

HP Forum Archive 21

[[Return to Index](#) | [Top of Index](#)]

Estimating accuracy in finite precision computations

Message #1 Posted by [mpi](#) on 20 Feb 2013, 6:22 p.m.

Is there any calculator able to provide actual accuracy information along with numeric value, at the end of a computation chain?

It's a bit unfortunate a machine displays a numeric value, after eventually several rounding stages it's the only one to know about, going through functions and domains it knows about, and still does not translate some sort of accuracy estimate about the final number in a human readable form. (ok, I sound a bit provocative here ;)

I'm interested about any feedback on this, and eventual pointers for solutions.

Thanks.

(initial but incomplete discussion here <http://h30499.www3.hp.com/t5/Calculators/Providing-accuracy-information-in-finite-precision-computations/td-p/5961807>)

Edited: 20 Feb 2013, 6:24 p.m.

Re: Estimating accuracy in finite precision computations

Message #2 Posted by [Paul Dale](#) on 20 Feb 2013, 11:09 p.m.,

in response to message #1 by [mpi](#)

Interval arithmetic can provide a bounds estimate on a result. No calculator currently implements this natively, although I did consider it for the WP 34S.

There are a few interval arithmetic libraries around (mpfi e.g.). None are really suitable for a calculator.

- Pauli

Re: Estimating accuracy in finite precision computations

Message #3 Posted by [Valentin Albillo](#) on 21 Feb 2013, 4:22 a.m.,

in response to message #1 by [mpi](#)

Quote:

Is there any calculator able to provide actual accuracy information along with numeric value, at the end of a computation chain?

It is perfectly possible and actually easy to do that with some advanced HP calculators. With the proper technique, you can actually get both the result of a completely arbitrary computation chain and a pretty good estimate of the number of correct digits in said result.

A few years ago (2008 ?) I wrote a comprehensive 12-page article intended for timely publication in **Datafile** as part of my "**Boldly Going ...**" then-regular series of articles, showing in full detail how to do it and including plenty of examples to both see the technique at work and show its reliability. Alas, it's still unpublished as of today, as is the case with a dozen or so other such articles I had prepared.

If and when I eventually find some suitable publication which will accept it (them), I'll let the HP Forum know.

Best regards from V.

Re: Estimating accuracy in finite precision computations

*Message #4 Posted by **Gjermund Skailand** on 21 Feb 2013, 5:04 a.m.,
in response to message #3 by Valentin Albillo*

In my longfloat library for hp49 and hp50g I did include interval arithmetic. It was not tested much, and It does not cover matrix calculations. see www.hpcalc.org

Re: Estimating accuracy in finite precision computations

*Message #5 Posted by **Paul Dale** on 21 Feb 2013, 5:50 a.m.,
in response to message #4 by Gjermund Skailand*

Does it cover trig, exponential and logarithmic functions?

Doing basic arithmetic using intervals isn't difficult and the 34S would be more than capable of supporting the required operations -- it has a very good selection of rounding modes which are required for this.

- Pauli

Re: Estimating accuracy in finite precision computations

*Message #6 Posted by **Gjermund Skailand** on 21 Feb 2013, 6:06 a.m.,
in response to message #5 by Paul Dale*

yes it covers trig, exponential and logarithmic functions, and you can use it for automatic evaluating of formulas. However it is sys-rpl and so it is rather slow... I have been trying to port it to hpgcc3 (for the hp50g) but can't get the decnumber library to round figures correctly. - Gjermund

Re: Estimating accuracy in finite precision computations

*Message #7 Posted by **Walter B** on 21 Feb 2013, 5:08 a.m.,
in response to message #3 by Valentin Albillo*

Buenas dias Valentin,

Your post's mouth watering on one hand and frustrating on the other. I know about your dispute with Datafile but don't see any solution so far. OTOH I don't like the community being taken hostage for that. Is there a chance you find a way to share your findings with the people here?

d:-)

P.S.: I'm no member of HPCC anymore for some years now (but have published a bit there, too, though far more basic than your contributions).

Re: Estimating accuracy in finite precision computations

*Message #8 Posted by **Valentin Albillo** on 21 Feb 2013, 7:29 a.m.,
in response to message #7 by Walter B*

Quote:

Buenos dias Valentin,

Gutten Morgen, Walter.

Quote:

Your post's mouth watering on one hand and frustrating on the other.

Thanks a lot for your interest and kind appreciation but I must say it's not my intention to frustrate anyone, why would you say that ? ... 8-D

Quote:

I know about your dispute with Datafile but don't see any solution so far.

It's probably the case that discussion about this dispute is out of topic in this Forum so I'll refrain from *starting* it.

Nevertheless, for any dispute or problem, arriving at a solution requires as a bare minimum that the other part shows some *interest* and at the very least does try to *communicate*. No such communication attempt has ever taken place for six years so I take it there's simply no interest whatsoever in my articles, which is fine by me.

Quote:

OTOH I don't like the community being taken hostage for that.

"... *the community being taken hostage* ..." !?

That's rich ! ...Why would you say that ? ... XD

Quote:

Is there a chance you find a way to share your findings with the people here? d:-)

As I see it, there are mainly two possibilities, namely: (a) The other part shows interest and attempts communication. (b) Eventually some third party is found which would publish the articles.

Once the articles appear in print I would make the PDF versions available online timely and for free, of course.

Quote:

P.S.: I'm no member of HPCC anymore for some years now (but have published a bit there, too, though far more basic than your contributions).

Neither am I.

I went on to pay for membership for six months after the trouble started, waiting for the matter to be discussed and resolved at the Annual General Meeting.

It wasn't so I terminated my subscription as I wasn't paying to read my own published materials (which constituted a considerable percentage of every issue) and the rest (most of it RPL and hardware topics) were mostly of little interest to me, exception being made of occasional material by people such as Bill Butler, Gene Wright, or yourself, who, regrettably, didn't contribute enough to justify a subscription.

Best regards from V.

Re: Estimating accuracy in finite precision computations

*Message #9 Posted by [Wlodek Mier-Jedrzejowicz](#) on 24 Feb 2013, 7:27 p.m.,
in response to message #8 by [Valentin Albillo](#)*

Dear Valentin, as you say, this is not the right place to discuss Datafile, but where is? You were upset that we did not communicate to you the result of an AGM and you told me that you would consider sometimes writing for Datafile again, but not regularly. Can I ask you if you would write about this topic, please? Many thanks and Best Wishes.

Re: Estimating accuracy in finite precision computations

*Message #10 Posted by [Valentin Albillo](#) on 25 Feb 2013, 8:21 a.m.,
in response to message #9 by [Wlodek Mier-Jedrzejowicz](#)*

Dear Wlodek:

Quote:

Dear Valentin, as you say, this is not the right place to discuss Datafile, but where is?

First of all, thanks for replying to my post here, Wlodek, I really hope you're doing well.

Then again you might perhaps remember that I did send you a lengthy e-mail five months ago (2012-10-07), which you promptly replied to a few hours later. It was a very, very short reply as, in your own words: "*Please forgive me, I'll reply in detail later - I am truly short of time*".

Alas, it wasn't to be. You *didn't* reply later "*in detail*" or at all, as you said you would do, and five months have elapsed with no forthcoming reply from you so, as I can't buy that you've been "*truly short of time*" for a whole five months, then I'm forced to guess that it all boils down to a matter of interest, as I said in my previous post.

Of course I understand you're very busy with all kinds of matters, including personal ones and such, but actually that's also my case and it's probably the case for many people participating in this forum, many of which also have their hands full of things and businesses to attend to.

In the end it's just a matter of interest and priorities, and it seems your interests and your priorities do not extend to emailing detailed replies to my emails or indulging in a productive exchange of ideas with me.

This I can understand and fully respect but I also value my time and have my priorities so you'll receive no further "lengthy" emails from me unless and until you're willing to reply to them in a reasonable amount of detail, so that I'll know for sure that you're indeed interested and not going through the motions out of mere politeness.

Take care, and Best regards from V.

Re: Estimating accuracy in finite precision computations

Message #11 Posted by [mpi](#) on 21 Feb 2013, 8:25 a.m.,
in response to message #3 by Valentin Albillo

Thanks Paul and Valentin.

Valentin, this is really great news, and I can't wait to see such function being tightly integrated by some manufacturer: that would be a really useful step-change feature!

Have you implemented a version on recent products such as HP50 or HP39gII?

While there seem to be some unfortunate issues preventing release of this algorithm, could you share some actual results in a few example for now (could even be screen-capture from HP emulators)?

For instance, what would be the estimation in case of the calculator forensic test? Could you share the output of your algorithm at each calculation step?

Thanks!

Edited: 21 Feb 2013, 9:16 a.m.

Re: Estimating accuracy in finite precision computations

Message #12 Posted by [hugh steers](#) on 21 Feb 2013, 11:24 a.m.,
in response to message #1 by [mpi](#)

Some years ago i wrote a calculator on this basis. Also a version of LUA that incorporated a dynamic precision number type.

Here's a really old page with a calculator for the pocket PC [exact](#)

Also, you can download windows binary to play with the calculation engine, here. [exact.exe](#)

I just recompiled it so you wont have to find the DLLs from 1974 :-)

The calculator uses dynamic precision. I did a talk a while back on the implementation of this idea. Basically you encode an interval as a single number and perform the appropriate interval arithmetic. For trig functions you have to widen your uncertainty by an upper error bound from your approximation. eg for Taylor series, use the truncated term as the error bound. and so on...

Yes, it's definitely possible on handheld calculators. Personally, i think they should _all_ do this internally, even for a cheap college model. It would be educational to show kids where their calculations go bad.

Re: Estimating accuracy in finite precision computations

Message #13 Posted by [mpi](#) on 21 Feb 2013, 12:14 p.m.,
in response to message #12 by [hugh steers](#)

Thanks Hugh, I'll try to grab a PC to have a look

Quote:

Personally, i think they should _all_ do this internally, even for a cheap college model. It would be educational to show kids where their calculations go bad.

Can't agree more! And I think first maker that integrates something like this (and of course properly market it to teachers), will see some good return on marketshare: it's a such a striking and valuable differentiator (until some else gets it done).

PS:

maybe a nice addition to contemplate for Reckon then! ;)

Edited: 21 Feb 2013, 12:20 p.m.

Re: Estimating accuracy in finite precision computations

*Message #14 Posted by [hugh steers](#) on 21 Feb 2013, 12:36 p.m.,
in response to message #13 by mpi*

Yes, i think something like this could be a market opener - the calculator that never gets the wrong answer or something. I'm sure marketing could come up with a great way to sell this idea. The Education and financial calculator market strike me as obvious targets.

Regarding hardware, pretty much all existing platforms could easily accommodate the additional calculation overhead. CPU is plentiful, you'd need a little bit more RAM than before, but not a lot.

My old code was a mashup of ideas during the time i was discovering this technique, that's why i never released source code. Maybe I'll rewrite it properly. If i do, it will be a lot smaller and faster.

The Reckon project will morph into something else, as well as stuff like this, i wanted to add a programming language and a CAS. The idea is the CAS would be written in the programming language itself. Also, before anyone says so, i'd like to stress that the "CAS" will be of a fairly pedestrian nature. There's no point in dreaming it will be as good as the big cas packages, it can't be. But it could, nevertheless still be useful in a limited way when on a portable device (eg polynomials, series, simple calculus etc.).

Re: Estimating accuracy in finite precision computations

*Message #15 Posted by [Thomas Klemm](#) on 21 Feb 2013, 1:50 p.m.,
in response to message #14 by hugh steers*

Quote:

the calculator that never gets the wrong answer

I'd be interested to see what's happening after a few iterations of the [logistic map](#) when $r = 4$:

$$x_{n+1} = 4x_n(1 - x_n)$$

Kind regards

Thomas

Re: Estimating accuracy in finite precision computations

Message #16 Posted by [hugh steers](#) on 21 Feb 2013, 4:29 p.m.,
in response to message #15 by Thomas Klemm

Regions that are stable will calculate quickly, but unstable regions - where lots of cancellation is happening, wind up calling in more and more precision.

Your choices are to keep expanding the precision to try to meet a target, where eventually you have to stop due to constraints, OR to expand only to a limit then deliver the number of digits you can prove are correct.

Some calculations can result in the total annihilation of all precision. Amusingly you need a new symbol to display in this case (eg "-" or just ".", or perhaps "-." if known negative!).

But at least you know you haven't got an answer and, like i said, "never gets the wrong answer"

:-)

Re: Estimating accuracy in finite precision computations

Message #17 Posted by [Thomas Klemm](#) on 21 Feb 2013, 5:54 p.m.,
in response to message #16 by hugh steers

A fixed point of $x = 4*x*(1 - x)$ is $3/4$, the other being 0 , but that's not so interesting.

We obviously can represent 0.75 exactly in a calculator, even in binary. So if you iterate the logistic map you will stay at the same number. If however you use a small interval around this fixed point the iteration of the logistic map will smear it all over the whole interval $[0, 1]$.

This little program will illustrate this behavior:

```
#!/usr/bin/python
```

```
def f(x):
    return 4*x*(1 - x)
```

```
eps = 1E-6
```

```

x = 3 / 4.0
u = x - eps
v = x + eps

for i in range(20):
    print "%.6f %.6f %.6f" % (u, x, v)
    [u, x, v] = map(f, [u, x, v])

```

This is the result:

```

0.749999 0.750000 0.750001
0.750002 0.750000 0.749998
0.749996 0.750000 0.750004
0.750008 0.750000 0.749992
0.749984 0.750000 0.750016
0.750032 0.750000 0.749968
0.749936 0.750000 0.750064
0.750128 0.750000 0.749872
0.749744 0.750000 0.750256
0.750512 0.750000 0.749488
0.748975 0.750000 0.751023
0.752045 0.750000 0.747949
0.745893 0.750000 0.754085
0.758147 0.750000 0.741764
0.733441 0.750000 0.766201
0.782021 0.750000 0.716548
0.681856 0.750000 0.812428
0.867713 0.750000 0.609554
0.459147 0.750000 0.951991
0.993324 0.750000 0.182815

```

While the interval might indicate a "total annihilation of all precision" the main value is still 100% correct.

It might seem obvious in the case above where the periodicity is 1 because you notice that 0.75 is a fixed point.

Quote:

For the $r = 4$ case (...) there exist cycles of length k for all integers $k \geq 1$.

You might return after 17 iteration to your initial value (or to something very close) but think after 16 iterations that the result is utter garbage because the interval indicates so.

Kind regards

Thomas

Just to illustrate the last point we can calculate the fixed points of the n-th iteration using the formula: $x_n = \sin^2(2\pi y_n)$

You can use the following lines in the program above to calculate the fixed point for 5 iterations:

```
from math import *
y = 2**5 - 1
x = sin(2*pi/y)**2
```

This results in the following table:

0	0.040520	<u>0.040521</u>	0.040522
1	0.155513	0.155517	0.155520
2	0.525314	0.525325	0.525335
3	0.997437	0.997435	0.997433
4	0.010227	0.010235	0.010243
5	0.040489	<u>0.040521</u>	0.040553
6	0.155399	0.155517	0.155634
7	0.525000	0.525325	0.525649
8	0.997500	0.997435	0.997369
9	0.009975	0.010235	0.010498
10	0.039503	<u>0.040521</u>	0.041551
11	0.151771	0.155517	0.159299
12	0.514947	0.525325	0.535692
13	0.999106	0.997435	0.994904
14	0.003571	0.010235	0.020278
15	0.014234	<u>0.040521</u>	0.079468
16	0.056125	0.155517	0.292613
17	0.211900	0.525325	0.827962
18	0.667992	0.997435	0.569763
19	0.887114	0.010235	0.980532
20	0.400571	<u>0.040521</u>	0.076354

As can be seen we return to the initial value after the 5th, 10th, 15th, ... iteration, while the interval degrades.

Or if you prefer the example where $n = 17$:

```
from math import *
y = 0b11011100101110 / (2.**17 - 1)
x = sin(2*pi*y)**2
```

0	0.392606	<u>0.392607</u>	0.392608
1	0.953866	0.953867	0.953868
2	0.176022	0.176019	0.176016
3	0.580154	0.580145	0.580137
4	0.974302	0.974307	0.974312
5	0.100152	0.100132	0.100112
6	0.360486	0.360423	0.360360
7	0.922143	0.922073	0.922002
8	0.287182	0.287419	0.287656
9	0.818833	0.819237	0.819640
10	0.593381	0.592351	0.591320
11	0.965120	0.965885	0.966642
12	0.134655	0.131804	0.128980
13	0.466091	0.457728	0.449377
14	0.995401	0.992852	0.989749
15	0.018312	0.028387	0.040583
16	0.071908	0.110323	0.155746
17	0.266948	<u>0.392607</u>	0.525956
18	0.782748	0.953867	0.997305
19	0.680215	0.176019	0.010750
20	0.870091	0.580145	0.042540

Edited: 21 Feb 2013, 7:21 p.m.

Re: Estimating accuracy in finite precision computations

Message #18 Posted by *mpi* on 22 Feb 2013, 9:44 a.m.,
in response to message #14 by *hugh steers*

Quote:

My old code was a mashup of ideas during the time i was discovering this technique

Bart on the HP Community forum pointed me to prof. William Kahan's articles <http://www.cs.berkeley.edu/~wkahan/>
Following one is particularly interesting about causes & remedies...
<http://www.cs.berkeley.edu/~wkahan/Mindless.pdf> (and it's longer version for the most courageous)

Have similar type studies been of inspiration for the techniques you or Valentin developed?

Edited: 22 Feb 2013, 9:47 a.m.