Exact Real Arithmetic Systems:
Results of Competition

Jens Blanck

University of Wales Swansea, Singleton Park, Swansea, SA2 8PP, UK

Abstract. A competition between systems for doing exact real number computations was held in September 2000. We present the results obtained and give a short evaluation of the different approaches used.

Introduction

Computable Analysis is flourishing theoretically, and during the last decade there have also been some practical advances. Systems for doing exact real number computations have appeared. Although still in their infancy, some of these systems are capable of non-trivial practical computations. The systems use very different approaches and it was therefore decided to have a competition between existing systems for exact real number computations as part of the 4th workshop of Computability and Complexity in Analysis at Swansea (CCA 2000). The competition was organised by David Lester.

The aims of the competitions were: To establish the state of art in exact real number computations; and to be a forum where competitors may debate ideas and implementation techniques from different systems. This was a first attempt at staging such a competition. The problems set were simple calculations that would check basic capabilities of the systems.

The contestants are listed in Table 1.

Table 1. The contestants.

Competitor	System	Acronym
David Lester (Manchester)	Manchester Arithmetic Package	MAP
Norbert Müller (Trier)	iterative Real RAM	iRRAM
Marko Krznaric (Imperial College)		IC
Tom Kelsey (St Andrew)		Kelsey
Paul Zimmermann (INRIA Lorraine)	MPFR[1]	MPFR

The person running the system for the competition has been entered as the competitor. The system name has also been listed in case a specific name was known. The entries will henceforth be referred to by the acronyms listed in the

---

[1] Multiple Precision Floating-point Reliable library.

table. The competitor has been the main developer of their systems except for IC and MPFR, which are joint projects. See [1,2,3,4,5]. MPFR was a remote entry. All contestants except Kelsey were running their systems on AMD 800 MHz machines with 256 Mbyte RAM. Since Maple was not available on these computers, Kelsey was forced to run his system on a 233 MHz Pentium Laptop.

There were two sets of problems. One set containing the precirculated problems testing the minimum capability of the systems. The other set was announced to the contestants shortly before the competition.

The competition was supervised by a committee consisting of Günter Hotz, Daniel Richardson, Dieter Spreen, and the author.

Results

The results of the first set of problems is in Table 2. The tabulated timings are in seconds. Note that the table is given for different accuracy depending on implementation. The MAP, iRRAM, and MPFR systems computed the results to 10000 decimal places. The IC system is for 1000 decimal places, and Kelsey's system is for 100 decimal places. Also note that MPFR did not restart the system after each example as the rest of the contestants did; this explains why MPFR gets much shorter times for cosine than for sine. IC withdrew from the competition after two problems.

Table 2. Level 0 problems.

To Calculate	MAP	iRRAM	MPFR	IC	IC^2	Kelsey
$\sqrt{\pi}$	.05	.13	.75	.78	.13	
$\log \pi$	16.73	1.10	1.70	30.00	.39	89.22
$\sin e$	2.03	1.30	4.00		.30	111.26
$\cos e$	2.10	1.30	.09		.28	126.90
$\sin(\sin(\sin 1))$	5.79	.99	4.63		2.39	106.72
$\cos(\cos(\cos 1))$	5.88	.99	4.65		1.94	111.83
e^{e^e}	2.12	1.60	1.05		1.90	
$\log(1+\log(1+\log(1+\pi)))$	32.42	1.37	.73		6.25	102.49
$\log(1+\log(1+\log(1+e)))$	78.60	1.53	.95		8.96	123.29
$\log(1+\log(1+\log(1+\log(1+\pi))))$	112.73	1.90	1.29		10.19	141.36
$\log(1+\log(1+\log(1+\log(1+e))))$	179.87	2.03	1.51		11.67	136.19
$\sin 10^{50}$	2.02	1.48	49.28		90.33	139.52
$\cos 10^{50}$	2.03	1.47	.08		90.81	
e^{1000}	1.15	2.26	.39		1.11	
$\arctan 10^{50}$	.17	.20			10.79	

The results of the second set of problems is in Table 3. The contestants were given one hour to run these problems. The second set was not attempted by all

---

[2] These results for IC were supplied at a later date and are for 100 decimal places and run on a 233 MHz Pentium laptop.

contestants. MPFR did not get this set of problems. Kelsey did not compute any of them. Again, the accuracy was 10000 decimal places except for IC giving 1000 decimal places.

The logistic map problem was to compute 1000 iterations of the logistic map

$$x_{n+1} = \frac{15}{4}\left(x_n - x_n^2\right),$$

with starting value $x_0 = \frac{1}{\pi}$, to ten decimal places.

Table 3. CCA 2000 problems.

To Calculate	MAP	iRRAM	IC	IC[3]
$e^{\pi\sqrt{163}}$	1.07	2.43	26.72	1.40
$\sqrt[3]{\sqrt[5]{\frac{32}{5}} - \sqrt[5]{\frac{27}{5}}} - \frac{1 + \sqrt[5]{3} - \sqrt[5]{9}}{\sqrt[5]{25}}$	199.53	3.74		11.47
$\sin\left(\frac{3 \log 640320}{\sqrt{163}}\right)$	15.87	1.31		3.28
Logistic map	>500.00	0.07		4h

Correctness

A comprehensive checking of correctness has not been performed, although correctness of the computed results is obviously very important. Ideally, each system should be accompanied by a proof of the correctness of the system. Such proofs exists for IC and an earlier version of MAP.

As a quick check of correctness, MAP and iRRAM results have been checked for some of the problems and agree up to 10000 decimal places. As these two systems are implemented in different ways this is a good indication that they actually compute the correct result. This consensus approach to validate results is of course only an indication of correctness, not a proof.

Evaluation

The three systems, iRRAM, MAP, and MPFR, have in common that they compute on *fast converging dyadic Cauchy sequences*. The iRRAM uses an iterative bottom-up approach. It starts with a predefined precision of the inputs. The subexpressions are then evaluated bottom-up and only the guaranteed precision after the operation is forwarded to the next subexpression. If the precision of the

---

[3] These results for IC were supplied at a later date and are for 100 decimal places and run on a 233 MHz Pentium laptop.

result is not sufficient, it recomputes the whole expression with increased precision. The method used in MPFR is also bottom-up but it uses floating point calculations with *directed* rounding. MPFR rounds the expression both up and down and if the resulting interval is too wide, it increases the input precision. In contrast, MAP uses a top-down approach. The precision needed for each subexpression is derived from the expression tree. The MAP does not need to do any recomputations.

The IC uses *linear fractional transformations* (LFT). In this approach, trees of LFTs are constructed, which are then normalised to a normal form. Some operations are derived from algorithms used on continued fractions, since these can be encoded into LFTs. This approach has the advantage that the operations have the best possible convergence speed. However, it also means that IC suffers from the rapidly growing coefficients that come with long continued fractions.

Kelsey uses symbolic computations in Maple. It is clearly slower than the other approaches. This can only partly be explained by the inferior hardware that it was run on.

That IC did not perform very well must be taken as indication that the LFT approach is not as efficient as the Cauchy sequence approach. This probably also applies to approaches using pure continued fractions.

There is not much to separate the Cauchy sequence based approaches for many of the expressions tested. However, the consistency of the iRRAM made it the clear choice as winner in this competition.

Winner. The winner of the competition was Norbert Müller's iRRAM.

Criticism

The value of the results presented herein should not be overestimated. It is of course always hard to make such a competition fair and conclusive. However, there are some issues that really should be addressed. There was no attempt to deduce the performance of the systems as a function of the size of the problem. By problem size one can consider both the precision required and the size of the expression. (The logistic map example can be seen as a really big expression if the iterations are unfolded.) One should also be careful with the choice of problems. It may be that one approach is particularly well-suited to certain problems.

Another shortcoming of the Swansea 2000 competition was that only calculator style problems were included. For the next competition it would be nice to see some other types of problem, e.g., integration, differential equations, linear algebra (the iRRAM already does matrix operations). This would require the competitors to implement other data types apart from the reals, e.g., the space of continuous functions.

References

1. A. Edalat and J.P. Potts, A New Representation for Exact Real Numbers, *Electronical Notes in Theoretical Computer Science* 6 (1997),
2. P. Gowland and D. Lester, A Survey of Exact Arithmetic Implementations, in: J. Blanck, V. Brattka and P. Hertling (editors), *Computability and Complexity in Analysis*, Lecture Notes in Computer Science, Springer, this volume.
3. MPFR group, The MPFR library, `http://www.mpfr.org/`.
4. N. Müller, The iRRAM: Exact Arithmetic in C++, in: J. Blanck, V. Brattka and P Hertling (editors), *Computability and Complexity in Analysis*, Lecture Notes in Computer Science, Springer, this volume.
5. P.J. Potts, Exact Real Arithmetic Using Möbius Transformations. PhD Thesis, University of London, Imperial College, 1998.