Inverse Symbolic Calculator Two

Overview The project intends to provide an enhanced parallel implementation of a much updated version of the Inverse Symbolic Calculator presently available at http://oldweb.cecm.sfu.ca/projects/ISC/. The ISC and the identify function in Maple (or Recognize in Mathematica) both attempt to provide symbolic form from numerical input. For example

> identify(3.146264369941972); returns $\sqrt{3} + \sqrt{2}$.

This highly-used interface is based on an earlier book


The ISC is now over a decade old and sits in a very aged version of Solaris which will soon be un-maintainable. Moreover, many of the advances the Symbolic Analysis project made before embedding the 'identify function' in Maple have not been added; nor have improvements in Integer relation algorithms been incorporated. Additionally, many user interface issues were not well addressed in the current version.

Team The project will be overseen by JM Borwein (Dalhousie FCS) and DH Bailey (Lawrence Berkeley Labs). Other members include Dante Manna (AARMS PDF), Peter Ye (PhD student, Dalhousie FCS), several students working on NSERC Undergraduate Research Fellowships (May 1- Aug 31, still to be finally identified), and Thomas Daske (Munich exchange student, July 1-Sept 30).

Goals The current project intends to marry such algorithmic advances with a parallel/distributed version of the software while providing a much enhanced user interface. This should enable more comprehensive searches to performed rapidly while reducing the reliance on lookup tables (currently over 10Gb).

Suggested Changes

1. Instead of the current four options there will be only two

**Automated Lookup**

1. Multiple representations will be filtered and only when there are distinct 'hits' will more than one answer be offered. Heuristics
will be developed to offer the simplest or most elegant representation.

2. The user input will be a real number to a minimum required precision.
3. The user may be able to set a processor number or a time limit.
4. If the answer is algebraic, we may attempt to offer a solution in radicals when such exists.

User Guided Lookup

1. In addition to the features above the user will be able to specify basis elements to be included or excluded from the search.
2. We will attempt to exploit as many of the features of identify as possible.
3. Products and sums will be allowed.
4. A hunt only for algebraic numbers is another option.
5. We will be guided by and may be able to use some of the features of the EZface-plus and Integer Relations services still running at CECM.

Other Features

1. Design should allow for export or mirrors
2. Answers should be easily exportable into TeX, Maple (or Mathematica?)

JMB  February 5, 2007

Comments from DHB

1. We need to re-think the constants that are in the index. Some of the ones present are unlikely to appear (physical constants, for instance, seem pointless to include), and so the list could be scrubbed a bit. Other classes arguably should be added.
2. We need to discuss and carefully define exactly what checks we will make, and in what order. Some more expensive checks, such as multi-term expressions, may be specified with the proviso that it will run longer.
3. We might also offer the user the option, if he/she wishes to run a long check (longer than we might care to run on our public, web-based server), the option of downloading the software and running it on their own system or cluster.
4. I recommend contenting ourselves with accepting numerical input only, rather than attempting to permit Maple or Mathematica expressions. This would add another major level of complexity, bugs and license issues that I would rather not deal with. Besides, there is some virtue in having an
identical software framework in the two or three places that host it (and possibly others who wish to download the software as well).