometry or numerical nonlinear algebra, uses probability-one arguments to theoretically justify algorithm aspects, such as choosing a random complex number for the infamous gamma trick. However, the floating point numbers are discrete and finite in number -- there is no such thing as a probability-zero set in them! Furthermore, high condition numbers make some problems impossible to solve with built-in numeric types, such as 64-bit double. A cure is multiprecision, and it comes with a cost -- and not just one in terms of longer time to compute. There are additional challenges to implementation of adaptive precision algorithms, including storage choices, type conversions, the spectre of mixed precision arithmetic, and the attracting wormhole of spaghetti code that awaits those who need to vary numeric type.