

HP Forum Archive 16

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

HP-15C Mini-challenge

Message #1 Posted by [Valentin Albillo](#) on 25 Sept 2006, 10:01 a.m.

Hi all,

I don't like Mondays, but let's have a positive attitude and start the week with a quick-and-easy brain teaser to flex your HP neurons a little bit, by solving this nifty HP-15C Mini-challenge (tm) o'mine:

Imagine you've just got a mint HP-15C from eBay, a real bargain at just \$687.99, which happens to be absolutely mint (except for those large initials carved in the LCD display glass) and also works superbly, except for the fact that the ROM seems to be slightly damaged and no trigonometric functions work, including both direct (SIN, COS, ...) and inverse (SIN^{-1} , COS^{-1} , ...), whether plain circular or hyperbolic. Most regrettably, this also extends to the Polar <> Rectangular conversions, which refuse to work either. Also, as a minor inconvenience, complex mode doesn't work at all, but never mind as you don't care for complex number calculations.

This being so, and as you badly need to compute lotsa trigonometrics with this fine machine, you set to the task of implementing a bootstrap $\sin(x)$ subroutine, which you can then use to compute all the rest. So, in brief, your task is:

"Try and write a routine to compute $\text{Sin}(x)$, for x given in radians in the interval $[-3, +3]$, say, at the very least, to full 10-digit accuracy (save a few rounding-induced ulps at most), provided you can't use neither any built-in trigonometric functions, direct or inverse, circular or hyperbolic, nor any complex mode calculations whatsoever. Oh, and the faster the better !"

Easy, right ? :-) I fully expect a number of you to solve it within a few hours at most. In any case, I'll post *my original 12-step solution* (including LBL and RTN) within a few days. By pre-storing a constant, it can be made to fit *in just 10 steps* (LBL and RTN always included) and run slightly faster as well.

Best regards from V.

Re: HP-15C Mini-challenge

Message #2 Posted by [Gerson W. Barbosa](#) on 25 Sept 2006, 12:09 p.m.,
in response to message #1 by [Valentin Albillo](#)

Hello Valentin,

Here is an unoptimized version:

```
f LBL A
g PI
 /
 1
 -
f x!
g LASTx
 1
 +
 CHS
f x!
 *
g PI
 /
 1/x
g RTN
```

Reference: <http://mathworld.wolfram.com/GammaFunction.html> Formula (41)

Best regards,

Gerson.

Re: HP-15C Mini-challenge

Message #3 Posted by [Gerson W. Barbosa](#) on 25 Sept 2006, 1:25 p.m.,
in response to message #1 by [Valentin Albillo](#)

Hi, Valentin!

That's better :-)

01 fLBL A 02 ENTER 03 ENTER 04 g PI 05 / 06 fX! 07 g LASTx 08 CHS 09 fX! 10 * 11 / 12 g RTN

Too bad the 12C doesn't have gamma built-in... (unless we get one of those nice Victor Toth's implementations...)

Best regards,

Gerson.

Re: HP-15C Mini-challenge

Message #4 Posted by [Egan Ford](#) on 26 Sept 2006, 2:09 a.m.,
in response to message #3 by [Gerson W. Barbosa](#)

Quote:

Too bad the 12C doesn't have gamma built-in...

My first attempt at Gamma for the 12C (Classic) using Stirling's approximation. I expect others can improve on this.

Only 43 steps. $n > 0$. More accurate for larger n , but fast. Given the default FIX 2 of the 12C this may be accurate enough. :-) A check for if FRAC = 0 then use $n!$ should be added.

Unfortunately this implementation does not support $n < 0$, so it cannot be used to fix a "broken" 12C SIN key.

n	gamma	12c gamma
0.2	4.5908	29.9798
1.1	0.9514	0.9512
2.1	1.0464	1.0465
3.15	2.3069	2.3070
10.187	553895.9150	553895.7118

```
STO 0
6
Y^X
8
1
0
*
1/X
RCL 0
1/X
+
e^X
LSTx
CHS
e^X
-
2
/
RCL 0
*
SQRT
RCL 0
```

```

1
e^X
/
*
RCL 0
Y^X
2
ENTER
3
.
1
4
1
5
9
*
RCL 0
/
SQRT
*
GTO 00

```

Edited: 26 Sept 2006, 2:17 a.m.

Re: HP-15C Mini-challenge

Message #5 Posted by **Gerson W. Barbosa** on 26 Sept 2006, 6:10 p.m.,
in response to message #4 by Egan Ford

Quote:

Given the default FIX 2 of the 12C this may be accurate enough. :-)

That's exactly what I thought when I wrote this:

[Hyperbolic Approximations for Trigonometric Functions on the HP-12C](#)

Actually, when compared to the exact result, $(\sqrt{5}-1)/4$, $\sin(18) = 0.31$ is as bad as $\sin(18) = 0.3090169943749474$:-)

Thanks for sharing your 12C Gamma solution with us. Perhaps you'd like to take a look at Viktor Toth's program:

<http://www.rskey.org/detail.asp?manufacturer=Hewlett-Packard&mode=HP-12C>

Keep in mind he's a master on the subject though :-)

That has made me wonder: had the 12C Platinum a fast built-in Gamma implementation (no other key position needed, just relabeling n! as x!) we'd get trigs at low computational cost...

Best regards,

Gerson.

Edited: 26 Sept 2006, 8:07 p.m.

12C/12CP x! (Was: Re: HP-15C Mini-challenge)

Message #6 Posted by **Egan Ford** on 27 Sept 2006, 4:10 p.m.,
in response to message #5 by Gerson W. Barbosa

Quote:

Perhaps you'd like to take a look at Viktor Toth's program:

<http://www.rskey.org/detail.asp?manufacturer=Hewlett-Packard&mode=HP-12C>

Looks like a similar example from Numerical Recipes. The problem I have with it is that if you use your 12C for anything else that uses registers (e.g. statistics), then you will lose your constants. Entering the constants appears to be as laborious as entering the program itself.

I offer an alternative with only one constant (PI). Anyone that is seeking $x!$ for real numbers surely knows PI to more than a few digits. :-) Just store PI in register 6. The more digits you can remember the more digits of accuracy you will get. For the 12CP it is ~ 1:1 ratio.

Unlike the above link this is an $x!$ implementation, not $\gamma(x)$. It did this to match the 15C. $\gamma(x)=(x-1)!$

The implementation below uses at its core the following from Viktor Toth's web site:

$$x! = x^x \sqrt{2\pi x} e^{1/[12x + 2/(5x + 5/4x)] - x} \quad (1)$$

I selected this because it was keystroke friendly and it can use a unique feature of the 12C--the 12x key. It is fairly accurate for numbers >8 , but like my previous code above does poorly with $n < 8$ and does not support negative numbers. To address the accuracy for $n < 8$ I add 8, then use the follow to get the actual accurate number:

$$x! = (x+n)! / [(x+1)(x+2)\dots(x+n)], \text{ where } n=8. \quad (2)$$

This can be best illustrated with numbers close to 0, e.g.:

```
15C          0.0001! = 0.999942288
EQ(1)       0.0001! = 5.405375606E317
EQ(1),EQ(2) 0.0001! = 0.9999422882 (50g implementation)
```

Lastly to address $(-x)!$ I used:

$$(-x)! = x\pi/x! \sin(x\pi)$$

To implement this I used Valentin Albillo's 12C SIN routine.

To test I generated 10 random numbers on my 15C from -50 to 50.

Results (space delimits 12C/12CP results where they differ from 15C):

x	15C (1985)	12C (1981)	12CP
-17.5248	-3.429996115 E-14 (6s)	-3.429996 072 E-14 (34.5s)	-3.429996115 E-14 (14 s)
36.8521	8.054944842 E 42 (2s)	8.054944 587 E 42 (6 s)	8.05494484 3 E 42 (2 s)
-22.1097	1.299530964 E-19 (7s)	1.29953096 2 E-19 (17 s)	1.299530964 E-19 (6 s)
-13.8499	-0.000000002 (5s)	-0.000000002 (38.5s)	-0.000000002 (16.5s)
-39.2643	-3.097787591 E-45 (9s)	-3.097787 711 E-45 (32.5s)	-3.09778759 0 E-45 (13.5s)
-23.7123	-3.829069924 E-22 (6s)	-3.829069 864 E-22 (35 s)	-3.82906992 6 E-22 (15 s)
42.5232	1.002419255 E-52 (2s)	1.00241925 7 E-52 (6 s)	1.002419255 E-52 (2 s)
5.9853	700.4604360 (2s)	700.46043 33 (10.5s)	700.460436 1 (4 s)
-15.9205	-1.208388189 E-11 (5s)	-1.2083881 39 E-11 (39 s)	-1.2083881 93 E-11 (16 s)
-2.2782	3.556921585 (3s)	3.5569215 85 (23 s)	3.556921585 (9 s)

Exact same code on 12C and 12CP (exception, 12CP GTO needs a leading 0, LSTx is on +). The 12C had no less than 7 digits of accuracy, the 12CP no less than 8. 12CP is more accurate and faster.

Code:

```
1 STO 5
2 2
3 Y^X
4 SQRT
5 STO 4
6 STO 3
7 8
8 X<=Y?
9 0
10 STO+ 4
11 RCL 4
12 RCL 4
13 Y^X
14 RCL 6
15 RCL 4
```

```
16 *
17 2
18 *
19 SQRT
20 *
21 4
22 RCL 4
23 *
24 1/X
25 RCL 4
26 +
27 5
28 *
29 2
30 /
31 1/X
32 RCL 4
33 12X
34 +
35 1/X
36 RCL 4
37 -
38 E^X
39 *
40 STO 2
41 RCL 3
42 8
43 X<=Y?
44 GTO 56
45 8
46 STO 4
47 RCL 3
48 +
49 STO/ 2
50 RCL 4
51 1
52 -
53 X=0?
54 GTO 56
55 GTO 46
56 RCL 5
57 0
58 X<=Y?
59 GTO 98
60 RCL 3
61 2
62 /
63 FRAC
64 2
65 *
66 RCL 6
67 *
68 STO 0
69 1
70 STO 1
71 RCL 0
72 RCL 0
73 CHS
74 STO 0
75 RCL 1
76 2
77 +
78 STO 1
79 Y^X
80 LSTx
81 n!
82 /
83 +
84 -
85 X=0?
86 GTO 89
```

87 LSTx
 88 GTO 72
 89 LSTx
 90 RCL 2
 91 *
 92 RCL 3
 93 RCL 6
 94 *
 95 X<>Y
 96 /
 97 GTO 00
 98 RCL 2
 99 GTO 00

Edited: 27 Sept 2006, 5:06 p.m.

Re: 12C/12CP x! (Was: Re: HP-15C Mini-challenge)

Message #7 Posted by [Gerson W. Barbosa](#) on 27 Sept 2006, 5:58 p.m.,
 in response to message #6 by Egan Ford

That's really nice! And you have a step left for a future upgrade :-)

I have keyed it in my HP-12C. Now, when I want sines in degrees I just have to type the following (this can be improved, I think) :

ENTER 180 / STO PMT R/S STO FV RCL PMT CHS R/S RCL FV * RCL PMT RCL 6 * / 1/x

Re: 12C/12CP x! (Was: Re: HP-15C Mini-challenge)

Message #8 Posted by [Egan Ford](#) on 27 Sept 2006, 11:11 p.m.,
 in response to message #7 by Gerson W. Barbosa

Quote:

That's really nice! And you have a step left for a future upgrade :-)

Thanks. It was a rush job. Perhaps in the future I will try to trim it down or try another method.

Re: 12C/12CP x! (Was: Re: HP-15C Mini-challenge)

Message #9 Posted by [Gerson W. Barbosa](#) on 28 Sept 2006, 9:52 a.m.,
 in response to message #8 by Egan Ford

And please don't forget to submit it to the Articles Forum. My interest in implementations of scientific functions on the HP-12C began last year when I purchased an 12C as a portable RPN calculator option. I don't need the 12C anymore however these implementations are interesting and might be useful to some people.

Best regards,

Gerson.

Re: HP-15C Mini-challenge

Message #10 Posted by [Gerson W. Barbosa](#) on 25 Sept 2006, 6:06 p.m.,
 in response to message #1 by Valentin Albillo

Hello again, Valentin!

I was in a hurry I forgot to format the listing before posting. As I cannot edit my previous post, this is to do it:

001 f LBL A
 002 ENTER

```
003 ENTER
004 g PI
005 /
006 f x!
007 g LST x
008 CHS
009 f x!
010 *
011 /
012 g RTN
```

It works as specified in the range]-pi, pi[.

Quote:

Oh, and the faster the better !

On my double-speed HP-15C it runs in 3.1 seconds (6.8 s on the HP-34C and about half a second on the HP-32SII).

Thanks for this really nifty mini-challenge!

Gerson.

Edited: 25 Sept 2006, 8:15 p.m.

Re: HP-15C Mini-challenge

Message #11 Posted by [Eamonn](#) on 25 Sept 2006, 9:03 p.m.,
in response to message #10 by Gerson W. Barbosa

Very nice.

Re: HP-15C Mini-challenge

Message #12 Posted by [Gerson W. Barbosa](#) on 25 Sept 2006, 9:48 p.m.,
in response to message #11 by Eamonn

Thanks, Eamonn!

I was kind of lucky though. That had to do with the gamma function, I guessed since the other few possibilities left wouldn't suffice for so small a program. Coincidentally I had taken a look at the reference in my first post just some days ago. So I knew where the answer to Valentin's challenge might be...

Gerson.

Re: HP-15C Mini-challenge

Message #13 Posted by [Namir](#) on 26 Sept 2006, 3:05 a.m.,
in response to message #10 by Gerson W. Barbosa

Very clever!!!!!!!!!!!!

Re: HP-15C Mini-challenge: My solutions & Comments

Message #14 Posted by [Valentin Albillo](#) on 26 Sept 2006, 4:50 a.m.,
in response to message #10 by Gerson W. Barbosa

Hi, Gerson:

Congratulations, your solution *exactly* duplicated my original 12-step solution ! As I stated, with such people as you contributing to this forum it's become harder and harder to concoct any kind of challenge able to resist unsolved even for a few hours, which is great !

My original solution was, of course, the very same you came up with, namely:

```

1  LBL A
2  ENTER
3  ENTER
4  PI
5  /
6  x!
7  LASTX
8  CHS
9  x!
10 *
11 /
12 RTN

```

which is based on this amazing, somewhat unexpected formula:

$$\sin(x) = \frac{x}{(x/\pi)! * (-x/\pi)!}$$

This routine gives correct results for all X arguments within very wide ranges, not only for [-3,+3], but Pi or exact multiples of Pi result in a blinking display, as gamma is evaluated for a negative integer, resulting in Overflow. The final returned result (0) is correct, however.

By pre-storing the constant 1/Pi in register I, say (PI, 1/X, STO I), the program can be two steps shorter and slightly faster:

```

1  LBL A
2  ENTER
3  RCL*I
4  x!
5  LASTX
6  CHS
7  x!
8  *
9  /
10 RTN

```

which also has the added advantage of not giving an *Overflow* error when given the arguments Pi or -Pi, which results in a blinking display (but correct result) in the previous routine. As the pre-stored constant is 1/Pi, which is less than one, you could also store it in the RAN# register, thus using no regular register at all, but that wouldn't save program steps and, apart from the fact that it doesn't blink for x=Pi or -Pi, I think the stack-only 12-step solution is preferable.

By the way, since exponentials and logarithms can be expressed as trigonometrics and inverse trigonometrics of complex arguments, and since trigonometrics can be expressed in terms of the gamma function or its inverse, it's the case that ultimately we could reduce them all to expressions involving exclusively the gamma function and its inverse.

This means that a complex implementation of gamma and its inverse could be used internally to replace pages and pages of ROM microcode devoted to compute the 6 trigonometric functions plus the 6 hyperbolic functions plus exponentials and logarithms ! What an enormous saving ! ;-)

I'm glad you liked this mini-challenge, thanks for your interest and

Best regards from V.

Edited: 26 Sept 2006, 7:07 a.m. after one or more responses were posted

Re: HP-15C Mini-challenge

Message #15 Posted by **Marcus von Cube, Germany** on 26 Sept 2006, 6:34 a.m.,
in response to message #14 by Valentin Albillo

Hi Valentin,

Quote:

This means that a complex implementation of gamma and its inverse could be used internally to replace pages and pages of ROM microcode devoted to compute the 6 trigonometric functions plus the 6 hyperbolic functions plus exponentials and logarithms ! What an enormous saving ! ;-)

That might be more difficult than it looks, because the various definitions of Gamma seem all to include logs and/or exponentials.

BTW, the challenge made me play around with complex Gamma and the only computer software, I have access to, that supports it is "Derive 6" on the PC. Neither "Grapher" nor "Graphing Calculator" for Mac OS can evaluate Gamma for nonreal arguments.

The HP 15C doesn't support factorials (and hence Gamma) for complex arguments, it just takes the factorial of the real part. Modern CAS offerings like the 50g or Voyage 200 happily compute the function for complex arguments while the HP 48G doesn't (no GAMMA function included, just the factorial).

Marcus

Re: HP-15C Mini-challenge

Message #16 Posted by [Valentin Albillo](#) on 26 Sept 2006, 7:23 a.m.,
in response to message #15 by Marcus von Cube, Germany

Hi, Marcus:

Marcus posted:

"That might be more difficult than it looks, because the various definitions of Gamma seem all to include logs and/or exponentials."

Yes, I know, my comment was to be taken lightly, as the ironic ;-) smiley at the end tried to indicate. Implementing a fast, accurate gamma function for any complex argument with maximum accuracy is really tricky, and using it to compute $\sin(x)$, for example, does require at least two such evaluations, which is bound to be much slower and less accurate than optimized algorithms specifically designed for $\sin(x)$.

But theoretically you could ! :-) Should you have a calculator with broken \sin , \cos , \tanh , \arcsin , \arccos , \arctan , \sinh , \cosh , \tanh , $\operatorname{arcsinh}$, $\operatorname{arccosh}$, $\operatorname{arctanh}$, \exp , \ln , 10^x , \log , $R \rightarrow P$, and $P \rightarrow R$, but working complex-defined gamma function and inverse (or else complex-capable Solve!), you would be able to make do ... !

Same applies if HP-12C's factorial function had been a gamma function *a la* HP-11C instead, you'd have sines, cosines, and tangents nearly as good as the real thing.

Best regards from V.

Re: HP-15C Mini-challenge

Message #17 Posted by [Ron Ross](#) on 26 Sept 2006, 8:15 a.m.,
in response to message #15 by Marcus von Cube, Germany

The Hp48G DOES indeed have the gamma function. It is the factorial function +1.

Re: HP-15C Mini-challenge

Message #18 Posted by [Marcus von Cube, Germany](#) on 26 Sept 2006, 9:08 a.m.,
in response to message #17 by Ron Ross

You're of course right that the function exists but it is defined for real arguments only. Viktor Toth has written a complex implementation that you can find on rskey.org:

[Programmable Calculators: Hewlett-Packard HP-48SX](#)

Marcus

Re: HP-15C Mini-challenge

Message #19 Posted by [Les Wright](#) on 26 Sept 2006, 3:54 p.m.,
in response to message #18 by Marcus von Cube, Germany

I think it is actually possible to improve on Viktor's complex gamma approximation, which is simply an implementation the version of the Lanczos approximation made famous in Numerical Recipes.

Glen Pugh did his [doctoral thesis](#) on the Lanczos approximation, and I think it is feasible to get better precision with fewer coefficients by using slightly different parameters in constructing the approximation. Pugh lists the optimal parameters in appendices to his thesis, and he also shared with me directly an extended list for series up to 100 terms. Using [Paul Godfrey's matrix version](#) of the Lanczos formula in Maple I think I can produce a version of Victor's routine that has better accuracy, or, at least, has the same accuracy but has fewer terms.

Let me work on it and report back.

Les

Re: HP-15C Mini-challenge

Message #20 Posted by [Les Wright](#) on 26 Sept 2006, 4:46 p.m.,
in response to message #19 by [Les Wright](#)

As promised, here is my revision of Viktor's routine, using parameters reported in Glen Pugh's work:

```
«
-> x
«
x 1
WHILE OVER RE 0 < REPEAT
  OVER * SWAP 1 + SWAP
END
SWAP DUP 1 + 533.368658306 SWAP /
OVER 2 + 949.861269362 SWAP / -
OVER 3 + 545.669697050 SWAP / +
OVER 4 + 112.813079042 SWAP / -
OVER 5 + 6.62153565654 SWAP / +
OVER 6 + 4.63601850621E-2 SWAP / -
1 + OVER / pi ->NUM 2 * sqrt
* LN SWAP DUP DUP 7.27950574752 + LN SWAP .5 + *
SWAP 7.27950574752 + - + EXP SWAP /
»
»
```

If Pugh's theory is correct, this version of the 7-term Lanczos approximation should get a couple of extra digits of accuracy compared to the original one reported in Lanczos' 1964 paper which Numerical Recipes and Victor both use. According to Pugh's tables, this maximized version should achieve a max relative error of about 1.4e-12 throughout the complex plane (vs. Lanczos' original report of about 2e-10), though I suspect on a 12 digit calc with coefficient rounding the actual performance may not be so impressive.

Please keep in mind that the original routine with its ingenious use of the stack is due entirely to Viktor Toth and is copyright by him. All I did was compute and plug in different coefficients. I hope this extension of Viktor's excellent work is of use to someone here.

Les

Re: HP-15C Mini-challenge

Message #21 Posted by [Gerson W. Barbosa](#) on 2 Oct 2006, 10:14 p.m.,
in response to message #20 by [Les Wright](#)

Hello Les,

Do you have 16-figure version of these coefficients so one could use to implement the function in Delphi? (Factorial can be computed up to 1754!)

Something that might be of interest: I've found out those sine and Gamma relationships may be useful to compute Gamma(x) in Excel (in case there is not anything better). I thought Excel included Gamma but the only thing I could find in the Help file was GAMMALN, which is not enough to compute the function for negative arguments. This excel formula can do the job:

```
=IF(B7>=0;EXP(GAMMALN(B7));-PI()/(B7*SIN(B7*PI()))*EXP(GAMMALN(ABS(B7))))
```

I don't know how Gamma is implemented in the HP-15C and HP-34C. I had noticed it is rather slow for negative arguments. For instance, -69.5! is computed in 15.8 seconds on the HP-34C. The user program below computes it in one third of that time:

```
LBL A
ENTER
```


Les

Re: HP-15C Mini-challenge

Message #24 Posted by [Gerson W. Barbosa](#) on 5 Oct 2006, 9:55 p.m.,
in response to message #22 by Les Wright

Thanks, Les!

I'll try it later. It will be nice to replace $n!$ with $x!$ in my Delphi calculator (just a very simple apprentice project, no complex number support). But first I have to find out what is wrong with my Pentium III computer. In radians mode, $\sin(3.141592654)$ returns $-4.10206857034707E-10$; it should return $-4.10206761624916E-10$, that's what I get on two other computers I have at home (an Athlon 1700+ and a DX4-100), using the same program. It seems I have a buggy Pentium III...

It should be noticed that the result in Delphi is not so accurate as that on HP calculators: $-4.10206761537E-10$.

Gerson

Re: HP-15C Mini-challenge

Message #25 Posted by [Ale](#) on 18 Oct 2006, 4:17 p.m.,
in response to message #24 by Gerson W. Barbosa

May be there is nothing wrong with your computer (P III), transcendental functions are not calculated directly (normally) they are calculated in a library like `libc`. The HP calcs use (as you may know) BCD arithmetic insted of binary. On the other hand, those numbers look like they are doubles (51 bit significand) and not longdoubles (64 bit significand). check if the type is correct. (Check also with the windows calculator to confirm the result).

I hope it helped

Re: HP-15C Mini-challenge

Message #26 Posted by [Gerson W. Barbosa](#) on 28 Oct 2006, 8:29 p.m.,
in response to message #25 by Ale

I had already checked the results upon the Windows Calculator:

$\sin(3.141592654)$:

Windows Calculator: $-4.1020676153735661670899289539681e-10$
 HP-48/49/50: $-4.10206761537E-10$
 Calc98: $-4.1020685703e-10$
 My own (Delphi): $-4.10206857034707E-10$

Interestingly both [Calc98](#) and my program give better results on other computers, except on mine, which makes me believe there's something wrong with it. Perhaps it's time to an upgrade :-)

Re: HP-15C Mini-challenge

Message #27 Posted by [Les Wright](#) on 5 Oct 2006, 3:20 p.m.,
in response to message #20 by Les Wright

Update:

The rounding error introduced by truncating the coefficients to 12 digits is not compensated for by the theoretically improved accuracy of the approximation, so this routine does not do much better than the seven term approximation that Viktor has already adapted from Numerical Recipes. That's a shame, because when I set the arbitrary precision to as low as 14 or 15 digits in Maple the improvement in relative error, at least for positive real arguments, is fairly impressive to behold.

That said, I was able to use Maple and Pugh's parameters to produce a five term Lanczos approximation that gives 10 or 11 digits, and at least 9 it seems, at least for positive real arguments on HP48, 49, and 49gp (haven't tested it much with complex values I am afraid). Here it is:

$\Gamma(z+1)$ approx. = $.0326489241295*(1+36.4082129181/(z+1)-32.7242118344/(z+2)+5.78003702184/(z+3)-.0840769422503/(z+4))*((z+4.84088190863)/\exp(1))^{z+1/2}$

If I understand correctly, this should only be used for positive $\text{Re}(z)$, otherwise the reflection formula that inspired this thread in the first place should be used.

Hope someone can use this. If anyone out there works in Maple, I would be glad to share the worksheet, plus the parameter and error table that Glen Pugh sent me.

Les

Re: HP-15C Mini-challenge: My solutions & Comments

Message #28 Posted by [Gerson W. Barbosa](#) on 26 Sept 2006, 12:34 p.m.,

in response to message #14 by [Valentin Albillo](#)

Hi, Valentin!

I am glad you liked my "original" solution. I just hope neither you nor Namir thinks I am smarter than I really am :-). As I said in my reply to Eammon, that was kind of a lucky finding. I am sure more people would come up with a solution if allowed more time (I had about one free hour during lunch time to give it a try).

Also, my first solution was 16 steps long. After simplifying the formula in my reference by hand I came up with a 15-step one. It was only then I remembered to use the HP-50G that was lying on my desk to get the formula you've presented. I hope this hasn't been cheating :-)

Quote:

it's become harder and harder to concoct any kind of challenge able to resist unsolved even for a few hours, which is great !

Keep on concocting them. Most of them I dare not even start solving. But we always learn from the solutions.

Best regards,

Gerson.

Re: HP-15C Mini-challenge: My solutions & Comments

Message #29 Posted by [Valentin Albillo](#) on 26 Sept 2006, 4:55 p.m.,

in response to message #28 by [Gerson W. Barbosa](#)

Hi, Gerson:

Gerson posted:

"It was only then I remembered to use the HP-50G that was lying on my desk to get the formula you've presented. I hope this hasn't been cheating :-)"

Not at all. Quite on the contrary, I welcome your ingenuity in using every helpful device at hand, and if it was the new HP model, so much the better.

"Keep on concocting them. Most of them I dare not even start solving. But we always learn from the solutions."

Thank you very much, matter of fact I do have original ideas for at least three more of them, but I'd better space them out lest they'd overstay their welcome.

Best regards from V.

Re: HP-15C Mini-challenge: My solutions & Comments

Message #30 Posted by [e.young](#) on 26 Sept 2006, 2:17 p.m.,

in response to message #14 by [Valentin Albillo](#)

Hi, this was a very interesting challenge, but I have to confess to being confused. It is my understanding that the gamma function is calculated as $(n-1)!$. In the equation you cite it looks like only the factorial of the 2 terms is being taken. Are there different definitions of gamma? What am I missing here?

Please note that I am not a mathematics "wizard" as are the posters in this thread, and in my daily work I never encounter factorials or gamma, so maybe I am asking a rudimentary question.

When I used the equation you indicate on my HP42S with its "GAM" function, I didn't get the right answer. The 42s is calculating "GAM" as $(n-1)!$. I then modified the equation as follows for use with the 42s and its "GAM" function"

$$\sin(x) = x \frac{((x/\pi)+1)! * ((-x/\pi)+1)!}{((x/\pi)+1)! * ((-x/\pi)+1)!}$$

This gives the right answer on the 42S. When I tried the equation with N! on the 42s I got an "invalid data" message.

When I use your equation on the 32Sii with its factorial function I get the correct answer.

Re: HP-15C Mini-challenge: My solutions & Comments

Message #31 Posted by [e.young](#) on 26 Sept 2006, 2:19 p.m.,

in response to message #30 by e.young

I messed up the formatting of my equation,

x is the numerator

$((x/\pi)+1)! * ((-x/\pi)+1)!$ is the denominator

Re: HP-15C Mini-challenge: My solutions & Comments

Message #32 Posted by [Gerson W. Barbosa](#) on 26 Sept 2006, 5:03 p.m.,

in response to message #31 by e.young

Quoting from the HP-15C Owner's Manual, pg 25:

Quote:

You can also use $x!$ to calculate the Gamma function, used in advanced mathematics and statistics. Pressing **f x!** calculates **Gamma(x+1)**, so you must subtract 1 from your initial operand to get **Gamma(x)**. For the Gamma function, x is not restricted to nonnegative integers.

As you have correctly noticed. That was obviously to save a key position. Not having this restriction, the HP-42S has both **!** and **GAM**.

Most of us, myself included, hardly use the Gamma function, if at all (actually I used it somewhat briefly while studying Calculus III years ago). Anyway, it pays to read Viktor Toth's passionate article on the subject (more suitable for us non pros):

[Calculators and the Gamma Function](#)

Re: HP-15C Mini-challenge: My solutions & Comments

Message #33 Posted by [Crawl](#) on 26 Sept 2006, 5:18 p.m.,

in response to message #30 by e.young

"Gamma" is not a natural function. The factorial is the natural function. Don't ask me why "Gamma" has become so widespread as compared to factorial. Historically, the factorial function was used by Gauss and Reimann, but Gamma was used by Legedre. It's not clear what advantages Legedre thought shifting the argument to get Gamma had (except that the first singularity is at zero rather than -1), but the result is that virtually every time you'd use it, you'll have to shift the argument again to get back to the factorial.

Now, when we're talking about calculators, some require factorial to be integers, in which case the factorial function could be defined as

$$(n)! = 1 * 2 * 3 * \dots * (n-1) * n$$

or

$$(1)! = 1$$

$$(n)! = n * (n-1)!$$

Calculators that do this for factorial will not allow the factorial function of a noninteger to be calculated. It seems the 42s uses the integer definition for factorial, but the more general definition of gamma. Other calculators do different things.

For instance, the 49G+ lets you take the factorial of any real number, but not of complex numbers. But for some reason, Gamma works on real and complex numbers. Who knows why.

There are various ways of generalizing the factorial function to nonintegers (but all ways will give the same function! For example, $(2)!$ will always be 2, regardless of the exact generalizing method, $(1/2)!$ will always be $\sqrt{\pi}/2$, etc.)

One illuminating way is

$(s)!$ = infinite product of terms $(1+1/n)^s / (1 + s/n)$, as n goes from 1 to infinity.

Now imagine we multiply $(s)!$ by $(-s)!$. In that case, the numerator of those terms goes away! We're left with a product of terms of the form

$1/(1 - (s/n)^2)$, as n goes from 1 to infinity.

Now, the function $\sin(\pi x)/(\pi x)$ has roots at $x = \pm 1, \pm 2, \pm 3, \pm 4, \dots$. It also has a value of 1 when $x = 0$

So, if (as was Euler's idea) we would factor it like a polynomial, we'd get

$$\sin(\pi x)/(\pi x) = (1-x)(1+x)(1-x/2)(1+x/2)\dots$$

=

$$(1-x^2)(1-(x/2)^2)(1-(x/3)^2)(1-(x/4)^2)\dots$$

Which is a polynomial which equals 1 when $x = 0$, and is zero if $x = \pm 1, \pm 2$, etc.

And this is exactly the same as the reciprocal of what we found for $(x)!(-x)!$ above!

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

 [Go back to the main exhibit hall](#)