Accurate and efficient expression evaluation and linear algebra, or
Why it can be easier to compute accurate eigenvalues of a Vandermonde matrix than the accurate sum of 3 numbers

James Demmel

June 10, 2011

We survey and unify recent results on the existence of accurate algorithms for evaluating multivariate polynomials, and more generally for accurate numerical linear algebra with structured matrices. By "accurate" we mean the computed answer has relative error less than 1, i.e. has some leading digits correct. We also address efficiency, by which we mean algorithms that run in polynomial time in the size of the input. Our results will depend strongly on the model of arithmetic: Most of our results will use what we call the *Traditional Model (TM)*, that the computed result of $op(a, b)$, a binary operation like $a + b$, is given by $op(a, b) * (1 + \delta)$ where all we know is that $|\delta| \leq \varepsilon \ll 1$. Here ε is a constant also known as machine epsilon.

We will see a common reason that the following disparate problems all permit accurate and efficient algorithms using only the four basic arithmetic operations: finding the eigenvalues of a suitably discretized scalar elliptic PDE, finding eigenvalues of arbitrary products, inverses, or Schur complements of totally nonnegative matrices (such as Cauchy and Vandermonde), and evaluating the Motzkin polynomial. Furthermore, in all these cases the high accuracy is "deserved," i.e. the answer is determined much more accurately by the data than the conventional condition number would suggest.

In contrast, we will see that evaluating even the simple polynomial $x+y+z$ accurately is impossible in the TM, using only the basic arithmetic operations. We give a set of necessary and sufficient conditions to decide whether a high accuracy algorithm exists in the TM, and describe progress toward a decision procedure that will take any problem and either provide a high accuracy algorithm or a proof that none exists.

When no accurate algorithm exists in the TM, it is natural to extend the set of available accurate operations $op()$ by a library of additional operations, such as $x+y+z$, dot products, or indeed any enumerable set which could then be used to build further accurate algorithms. We show how our accurate algorithms and decision procedure for finding them can be extended in this case.

Finally, we address other models of arithmetic, and address the relationship between (im)possibility in the TM with (in)efficient algorithms operating on numbers represented as bit strings.