

HP Forum Archive 21

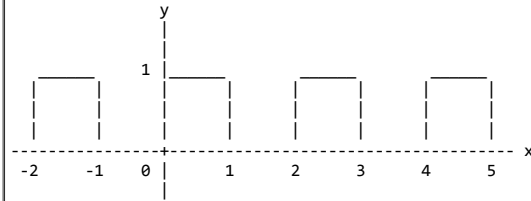
[[Return to Index](#) | [Top of Index](#)]

Another tough integral

Message #1 Posted by [Gerson W. Barbosa](#) on 13 May 2012, 3:09 p.m.

$$\text{Let } f(x) = \frac{1}{2} + \frac{\cos^{-1}(\cos^2(\pi x))}{3 \sin(\pi x)},$$

a fair approximation to the square wave function with an offset of 1/2 below:



Let us find the integral of $f(x)$ from $x=0$ to $x=1$.

WolframAlpha returns only a numerical result, [0.995555](#), which suggests there is no known closed-form solution.

Let's see how some calculators handle the task:

WP 34S:

```
001 LBL 00
002 # pi
003 *
004 COS
005 x^2
006 ACOS
007 x<>y
008 # pi
009 *
010 SIN
011 3
012 *
013 /
014 .
015 5
016 +
017 RTN

0.9955554977636525 wp34s (v2)
0.9955554973850958 wp34s (v3.0 2458)
```

HP-15C LE:

```
0.995555102 (FIX 7)
```

HP-50g:

```
'Integral(0,1,1/2+ACOS(SQ(COS(pi*X)))/(3*SIN(pi*X)),X)'
0.99555497542 (Fix 9)
```

Emu71:

```
>RADIANS
>STD
>INTEGRAL(0,1,1E-10,1/2+ACOS(COS(PI*IVAR)^2)/(3*SIN(PI*IVAR)))
0.995555497542
>INTEGRAL(0,1,1E-11,1/2+ACOS(COS(PI*IVAR)^2)/(3*SIN(PI*IVAR)))
0.995555198305
```

It looks like our calculators arsenal cannot give us more than 8 correct decimal places, 0.995555497. (I haven't tried the latest WP 34S release yet).

Edited for grammar.

Edited: 13 May 2012, 3:22 p.m.

Re: Another tough integral

Message #2 Posted by [Marcus von Cube, Germany](#) on 13 May 2012, 3:59 p.m.,
in response to message #1 by Gerson W. Barbosa

The latest WP 34S release should return the same result but you can try in double precision and compare the figures.

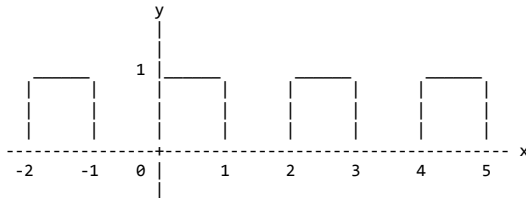
Re: Another tough integral

Message #3 Posted by [Valentin Albiilo](#) on 13 May 2012, 6:57 p.m.,
in response to message #1 by Gerson W. Barbosa

Quote:

$$\text{Let } f(x) = \frac{1}{2} + \frac{\cos^{-1}(\cos^2(\pi x))}{3 \sin(\pi x)},$$

a fair approximation to the square wave function with an offset of 1/2 below:



Let us find the integral of $f(x)$ from $x=0$ to $x=1$.

Well, it doesn't seem that tough at all, a simple integration routine I wrote for high-precision returns:

```
0.99555549776236782298677656111175223160096264596209...
```

essentially instantly (0:00:00, to the second), which is correct to all places shown. A second or so will get you 200 decimal places beginning with:

```
0.995555497762367822986776561111752231600962645962092224364259066155312626099042699264864294947084302054786866956732685735494339225156047080022695529860042854...
```

This might be useful to check for accuracy the results you get from calculators.

Best regards from V.

Re: Another tough integral

Message #4 Posted by **Gerson W. Barbosa** on 13 May 2012, 9:30 p.m.,
in response to message #3 by Valentin Albillo

Hello Valentin,

Quote:

A second or so will get you 200 decimal places beginning with:

0.99555497762367822986776561111752231600962645962092224364259066155312626099042699264864294947084302054786866956732685735494339225156047080022695529860042854...

I've checked the digit frequency of your result and compared it against that of Pi for the first 156 significant digits:

	your result	Pi
0	15	14
1	8	14
2	24	20
3	9	17
4	16	16
5	18	14
6	26	13
7	11	10
8	10	21
9	19	17

I would expect the result of the integral to have a more even digit distribution. Of course there is no solid reason behind this. Often integrals of periodic transcendental functions, depending on the limits of integration, are not transcendental numbers themselves, much less normal. The repetition of the digit '5' in the beginning might be an indication the result is not normal (as I may have wrongly assumed). Not that I don't trust it, but is there another integration method that may confirm your result?

Best regards,

Gerson.

Re: Another tough integral

Message #5 Posted by **Valentin Albillo** on 14 May 2012, 2:02 a.m.,
in response to message #4 by Gerson W. Barbosa

Quote:

Not that I don't trust it, but is there another integration method that may confirm your result?

It might be the case that the meager 156 digits I gave aren't enough for an statistically significant digit frequency calculation so I let my routine run for a minute and there you are, about 1,000 decimal places for you to fiddle with to your heart's content

0.9955549776236782298677656111175223160096264596209222436425906615531262609
90426992648642949470843020547868669567326857354943392251560470800226955298600428
54061223230260667542990343571102929104397030422418588620801216306986461891226713
12458538366567533281556903416297515134295492756873107350286487265699127726709068
02866195453154345434350558071871988003701209930033784114664742830891982441996720
14204397003759667135107370215878694197174112228164086756751024279765262148865802
14426869108178025469920516054579901976743735865432183534421777147289493851489995
34900344460945637494513818059645286910899589401623806133301582292846669833025305
75559408429314709927714148536378809730554519179542358963419925913618096446986160
89707018534727889549943105208309888722681503586468722935816711860533157811264171
46873750916264977276849274618602746297307576605027226287396682942279959930995857

63544972773656734777298255881221987614643146784249414791351811025549311074079353
 9181765455749920736361474456647892716652144730718950692026594512267763789970029...

For reliable digit frequency results it's frequently the case that many thousands or even millions of decimal digits are necessary, and even more. Consider for instance this simple square root computation:

$\text{sqrt}(308642) = 555.555577777773333333511111102222227199999...$

Seeing that 44-digit result for its 6-digit integer argument you might be induced to believe that it is anything but normal, with obviously biased digit counts. However, the mirage quickly fades away the moment you look further:

555.5555777777733333335111111022222271999997013333521066654464000813511055792
 35937757839912506782260202536969737338174042982693335359822074938022460559677134
 80453936671895444061294508193099456043131900500139679957197210080367577711228344
 88033492927605660571893841517753635572851565452015965971830849488612380140811581
 65086640947765975127949499079439218226790840268539816815091386804202414469605745
 72371836881741155667721985539461145895339203043493051583779961546365025268947444
 55853751944622774941855495715419031133120938831434794450531321168458752080467791
 63014094929372714096921707984170197616439799857571395157854920325300060444559270
 50627657849491643773882305857452761242714492527784638035394215448725969098842689
 41210190584508455675940399110409463416619952140752221724065975226358154445769047
 20955493675153265579204818884769350972126051641884022716404454461648737349442853
 71015481254270598108264574726782545171053067106881548003259525083330892109951136
 99664941708194946290983595047955472874158265042111361146447247573233473458118379
 10816149125024610778070772206907264334574601215626802573125232732663840608892935
 67078985853834062943931460643680720460050842175328769283886123614901626961378958...

which surely looks much more "normal" and about as normal as an irrational square root is going to be.

Making judgments based on scarce data is what Wikipedia calls "Hasty generalization, a logical fallacy also known as 'the law of small numbers'"

Best regards from V.

Re: Another tough integral

Message #6 Posted by **Gerson W. Barbosa** on 14 May 2012, 6:03 a.m.,
 in response to message #5 by Valentin Albillo

Thank you, Valentin, for taking the time to compute a staggering one thousand digits of the integral and providing yet another interesting example and explanations.

Best regards,

Gerson.

Re: Another tough integral

Message #7 Posted by **Valentin Albillo** on 14 May 2012, 6:55 a.m.,
 in response to message #6 by Gerson W. Barbosa

Quote:

Thank you, Valentin, for taking the time to compute a staggering one thousand digits of the integral and providing yet another interesting example and explanations.

You're welcome, Gerson, but this does not deserve any thanks, it was just a 1-min computation plus the time to copy & paste the result to post it.

What's weird about it all is that I can perform such computations, i.e., multi-digit integrals, matrix operations, general programming, etc., not just on a desktop PC, not just on a laptop, not just on a tablet or even a mobile phone, but also on a device as simple as my eReader !

So I think handheld calculators are really, really a thing of the past and can't compete at all with modern technology. Just imagine, doing pretty convoluted math on a humble eReader which weights even less than a typical advanced graphing calculator, doesn't tax your eyes, consumes extremely little power, and which you can carry with you at all times with no burden whatsoever.

We old-timers are now living in the future ... :D

Best regards from V.

Re: Another tough integral

Message #8 Posted by [Oliver Unter Ecker](#) on 14 May 2012, 12:24 p.m.,
in response to message #7 by [Valentin Albillo](#)

Hi Valentin,

Quote:

Just imagine, doing pretty convoluted math on a humble eReader which weights even less than a typical advanced graphing calculator, doesn't tax your eyes, consumes extremely little power, and which you can carry with you at all times with no burden whatsoever.

Yes, I second that.

Were you extrapolating, or do you have an actual implementation on your eReader? If so, native code or JavaScript? If latter, I'd be very interested to add your integration routine to my recent BigFloat implementation, if you'd be willing to share it.

Cheers.

Re: Another tough integral

Message #9 Posted by [Valentin Albillo](#) on 14 May 2012, 6:42 p.m.,
in response to message #8 by [Oliver Unter Ecker](#)

Quote:

Were you extrapolating, or do you have an actual implementation on your eReader? If so, native code or JavaScript?

No, no, I wasn't extrapolating, my high-precision numerical integration routine actually *does* run flawlessly on my [Sony eReader](#) (10.4 ounces ~ 295 grams) under DosBox for Android, as I wrote it many years ago for an MS-DOS environment.

Matter of fact lots of pretty useful MS-DOS software runs as well (Turbo Pascal, for instance, DOS-based emulators, etc.), as well as lots of native Android applications. As long as the particular app isn't heavy with animations (Chess, Sudoku, crosswords, word-search puzzles, comics readers, audio players, etc), it'll run fine on its beautiful E-Ink Pearl 800x600 screen.

Thanks a lot for your interest and

Regards from V.

Re: Another tough integral

Message #10 Posted by [Oliver Unter Ecker](#) on 15 May 2012, 1:58 a.m.,
in response to message #9 by [Valentin Albillo](#)

Ah, I see. Thanks for the clarification.

I thought of JavaScript because it's the one language that comes built-in with these devices, as part of their browsers.

I share your enthusiasm about what could be done with eReaders and calcs.

Edited: 15 May 2012, 2:00 a.m.

Re: Another tough integral

Message #11 Posted by **Crawl** on 14 May 2012, 1:02 p.m.,
in response to message #7 by Valentin Albillo

Quote:

So I think handheld calculators are really, really a thing of the past and can't compete at all with modern technology. Just imagine, doing pretty convoluted math on a humble eReader which weights even less than a typical advanced graphing calculator, doesn't tax your eyes, consumes extremely little power, and which you can carry with you at all times with no burden whatsoever.

We old-timers are now living in the future ... :D

Best regards from V.

I'm not sure why the moral here is "handheld calculators are a thing of the past" rather than "handheld calculators of the future have something to learn from existing ereaders".

Re: Another tough integral

Message #12 Posted by **bill platt** on 15 May 2012, 7:56 a.m.,
in response to message #11 by Crawl

It's the "convergence theorem"

Re: Another tough integral

Message #13 Posted by **Crawl** on 15 May 2012, 9:17 p.m.,
in response to message #12 by bill platt

I love the idea of a capacitive touch screen for a calculator. For one, I'd like to use it literally like scratch paper for "manual" calculations. I'd also like to see a way of inputting formulas through handwriting recognition.

And I'd like to see these two capacities work together.

Re: Another tough integral

Message #14 Posted by **DeboT** on 14 May 2012, 6:57 a.m.,
in response to message #5 by Valentin Albillo

Thank you for your insightfulness. However, are the 1000 digits generated from a different method than to 200 digits? If not, then you have not answered Gerson's question (you have merely refuted the basis of his query).

Quote:

Making judgments based on scarce data is what Wikipedia calls "Hasty generalization, a logical fallacy also known as 'the law of small numbers'"

The same goes for the number of answers *from fundamentally different sources* to affirm the accuracy. So far, yours is the only one more than 16 digits.

Quote:

This might be useful to check for accuracy the results you get from calculators.

It would be very stupid indeed for anyone to take your single source as a definitive answer to *check* all others against. Perhaps as a *comparison*, but certainly not a *check*. I do not even use Wolfram Alpha answers as definite checks, but merely as comparisons.

So far from this thread, the best we can come up with is the first 6 digits (0.995555) and then only with a very low confidence due to the small sample. I.e. it could all be horribly wrong... (unlikely, but possible).

Re: Another tough integral

Message #15 Posted by [Steve Perkins](#) on 14 May 2012, 7:48 p.m.,
in response to message #4 by Gerson W. Barbosa

I can confirm the result given by Valentin to the first 30 digits of precision, using a high-precision method I programmed a few years back. Adaptive integration using double exponential quadrature.

My method was designed for tough integrals, but doesn't achieve results very quickly in high precision applications. It took about a minute to get (and confirm) 30 digits.

As always I admire Valentin's abilities and the tools he creates.

Edited: 14 May 2012, 7:51 p.m.

Re: Another tough integral

Message #16 Posted by [Valentin Albillo](#) on 14 May 2012, 11:36 p.m.,
in response to message #15 by Steve Perkins

Quote:

I can confirm the result given by Valentin to the first 30 digits of precision, using a high-precision method I programmed a few years back. Adaptive integration using double exponential quadrature.

My method was designed for tough integrals, but doesn't achieve results very quickly in high precision applications. It took about a minute to get (and confirm) 30 digits.

As always I admire Valentin's abilities and the tools he creates.

First of all, thank you very much for your appreciation and kind words.

It would be simple to check my 1000+ digit result using Mathematica, Maple, or PARI/GP for instance but I have none of them at hand right now, just my eReader with 894 books, 1200+ (mostly math) papers, 550+ audio files, and about two dozen assorted applications installed on it which, regrettably, don't include any of those.

However, I have no doubts whatsoever that my results are correct, as I've thoroughly tested the routine innumerable times using much more troublesome cases than Gerson's fairly mild integral, with reasonable success so far.

Just for instance, this simple, elementary integral is probably a harder nut than Gerson's if dealing with it straightforwardly:

$$\int_0^{10} x^2 \sin(x^3) dx$$

To some 750+ decimal places, my quadrature routine produces:

```
0. 14587364123643233630725025779820134374806272608726769409905827138475545678
91873252258156604930609502083829281765371738264413094048636339249999429802062379
57961603563346425312582138935758511981714815713913326990354637466454659373462989
23856880365641819094067543287366102194594287500142135303138814301671482938831794
27974049757460366605942774949622910049520201352146447612765559727085467322830992
54379599727222565251775009728250768063382128194525269251261178725736672752044626
44859088056137118266467542972367110856139562847936891992188412495430195619961309
13352677329362058906113949895974412408123236712255902834497437414519914257696481
63439748933708957007997706981128617570867623590746630295371101538752591300416917
574665836311221415826565867926678856523022042718456...
```

which is demonstrably correct to *all* digits shown. On the other hand, the naive approach using the HP-71B and asking for full precision gives:

```
>INTEGRAL(0,10,0,IVAR^2*SIN(IVAR^3))
```

```
.145873640643
```

An interval approach fares somewhat better:

```
>S=0 @ FOR I=0 TO 9 @ S=S+INTEGRAL(I,I+1,0,IVAR^2*SIN(IVAR^3)) @ NEXT I @ S
```

.145873641554

You might want to try assorted calculators and your own high-precision quadrature to see how they cope with this simple integral.

Best regards from V.

Edited: 14 May 2012, 11:44 p.m.

Re: Another tough integral

Message #17 Posted by [Gerson W. Barbosa](#) on 15 May 2012, 12:19 a.m.,
in response to message #16 by Valentin Albillo

Quote:

However, I have no doubts whatsoever that my results are correct, as I've thoroughly tested the routine innumerable times using much more troublesome cases than Gerson's fairly mild integral, with reasonable success so far.

I should have known better. Sorry! I have yet to see a wrong result coming from you. Well, at least this has elicited some interesting discussions and insights.

Best regards,

Gerson.

Re: Another tough integral

Message #18 Posted by [Mike Morrow](#) on 15 May 2012, 10:10 a.m.,
in response to message #16 by Valentin Albillo

The Casio fx-115ES Plus ("The Quadra-nator") confirms the accuracy of the first 11 significant digits shown above. :-)

But seriously...when programmed in LineIO mode (using the 115's default integration tolerance value) as $S(x^2\sin(x^3),0,10)$ it returns 0.145873641232327 (accurate to 11 digits).

The 10 digits shown in the display are correct. Computation time is 1003 seconds...just making it before the 115's 1050-second time-out halt. Better have some bright light on the solar cell to do many of these!

The equivalent integral $(S\sin(u)du)/3$ evaluated from 0 to 1000 produced 0.145873641237629, a result accurate only to 11 digits. It took 737 seconds to compute.

When solved and computed analytically as $-(\cos(1000)-\cos(0))/3$ on the 115, the result is 0.145873641236197. This is accurate *only to 12* digits. I was expecting a better result.

Edited to add results from 115's analytic solution.

Edited: 16 May 2012, 1:01 a.m.

Re: Another tough integral

Message #19 Posted by [Steve Perkins](#) on 15 May 2012, 2:47 p.m.,
in response to message #16 by Valentin Albillo

That is certainly a numerically challenging function to integrate over the suggested range. It increases while oscillating more and more as x increases. And it has the benefit of being analytically easy to solve with a simple substitution.

My high-precision routine had the grace to give up, since it quickly hit built in limits.

Other double precision integrators in my toolkit had mixed results.

Some gave close results for lower precision demands, but hung (taking too much time) when trying for precision better than 1E-14 or so.

Other integrators gave close answers, but with error estimates that were off by a factor of 10 or more.

A nice test case indeed. And motivation to attempt to make my tools more robust.

Re: Another tough integral

Message #20 Posted by [Mike Morrow](#) on 15 May 2012, 10:00 p.m.,

in response to message #19 by Steve Perkins

Quote:

A nice test case indeed. And motivation to attempt to make my tools more robust.

I wish there were at least one HP calculator with a numerical integration function that could even remotely match the performance found in my cheap non-programmable calculator that is sold in mass-market department stores like Walmart and Target in the USA. It's embarrassing that all of the HP line, even the 50g, performs so poorly when executing its native quadrature firmware.

Like Casio, the TI line has utilized Gauss-Kronrod quadrature for many years. It would be interesting to try this test case on my TI-89 Titanium HW version 4, but I don't have it nearby at present.

It is HP that needs its tools to be more robust.

Re: Another tough integral

Message #21 Posted by [Oliver Unter Ecker](#) on 16 May 2012, 3:33 a.m.,

in response to message #16 by Valentin Albillo

I confirmed your 750 digit result with the following W|A command-line: `NIntegrate[x^2*sin(x^3), {x, 0, 10}] 1000 digits`

Strangely enough, this command-line does not work with Gerson's function (`"1/2+(acos(cos(pi*x)^2))/(3*sin(pi*x))"` or `"1/2+(ArcCos[Cos[Pi*x]^2])/(3*Sin[Pi*x])"` for Mathematica syntax).

Instead, this `NIntegrate[1/2+(acos(cos(pi*x)^2))/(3*sin(pi*x)), {x, 0, 1}, WorkingPrecision->100]` will work, but be limited to 200 digits.

While `NIntegrate[x^2*sin(x^3), {x, 0, 10}, WorkingPrecision->100]` also works, it doesn't work with `WorkingPrecision->1000` (though the first command proves that W|A can compute 1000 digits without timing out).

If someone can make rhyme or reason out of this, I'd love to hear it.

I'm presently working on a CAS for ND1 that uses W|A to do the hard work. This thread inspired me to sort out high-precision numerical integration, and the result of a little bit of integration work is that RPL code like this

```
\<< 200 setBigFPrecision
0 toBig
10
'x^2*sin(x^3)' 'x' integrate
\>>
```

will numerically integrate a function to a desired number of digits. The "integrate" function, when fed BigFloat values, will produce a BigFloat result. A type in ND1, it can be used for subsequent computation.

The internal implementation of "integrate" looks like this

```
"integrate": function(a, b, expr, name) { return BigFloat.fromString(calculator.callWA("", "NIntegrate[" + calculator.expressionToWAexpression(expr) + ", {" + calculator.unquote(name) + "}, " + a .
```

The idea is that users can write their own adapter functions and thereby compute anything they like in Mathematica/WolframAlpha, while using a stack-based calc that operates very much like a 50g.

Re: Another tough integral

Message #22 Posted by [Valentin Albillo](#) on 16 May 2012, 5:32 a.m.,

in response to message #21 by Oliver Unter Ecker

Quote:

I confirmed your 750 digit result with the following W|A command-line: NIntegrate[x^2*sin(x^3), {x, 0, 10}] 1000 digits

Thanks, you can get as many digits as you want for checking purposes or otherwise by simply evaluating:

$$(1 - \cos(1000))/3$$

which is the exact result.

Regards from V.

Re: Another tough integral

Message #23 Posted by [Steve Perkins](#) on 16 May 2012, 7:58 p.m.,
in response to message #16 by Valentin Albillo

Valentin, I'd be quite interested in learning any details you could share about the workings of your quadrature routine, as well as your extended precision setup.

I've programmed adaptive quadrature using Simpson's and Gauss+Lobatto (which both require endpoint evaluation).

Also programmed Gauss+Kronrod, and Gauss+Bond routines which don't require endpoint evaluation. For these I have fixed weights and abscissa in the code making variable precision hard.

I also programmed double exponential, which works much differently than the above methods, but lends itself to variable precision calculation admirably.

I use MAPM by Michael C. Ring for extended precision since it's freeware and fairly easy to use and quite reliable. It can also be slow, especially if you're not careful with rounding results.

Anyway, if you can share anything more about your methods, I'd appreciate it. If not, I certainly understand.

Re: Another tough integral

Message #24 Posted by [Valentin Albillo](#) on 17 May 2012, 6:15 a.m.,
in response to message #23 by Steve Perkins

Hi, Steve:

Quote:

Valentin, I'd be quite interested in learning any details you could share about the workings of your quadrature routine, as well as your extended precision setup.

Thanks for your interest, Steve, and congratulations on your many and varied implementations of assorted quadrature algorithms, I see you're really keen on numerical integration. As for details on my own quadrature routine which I mentioned a few posts ago, I'm sorry but I'll keep them private for the time being, mainly for the following reasons:

- 1) my routine uses a simple algorithm, nothing fancy at all, and thus I don't think you'd learn anything worthwhile by having a look at it, in fact I think you'd be somewhat disappointed, a likely case of unfulfilled high expectations.
- 2) I wrote this routine some years ago for an article I was going to submit to **Datafile** for publication as part of my "**Boldly going ...**" regular series of sizable articles featuring novel techniques, as it was purpose-made to achieve a "bold" goal. After a month or so I eventually finished the article, which includes the routine proper with full comments and, as usual, fully-worked-out examples *galore*, with lots of relevant comparisons and tests, caveats, etc.

Alas, things went south with **Datafile**, regrettably, and the 16-page article remains unpublished to this day so I'd rather not disclose its fine details till I eventually try and publish it somewhere else, together with about six or seven additional unpublished ones. A pity indeed because the finished article looks awesome, even if I say so myself. As a policy, I always write the kind of articles I'd love to read myself.

Thanks for your understanding and

Best regards from V.

Re: Another tough integral

Message #25 Posted by **W. Bruce Maguire II** on 17 May 2012, 1:32 p.m.,
in response to message #24 by Valentin Albillo

Valentin wrote:

Quote:

As for details on my own quadrature routine which I mentioned a few posts ago, I'm sorry but I'll keep them private for the time being ... I'd rather not disclose its fine details till I eventually try and publish it somewhere else...

You have **got** to be kidding!

What sort of reward do you envision for "publishing" your programmable-calculator code (outside of this forum)? And where might you "publish"? Do you really believe that there is a big, receptive audience outside of this forum?

Bruce.

Re: Another tough integral

Message #26 Posted by **Steve Perkins** on 17 May 2012, 4:27 p.m.,
in response to message #24 by Valentin Albillo

Since you've gone to a lot of work, and hope to publish, I certainly respect your choice to keep it private for now.

I was kind of hoping it might be appropriate for use in calculator circles, and it appears that this is the case!

In the meantime, my interest in numerical integration has been re-kindled. I've already found ways to speed up my variable precision routine by a factor of 3, so that has been worthwhile.

It's not so much finding test cases that a given routine can't handle that bothers me, but finding cases which the routine claims to handle with a reported error estimate, but is way off.

My code handles cases with singularities (integrated from 0 to 1) like:

$1/\sqrt{x}$

$\log(x)*\sqrt{x}$

without much problem. Also (integrating say from 0.01 to 1):

$\sin(1/x)/(x*x)$

also fairly readily. I'm always looking for more interesting test cases as have been presented here. Hopefully we can soon have calculator routines at least as good as Casio's available to us on our HP machines.

As always, I greatly appreciate your input to the forum.

Re: Another tough integral

Message #27 Posted by **Gerson W. Barbosa** on 17 May 2012, 5:07 p.m.,
in response to message #26 by Steve Perkins

Quote:

I'm always looking for more interesting test cases as have been presented here.

Have you tried Kahan's integral already?

$$I = \int_0^1 (\sqrt{x}/(x-1) - 1/\ln(x)) dx$$

It has the advantage of having a closed-form solution, as shown [here](#):

$$I = 2 - \ln(4) - \text{EulerGamma}$$

$$I = 0.036489973978576520559023667001244432806840395339565892952872746128345029282945897851326282715415875401$$

An alternate form is

$$I = \text{Psi}[3/2]$$

which can be checked at [W|A](#).

Re: Another tough integral

Message #28 Posted by [Mike Morrow](#) on 18 May 2012, 2:14 p.m.,
in response to message #27 by Gerson W. Barbosa

At the risk of boring folks with more fx-115ES Plus results, I report that Dr. Kahan's classic HP-34C test integral produces for once slightly mixed results. Adding 1E-12 offsets to both limits of integration to eliminate singularity at those points during the Gauss-Kronrod process, the 15-digit result after 121 seconds is:

0.0364899739857302

The actual displayed value is 0.03648997399, so only the first 9 significant digits of the result are correct...the last should have been 8.

Still, this is a much better result in far less time than any HP calculator I've used. This includes the HP 50G, which has much faster hardware producing results *for other problems* about 20 times faster than does the 115ES Plus. This is a trivial-cost high school student's calculator. I can't help being impressed with the algorithm engineering at Casio, and a little disappointed by HP's in this area. It's way past time for Gauss-Kronrod quadrature to be an option in HP scientific calculator firmware.

Edited: 18 May 2012, 2:27 p.m.

Re: Another tough integral

Message #29 Posted by [Steve Perkins](#) on 18 May 2012, 4:30 p.m.,
in response to message #28 by Mike Morrow

I'm surprised you had to modify the end-points.

My understanding is that Gauss-Kronrod does not need to evaluate the function at the extremes. That's part of it's usefulness.

My implementation of Gauss-Kronrod also comes up with a reasonable approximation of the integral, but the error estimate isn't as good as I would hope. Overly optimistic I guess I might say.

A good test case.

Re: Another tough integral

Message #30 Posted by [Steve Perkins](#) on 18 May 2012, 4:19 p.m.,
in response to message #27 by Gerson W. Barbosa

My double exponential quadrature routine with variable precision handles that one quite nicely too. I incorporated it into my test suite when I read about it in this forum.

Thanks for the suggestion. That function does have interesting behavior at certain spots.

Re: Another tough integral

Message #31 Posted by [Kiyoshi Akima](#) on 17 May 2012, 6:36 p.m.,
in response to message #26 by Steve Perkins

Another simple integral with a closed-form solution is the integral of $\text{FRAC}(x)$ from 0 to 6.4. Every HP handheld calculator to date with a built-in integrator produces a >50% relative error (or >100% depending on how you define relative error). The Casio fx-115ES Plus gets it exact.

I do wish you would ... Re: Another tough integral

Message #32 Posted by [gene wright](#) on 17 May 2012, 7:47 p.m.,
in response to message #24 by Valentin Albillo

Submit your articles to HP Solve. I know it is PDF, but it would reach a great deal more people than Datafile ever has.

Be good if more people would learn about Valentin!

Re: I do wish you would ... Re: Another tough integral

Message #33 Posted by [Valentin Albillo](#) on 18 May 2012, 7:22 p.m.,
in response to message #32 by gene wright

Quote:

Submit your articles to HP Solve. I know it is PDF, but it would reach a great deal more people than Datafile ever has.

Be good if more people would learn about Valentin!

Thanks for your always kind words and your continued appreciation, Gene.

The problem is, I already have the article written as a PDF document formatted per the mandatory Datafile template in use at the time and reformatting it per HP Solve guidelines would surely be a lot of work and thus a lot of additional time I simply don't have.

Also, I have two versions of the routine, one of them is written for and runs in a non-HP multiprecision environment and thus is likely to be unsuitable for HP Solve. The other version is a standard-precision HP-71B program but I think that HP Solve will probably be much more interested in publishing materials for newer, current, easily available models than for hard-to-get deprecated old timers like the HP-71B.

Anyway, thanks again for your interesting, well-meant advice and

Best regards from V.

Re: Another tough integral

Message #34 Posted by [Gerson W. Barbosa](#) on 13 May 2012, 11:41 p.m.,
in response to message #3 by Valentin Albillo

Quote:

Well, it doesn't seem that tough at all

Well, it is tough enough for all HP calculators I tried. For instance, on the ultra fast HP-15C LE it apparently will run forever when FIX mode is set to 8 or 9. Mike Morrow's Casio fx-115ES Plus below has 15 internal digits, but will get only the first 11 digits right. It is very fast and all 10 digits in the display will be correct, however.

Best regards,

Gerson.

Re: Another tough integral

Message #35 Posted by [Mike Morrow](#) on 14 May 2012, 1:32 a.m.,
in response to message #34 by Gerson W. Barbosa

The interesting thing about the speed of the results from the fx-115ES Plus (aka 991ES Plus C) is that the equivalent of a 2500-iteration Savage benchmark takes 23 times longer on the Casio than on the HP 15C-LE. The magic in the Casio's quadrature result comes from the algorithm, not the hardware.

Edited: 14 May 2012, 1:51 a.m.

Re: Another tough integral

Message #36 Posted by [Mike Morrow](#) on 13 May 2012, 9:49 p.m.,
in response to message #1 by Gerson W. Barbosa

The Casio fx-115ES Plus uses a Gauss-Kronrod quadrature method and requires the function to be evaluated at the endpoints. Adding 1E-12 to each endpoint and setting a tolerance value of 1E-14 yields the following result after 63.0 seconds:

0.995555497761582

What shows up in the 10-digit display is 0.9955554978, which is accurate for every digit.

That's not too shabby for an \$18 calculator.

Edited: 13 May 2012, 10:17 p.m.

Re: Another tough integral

Message #37 Posted by [Gerson W. Barbosa](#) on 14 May 2012, 6:17 a.m.,
in response to message #36 by Mike Morrow

Quote:

That's not too shabby for an \$18 calculator.

Agreed! One of these days I had the band of my wristwatch replaced at a Casio representative and saw one of these on the shelf. Coincidence or not the price tag was exactly R\$ 115 (~US\$ 57.50), what made me lose the interest.

Re: Another tough integral

Message #38 Posted by [Bart \(UK\)](#) on 14 May 2012, 9:59 a.m.,
in response to message #1 by Gerson W. Barbosa

www.numberempire.com gives:

[.9955554977623585](#)

Re: Another tough integral

Message #39 Posted by [Gerson W. Barbosa](#) on 14 May 2012, 5:41 p.m.,
in response to message #38 by Bart (UK)

Thanks for the link! Previously I had tried [this link](#), which offers various integration methods, but I wasn't able to make it work with this example. I tried the WP 34S (v2) again and divided the integration interval into three subintervals, [0,0.1], [0.1,0.9] and [0.9,1], and obtained 0.9955554977624464. Only slightly closer though.

P.S.: The following is better and will give 14 correct digits:

WP 34S v2:

```
001 LBL 00
002 # pi
003 *
004 COS
005 x^2
006 ACOS
007 x<>y
008 # pi
009 *
```

010 SIN
 011 /
 012 RTN

0 ENTER .5 g Integral 00 2 * 3 / .5 + --> 0.9955554977623607

Edited: 14 May 2012, 9:30 p.m.

Re: Another tough integral

Message #40 Posted by **Bart (UK)** on 16 May 2012, 3:24 a.m.,
 in response to message #39 by Gerson W. Barbosa

That is an interesting link, I had not come across it. I like the choice of different methods too.

Re: Another tough integral

Message #41 Posted by **Les Wright** on 14 May 2012, 10:46 p.m.,
 in response to message #1 by Gerson W. Barbosa

Gerson, you seem to have a glitch in the code for your functions program. Once you consume your x with the cos(Pi x) calculation, etc, where are you getting a second x for the denominator?

This is what I used:

```
LBL 'SS'
# Pi
*
FILL
COS
STO* X
ACOS
SWAP
SIN
3
*
/
.
5
+
RTN
```

I use my own integrator on the 34S, which is a simple port of the PPC IG routine. My routine requires that only the last two extrapolated estimates agree, whereas the internal routine requires three and this means things take much longer.

With my routine, on a WP34S with the latest firmware, I get 9.95555497791e-1 at SCI 11. This is off just one ULP in the 12th digit.

The integral isn't that pathological and deserves another look with a routine that computes it correctly.

Les

Re: Another tough integral

Message #42 Posted by **Les Wright** on 14 May 2012, 10:59 p.m.,
 in response to message #41 by Les Wright

PS: My batteries are getting sluggish so the WP34S built-in integration routine in the first surge and shuts the calc off and I am too lazy to dig out fresh ones at the moment.

However, in the most recent emulator set to ALL 00 and DP, I not only get the first 12 digits exactly, but internally the routine gives us 6 more, to 18 in total. And the emulator does so instantaneously, which means this will not take forever on the real calc. And considering that the calc's routine picks the accuracy according to display mode (ALL 00 is treated as SCI 11), getting 6 more digits is darned impressive.

Les

Edited: 14 May 2012, 11:00 p.m.

Re: Another tough integral

Message #43 Posted by [Valentin Albillo](#) on 14 May 2012, 11:08 p.m.,
in response to message #41 by Les Wright

Quote:

With my routine, on a WP34S with the latest firmware, I get 9.95555497791e-1 at SCI 11. *This is off just one ULP in the 12th digit.*

Nope, it's 3 units off in the 11th digit, see my 1,000+ decimal places result above which begins with:

0.995555497762367822986776561111752231600962645962...

Regards. V.

Re: Another tough integral

Message #44 Posted by [Les Wright](#) on 15 May 2012, 12:12 a.m.,
in response to message #43 by Valentin Albillo

Sorry, typo. 11th digit in mine should be a 6, not a 9.

My claim is right. I transcribed my result incorrectly.

Les

Re: Another tough integral

Message #45 Posted by [Gerson W. Barbosa](#) on 14 May 2012, 11:16 p.m.,
in response to message #41 by Les Wright

Hello Les,

Quote:

Once you consume your x with the cos(Pi x) calculation, etc, where are you getting a second x for the denominator?

From the stack, which is automatic filled with the content of the X register every time the function to be integrated is called. From the manual:

"Lower and upper integration limits must be supplied in Y and X, respectively. Otherwise, the user interface is as in HP-15C. Please refer to the *HP-15C Owner's Handbook*, Section 14 and Appendix E, for more information about automatic integration and some caveats."

Quote:

With my routine, on a WP34S with the latest firmware, I get 9.95555497791e-1 at SCI 11. This is off just one ULP in the 12th digit.

By simplifying the integrand prior to integration I was able to obtain 14 correct digits. Please see my reply to Bart above.

Quote:

The integral isn't that pathological and deserves another look with a routine that computes it correctly.

I think you're right. The transitions at integer values of x appear to be fast enough to be a problem.

Regards,

Gerson.

Re: Another tough integral

Message #46 Posted by [Alexander Oestert](#) on 15 May 2012, 3:55 p.m.,
in response to message #1 by Gerson W. Barbosa

Please, someone give me a hint on how long this takes on the 34s and the 15c LE. It seems to take forever and I don't want to wear out my batteries.

Re: Another tough integral

Message #47 Posted by [Gerson W. Barbosa](#) on 15 May 2012, 4:34 p.m.,
in response to message #46 by Alexander Oestert

Hello Alexander,

On the WP 34S (v2) it takes only a couple of seconds. On the HP-15C you should set FIX mode to 7 at most. Alternatively use the following program:

```
001 LBL A
002 pi
003 *
004 COS
005 x^2
006 ACOS
007 x<>y
008 pi
009 *
010 SIN
011 /
012 RTN
```

```
0 ENTER .5 g Integral A 2 * 3 / .5 + f --> 0.995555504
```

It will take less than 3 seconds on the HP-15C LE, even when FIX mode is set to 9.

Cheers,

Gerson.

Re: Another tough integral

Message #48 Posted by [Alexander Oestert](#) on 15 May 2012, 4:52 p.m.,
in response to message #47 by Gerson W. Barbosa

Duh! Switching to RAD did it on the WP-34s V2 and the 15c LE, but on the WP-34s V3 2931 I can't get it to work.

EDIT: Solved on the WP-34s V3 by explicitly setting an END at the end instead of terminating only by RTN. I didn't know that.

Edited: 16 May 2012, 2:10 a.m.

Re: Another tough integral

Message #49 Posted by [Marcus von Cube, Germany](#) on 17 May 2012, 1:20 a.m.,
in response to message #48 by Alexander Oestert

This sounds strange. Can you send me an email with details so that I can have a closer look when I find the time?

Strange [WP-34s] behaviour - Re: Another tough integral

Message #50 Posted by [Alexander Oestert](#) on 18 May 2012, 2:02 a.m.,
in response to message #49 by [Marcus von Cube, Germany](#)

I can't remember what state the calculator was in when I encountered this, so I took a recently converted machine (VERS 3049, only calc.bin) and did a RESET.

Set it to [g][RAD]

Enter the WP-34s program from posting #1, exactly the 17 steps, no less, no more

0 [ENTER] 1 [g][2] 0 0

For a short time I see 'Running Program' then the integral solving symbol appears and the $9.9555497761E-1$ converges. So far so good. Because the RCL annunciator is still flashing, I [EXIT]. I key in again

0 [ENTER] 1 [g][2] 0 0

Now I see 'Running Program' and the RCL flashing, no integral symbol. After about 2 minutes RCL flashing is off, the matrix display shows 'Reset' and $9.96943604201E-1$ is in the lower display. Another

0 [ENTER] 1 [g][2] 0 0

has the machine running for 5 minutes before I forcibly [EXIT] with 'Stopped' and $4.72450120691E-1$ in the display.

Now I do [GTO][.][.] and enter the following program

001 LBL A

002 [g][x^2]

003 5

004 -

005 [g][RTN]

Then [f][SLV] brings up -2.2360679775 in the display.

Again

0 [ENTER] 1 [g][2] 0 0

does a fast conversion, with RCL still flashing when I [EXIT]. The next

0 [ENTER] 1 [g][2] 0 0

has to be aborted after 5 minutes with [EXIT].

[SLV]ing program A from above results in 'Running Program'. I end the experiment here.

Marcus, I send you this as an email but post this here so everybody can have fun... :-)

Re: Strange [WP-34s] behaviour - Re: Another tough integral

Message #51 Posted by [Paul Dale](#) on 18 May 2012, 2:19 a.m.,
in response to message #50 by [Alexander Oestert](#)

I'd be guessing that the return stack isn't being cleared each time and that is confusing things. Try a g-shirt RTN in run mode to clear things.

Still, it seems like a bug/annoyance.

- Pauli

Re: Strange [WP-34s] behaviour - Re: Another tough integral

Message #52 Posted by [Marcus von Cube, Germany](#) on 18 May 2012, 2:49 a.m.,
in response to message #50 by Alexander Oestert

Pauli is right, you need to do do g-RTN before you can try again. What happens is that you are effectively nesting calls to he integrator. EXIT stops the process while in your user code routine to integrate. You can R/S to continue. If, instead, you start a new integration operation this will nest. If you stop it again, it will eat more and more of the available memory.

There is still something wrong. Showing "Running Program" is not intended. "Reset" is most probably the watch dog kicking in.

Re: Strange [WP-34s] behaviour - Re: Another tough integral

Message #53 Posted by [Alexander Oestert](#) on 18 May 2012, 3:34 a.m.,
in response to message #52 by Marcus von Cube, Germany

[g][RTN] does it.

I just wanted to mention, if I enter program mode after an integral run and without [g][RTN], memory is only about 415 steps after the first try and 320 steps after the second try, 225 after 3rd...

Re: Strange [WP-34s] behaviour - Re: Another tough integral

Message #54 Posted by [Paul Dale](#) on 18 May 2012, 3:46 a.m.,
in response to message #53 by Alexander Oestert

About right. The integration code allocates a lot of local registers.

We probably should clear the locals/return stack when solve or integrate is run from the keyboard....

- Pauli

Re: Strange [WP-34s] behaviour - Re: Another tough integral

Message #55 Posted by [Marcus von Cube, Germany](#) on 18 May 2012, 1:58 p.m.,
in response to message #54 by Paul Dale

Quote:

We probably should clear the locals/return stack when solve or integrate is run from the keyboard....

That's what I came up with on the last kilometers of my latest bike trip. Let me see when I find the time or a clean implementation. The "Running Program" is indeed intentional because the software detects that it is not run from the top level and is therefore quiet (no display of intermediate results). This leaves the original "Running" message in the display. I'm still thinking about the "Reset" case. I have an idea how to fix this (if it's the watchdog) but I must check if I'm right.

Re: Strange [WP-34s] behaviour - Should be better now. :-)

Message #56 Posted by [Marcus von Cube, Germany](#) on 19 May 2012, 3:21 a.m.,
in response to message #55 by Marcus von Cube, Germany

I did something about the issue. Please report any findings!

Re: Strange [WP-34s] behaviour - Re: Another tough integral

Message #57 Posted by [fhub](#) on 18 May 2012, 4:06 a.m.,
in response to message #53 by Alexander Oestert

Quote:

I just wanted to mention, if I enter program mode after an integral run and without [g][RTN], memory is only about 415 steps after the first try and 320 steps after the second try, 225 after 3rd...

I guess this happens only if you abort the integral calculation with [EXIT] each time!?

I've now tried a few integrals with the emulator (last build) but I don't get this problem. Of course I couldn't interrupt the integration because the emulator is just too fast.

Pauli has already posted the solution: clearing the return stack (and local regs) for a manual call should solve the problem. This should also be done for SUM and PROD (not only INTG and SLV), I don't know if there are other such routines which call a user program.

Franz

Edited: 18 May 2012, 4:07 a.m.

Re: Another tough integral

Message #58 Posted by [Gerson W. Barbosa](#) on 19 May 2012, 9:49 p.m.,
in response to message #46 by Alexander Oestert

Quote:

It seems to take forever and I don't want to wear out my batteries.

Same here with versions 3.0 2609 and 3.1 2988. Has it been fixed? Thanks!

Re: Another tough integral

Message #59 Posted by [Marcus von Cube, Germany](#) on 20 May 2012, 2:25 a.m.,
in response to message #58 by Gerson W. Barbosa

The adaptive integrator in WP 34S V3 takes a few seconds even on the emulator so be prepared for some delay. I have no idea what the END/RTN issue is.

Re: Another tough integral

Message #60 Posted by [Gerson W. Barbosa](#) on 20 May 2012, 5:59 p.m.,
in response to message #59 by Marcus von Cube, Germany

After 25 minutes it is showing 9.95555498002^{-1} for five or six minutes now, diverging from a previous closer result. A useful feature here would be the possibility of interrupting the calculation whilst keeping the latest approximation on the display instead of the not meaningful result I get when I eventually abort it, 1.54227330201 (v3.1 2988).

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



Go back to the main exhibit hall