♦MoHPC♠   *The Museum of HP Calculators*

HP Forum Archive 16

[Return to Index | Top of Index]

HP-15C Mini-challenge: Speeding it up !

Message #1 Posted by Valentin Albillo on 12 July 2006, 5:49 a.m.

Hi all:

Today it's my **20th Wedding Anniversary** (yes, I'm *that* old), which certainly is a truly memorable day for me and my one and only wife, MJ, and I wish to share the celebration with all of you by issuing yet another ultra-small and cute HP-15C Mini-challenge (tm), which goes as follows:

> Note: You've got no HP-15C ? No problem, just download the free, awesome Eric Smith's Nonpareil - High-Fidelity Calculator Simulator, which simulates/emulates a real HP-15C (among many other HP models) at the ROM level, and you're all set up for this challenge and more, no lame excuses :-).

This well-known series for Pi is as simple as it gets ...

```
Pi = 4 * (1 - 1/3 + 1/5 - 1/7 + 1/9 - ...)
```

... but for the purpose of numerically computing Pi to some decent accuracy it is utterly useless, as it converges soooo *slowly* that you'll need *millions and millions* of terms just to achieve a handful of correct digits. For instance,

```
adding up to       30 terms you'll get:  3.11+
adding up to      300 terms you'll get:  3.138+
adding up to    3,000 terms you'll get:  3.1413+
adding up to   30,000 terms you'll get:  3.14156+
adding up to  300,000 terms you'll get:  3.141589+
```

so you see, even adding as many as 300,000 terms will get you only just about 5 correct decimal places. Doing it on an HP-15C would take *ages* !

Taking this into account, here's your

HP-15C mini-challenge:

"Using the above series for Pi as a basis, try and write an HP-15C program as short and fast as possible, to efficiently compute Pi to 10 correct digits."

Your program must accept as input the number of terms to use, and you must specify how many terms your program needs to achieve 10-digit accuracy, (give or take one or two units in the last place, to cater for error accumulation), and the time it takes to deliver the goods.

You may use whatever techniques you deem appropriate as long as you *do* use the above series as the basis for your computation and adequately justify the theoretical background for the techniques you're using (without this proviso, you could simply add up just one term of the series, and then subtract the "magical" constant 0.858407346 from it to get a correct 10-digit approximation to Pi and get along with it within the stated rules ! ;-)

Silly cheats aside, you must strive for minimum program size and resources used, while keeping running times *extremely* manageable, no week-long computations here. Within a few days I'll post my original solution, in two versions:

1. The first one is a *16-step* HP-15C program which uses this series as a basis to compute and output an approximation to Pi correct to 10 places (but for 2 units in the last place) in a few minutes.

2. The second one is slightly longer, at *22 steps*, but it will compute an approximation to Pi correct to <u>all 10 digits</u> (and thus identical with the value returned by the HP-15C's built-in Pi function), and does it in *under a minute*.

I'll also give the interesting theory behind it and assorted comments.

Best regards from V. *and* MJ.

Re: HP-15C Mini-challenge: Speeding it up !

Message #2 Posted by Giancarlo (Italy) on 12 July 2006, 6:15 a.m.,
in response to message #1 by Valentin Albillo

Valentin,
unfortunately I've got currently any possibility to get your challenge
but please accept my best wishes for your wedding anniversary!!
Warmest regards.
Giancarlo

Thank you *very* much, Giancarlo ! :-) [N.T.]

Message #3 Posted by Valentin Albillo on 12 July 2006, 6:23 a.m.,
in response to message #2 by Giancarlo (Italy)

Take care, and

Best regards from V.

Re: HP-15C Mini-challenge: Speeding it up !

Message #4 Posted by Namir on 12 July 2006, 6:39 a.m.,
in response to message #1 by Valentin Albillo

Divide 355 by 113 and you get a very accurate result even with a simple calculator!

Re: HP-15C Mini-challenge: Speeding it up !

Message #5 Posted by Valentin Albillo on 12 July 2006, 6:52 a.m.,
in response to message #4 by Namir

Hi, Namir:

Namir posted:

"Divide 355 by 113 and you get a very accurate result even with a simple calculator!"

> True, but I fail to see how that meets the Mini-challenge conditions, which are to produce an HP-15C program that uses that particular series as a basis to produce a 10-digit result.

Best regards from V.

Re: HP-15C Mini-challenge: Speeding it up !

Message #6 Posted by Namir on 12 July 2006, 7:27 a.m.,
in response to message #5 by Valentin Albillo

Speed!!!!!!!!!!!!!

Re: HP-15C Mini-challenge: Speeding it up !

Message #7 Posted by Valentin Albillo on 12 July 2006, 7:47 a.m.,
in response to message #6 by Namir

If you're running a car race from Los Angeles to Washington and you happen to travel there by airplane instead it will do you no good in order to win, no matter how high your speed. :-)

Best regards from V.

Re: HP-15C Mini-challenge: Speeding it up !

Message #8 Posted by Namir on 12 July 2006, 8:06 a.m.,
in response to message #7 by Valentin Albillo

true ... but in computing there is a threshold when a very very slow and inefficient algorithm can't win again a simple and elegant much faster solution.

Re: HP-15C Mini-challenge: Speeding it up !

Message #9 Posted by Valentin Albillo on 13 July 2006, 4:39 a.m.,
in response to message #8 by Namir

Hi, Namir:

Namir posted: (the bold is mine)

*"... in computing there is a threshold when a **very very slow** and inefficient algorithm can't win again a simple and elegant much faster solution."*

> That's exactly why I call this a "mini-**challenge**" and why it's titled "**Speeding** it up!"

> "Speeding it up!". Not: "Throw-it-out-because-it's-so-slow-and-try-something-different-instead".

> Else you wouldn't have much of a **challenge**, would you ? :-)

Best regards from V.

Re: HP-15C Mini-challenge: Speeding it up !

Message #10 Posted by Paul Dale on 12 July 2006, 6:54 a.m.,
in response to message #4 by Namir

3 * Ln(640320) / sqrt(163) is much better than 355/113. Pity neither are acceptable as a solution.

Pauli

Re: HP-15C Mini-challenge: Speeding it up !

Message #11 Posted by Namir on 12 July 2006, 7:21 a.m.,
in response to message #10 by Paul Dale

Thanks Paul!! Always nice to know about cool aprroximation to pi.

I checked your approximation and you are right. It has an error of 0.63 per billion vs 84 per billion using 355/113.

Error per billion = |1E9 * (pi - pi_approx)/ pi|

Namir

Edited: 12 July 2006, 7:26 a.m.

Re: HP-15C Mini-challenge: Speeding it up !

Message #12 Posted by Valentin Albillo on 12 July 2006, 7:41 a.m.,
in response to message #11 by Namir

Hi again, Namir:

Namir posted:

"Thanks Paul!! Always nice to know about cool aprroximation to pi."

This is an extremely well-known approximation to Pi, having to do with so-called "Ramanujan's constant". I've posted it myself several times in the past few months in a number of threads."

"I checked your approximation and you are right. It has an error of 0.63 per billion vs 84 per billion using 355/113. Error per billion = |1E9 * (pi - pi_approx)/ pi|"

He's right but you're wrong. This approximation to Pi is much better than your numerical computation shows. According to your own formula, we have:

```
> 3*log(640320)/sqr(163)

    3.14159265358979301649588865
```

```
      > Pi

          3.14159265358979323846264338


      > 1E9 * (Pi-3*log(640320)/sqr(163))


          0.0000002219667547300827791
```

Thus the error is a mere 0.00000022 per billion, many orders of magnitude smaller than your *"error of 0.63 per billion"* estimate. For someone so knowledgeable about numerical algorithms that's *gross* inaccuracy, you risk ruining your reputation ... :-)

Best regards from V.

Re: HP-15C Mini-challenge: Speeding it up !

Message #13 Posted by Namir on 12 July 2006, 8:08 a.m.,
in response to message #12 by Valentin Albillo

I was using an HP-41CX to calculate the errors. I guess that WILL ruin my reputation among HP calculator fans, won't it?

You also forgot to divide by pi. Now BOTH of our reputations are ruined!!

:-)

Namir

PS: I REALY DO ENJOY YOUR CHALLENGES and I HAVE A LOT OF ADMIRATION FOR YOUR CONTRIBUTIONS TO THIS FORUM, but I am a bit turned off by this one since it uses a horribly inefficient algorithm, especially when there are SO MANY other better approximations.

Edited: 12 July 2006, 8:23 a.m.

Re: HP-15C Mini-challenge: Speeding it up !

Message #14 Posted by Valentin Albillo on 12 July 2006, 9:08 a.m.,
in response to message #13 by Namir

Hi, Namir:

Namir posted:

"I was using an HP-41CX to calculate the errors. I guess that WILL ruin my reputation among HP calculator fans, won't it?"

Of course. Anything short of a 48/49 and you're considered a dinosaur, sort of :-)

"You also forgot to divide by pi. Now BOTH of our reputations are ruined!!"

He, he :-) You're right ... Oh, the shame !

"PS: I REALY DO ENJOY YOUR CHALLENGES and I HAVE A LOT OF ADMIRATION FOR YOUR CONTRIBUTIONS TO THIS FORUM, but I am a bit turned off by this one since it uses a horribly inefficient algorithm, especially when there are SO MANY other better approximations."

Give me some credit, Namir. If someone else doesn't post it before I do (in a few days), I guarantee you'll be delighted and amazed with this one as well. Things aren't as simple or hopeless as they seem, and *wonderful* mathematics may hide where least expected, trust me :-)

Besides, you wouldn't expect me to issue a clunker to commemorate my 20th Wedding Annyversary, right ? My wife would kill me if I did ! ;-)

Thanks a lot for your continued interest and kind words, and

Best regards from V.

Hey Namir! Email me?
Message #15 Posted by Gene Wright on 12 July 2006, 10:44 a.m.,
in response to message #13 by Namir

Have some questions for you...

Re: HP-15C Mini-challenge: Speeding it up !
Message #16 Posted by Paul Dale on 12 July 2006, 8:11 a.m.,
in response to message #12 by Valentin Albillo

Try here for a decent list of other Pi approximations.

Pauli

Re: HP-15C Mini-challenge: Speeding it up !
Message #17 Posted by Namir on 12 July 2006, 8:16 a.m.,
in response to message #16 by Paul Dale

Paul,

My browser(s) are not connecting to the site you are pointing out to.

Namir

Re: HP-15C Mini-challenge: Speeding it up !
Message #18 Posted by Paul Dale on 12 July 2006, 8:21 a.m.,
in response to message #17 by Namir

The full URL is:

http://mathworld.wolfram.com/PiApproximations.html

Regards,

Pauli

Re: HP-15C Mini-challenge: Speeding it up !
Message #19 Posted by Namir on 12 July 2006, 8:36 a.m.,
in response to message #18 by Paul Dale

Pauli,

Thank you .... looks like that web site is temporarely down. I can't even get the root web page at http://mathworld.wolfram.com/

I will try again later. Many thanks Pauli!!!

Namir

Re: HP-15C Mini-challenge: Speeding it up !

Message #20 Posted by Les Wright on 18 July 2006, 11:04 p.m.,
in response to message #12 by Valentin Albillo

On my HP41CX (and I have no idea if it is a pine nut, macadamia nut, brazil nut, etc ;) I get about 0.32 per billion. I can't seem to replicate Namir's 0.63. For the sake of comparison, my keystrokes are 640320 [LN] 3 [*] 163 [SQRT] [/] [SHIFT][Pi] [-] [SHIFT][Pi] [/] 1E-9 [*], with 0.318309886 on the display.

In Maple keeping oodles of digits I get about 7.07e-17 relative error. I also get the same thing with Free42. I have sort have gotten used to seeing errors compared in the numerical analysis literature and books as -100 times the common logarthm. In this case the error of the approximation is 1615, about double precision or better.

Les

Re: HP-15C Mini-challenge: Speeding it up !

Message #21 Posted by Paul Dale on 12 July 2006, 6:39 a.m.,
in response to message #1 by Valentin Albillo

Congratulations on your 20 years. We celebrate 17 next month so I guess I'm not so young either.

This is a cheater's solution. I'll see if I can do a real solution too if I can get Nonpareil going.

The taylor series for atan is:

    atan(x) = x - (x^3)/3 + (x^5)/5 - ...

Substituting x=1 into the above equation gives us the series we're required to evaluate.

Hence the program:

    ATAN
    4
    *

does the required job although it does ignore the number of terms it is supposed to evaluate. You might want to throw a switch to radians in there too.

Pauli

Re: HP-15C Mini-challenge: Speeding it up !

Message #22 Posted by Valentin Albillo on 12 July 2006, 7:06 a.m.,
in response to message #21 by Paul Dale

Hi, Paul:

Paul wrote:

"Congratulations on your 20 years. We celebrate 17 next month so I guess I'm not so young either."

Thank you very much. And congratulations in advance for your 17th Wedding Anniversary. Marriages lasting that far are increasingly rare nowadays ...

"This is a cheater's solution [...] The taylor series for atan is: atan(x) = x - (x^3)/3 + (x^5)/5 - ...

Correct.

"Substituting x=1 into the above equation gives us the series we're required to evaluate."

Correct.

"Hence the program: ATAN 4 * does the required job" "

Wrong !, because HP-15C's ATAN built-in function does *not* use the atan(x) Taylor's series to compute the arc tangent at all but some specialized (probably CORDIC-based) algorithm.

Thus, it is *not* evaluating the series object of this mini-challenge, and so it does not meet this mini-challenge's requirements, which mandatorily require using that particular series as a basis, not some other unrelated algorithm which approximately produces the same result (such as Namir's 355/113 ! :-)

Next ! :-)

Best regards from V.

Re: HP-15C Mini-challenge: Speeding it up !

Message #23 Posted by Paul Dale on 12 July 2006, 7:09 a.m.,
in response to message #22 by Valentin Albillo

Quote:

Wrong !, because HP-15C's ATAN built-in function does *not* use the atan(x) Taylor's series to compute the arc tangent at all but some specialized (probably CORDIC-based) algorithm.

Which is why I mentioned that bit about cheating :-) Still worth a try...

Pauli

Re: HP-15C Mini-challenge: Speeding it up !

Message #24 Posted by Paul Dale on 12 July 2006, 8:01 a.m.,
in response to message #1 by Valentin Albillo

I've made a little progress, not on a 15C though and still a long way to go yet.

I took your HP-11C program that sums an alternating series and modified it for my 42S so that more function evaluations could be stored and it gives an answer correct to ten digits using 21 terms in a little under 25 seconds. It uses lots of program steps.

I might try some more tomorrow,

Pauli

Re: HP-15C Mini-challenge: Speeding it up !

Message #25 Posted by Bram on 12 July 2006, 3:00 p.m.,
in response to message #1 by Valentin Albillo

Hi Valentin,

As usual your challenge immediately kept me busy for a while and on my way home I got an idea for an approach. Not sure if this is what you meant, but if not there's all the more reason for discussion.

```
if I rewrite
Pi = 4 * (1 - 1/3 + 1/5 - 1/7 + 1/9 - ...)
to
Pi = 4 * (1 - (1/3 - 1/5) - (1/7 - 1/9) - ...)
I get half the number of terms that can be computed to
Pi = 4 * (1 - 2/15 - 2/63 - 2/143 - ...)
being
Pi = 4 * (1 - 2 * (1/15 + 1/63 + 1/143 + ...)
```

or
Pi = 4 - 8 * (1/(4^2-1) + 1/(8^2-1) + 1/(12^2-1) + ...)
and both the slowness of the alterations is removed and the terms go quadratically towards zero

As I don't have a 15C and I don't like emulators (yet?), I took my 32SII to compute the final equation:

```
LBL Wedding
4
STO X
0
STO Sum
STO Count
LBL A
1
STO+ C
RCL X
x²
1
-
1/x
RCL S
x<>y
RCL+ S
STO S
x=y?
STOP
4
STO+ X
GTO A
```

and the result you have to multiply by 8 and subtract from 4. Still it takes ages to complete, also because of the 11 digit precision of the 32SII, but that's not the point I guess. Compared to the straight forward equation it runs much much quicker.

regards, and congrats, Bram

Re: HP-15C Mini-challenge: Speeding it up !

Message #26 Posted by Crawl on 12 July 2006, 4:18 p.m.,
in response to message #25 by Bram

The next step should be to use Euler Maclaurin summation.

The balance is between the number of terms we want to take in the original series, and the number of correction terms we wish to calculate. Also, the more correction terms we use, the longer (but faster) our program will be. For this case, I just went ahead and used

```
S = sum(1/(16*x^2-1) , x, 1, a) +


1/8 * ln((4a+1)/(4a-1))
-1/2 *(1/(16*a^2-1))
+1/12*(32*a)/(16*a^2-1)^2
-1/720*(6144a(16a^2+1)/(16a^2-1)^4
+1/30240*(983040a*(768a^4+160*a^2+3)/(16a^2-1)^6
```

With this many correction terms, to get an error of e, we need to have a be approximately

a = (1/(672e))^(1/7)

Note that only ELEVEN terms in the original sum will give an error less than 1e-10! So, this is a huge increase in speed.

Even with only 2 terms in the original sum, you still get something like 3.1415...

I'll have to programing this in later, if possible

Re: HP-15C Mini-challenge: Speeding it up !

Message #27 Posted by Kiyoshi Akima on 12 July 2006, 9:11 p.m.,
in response to message #26 by Crawl

After my first program using the general arctan series, I too went to the Euler-Maclaurin summation. Unlike Crawl, I stopped with two error terms, resulting in the following program:

```
001 LBL B      016 .         031 6
002 STO I      017 5         032 *
003 4          018 RCL I     033 1
004 *          019 x^2       034 -
005 ENTER      020 1         035 1/x
006 ENTER      021 6         036 +
007 1          022 *         037 DSE I
008 +          023 1         038 GTO 3
009 x<>y       024 -         039 8
010 1          025 /         040 *
011 -          026 -         041 4
012 /          027 LBL 3     042 -
013 LN         028 RCL I     043 CHS
014 8          029 x^2       044 RTN
015 /          030 1
```

With only two error terms, one needs to sum nearly 500 terms of the series, requiring about eleven minutes to generate ten correct decimal digits.

Adding more error terms will reduce the number of terms needed in the sum, but as Crawl points out, will increase the program length. And no amount of tweaking and optimizing will bring this program down to the 22 steps of V's longer program.

With Crawl's formulation (5 error terms and 11 terms in the series) the program should easily come home in under a minute, but probably would approach 100 steps in length.

Obviously, V has something else in his devious mind :-)

Re: HP-15C Mini-challenge: Speeding it up !

Message #28 Posted by Valentin Albillo on 13 July 2006, 4:29 a.m.,
in response to message #27 by Kiyoshi Akima

Hi, Kiyoshi Akima:

You wrote:

"Obviously, V has something else in his devious mind :-)"

> That's why I call it "a mini-challenge" instead of "a maxi-programming exercise" !

> As for the *"devious mind"* bit, we're now resorting to name-calling !? :-)

> Well, 36 more hours to go before I post my solutions, there's still plenty of time and I'm pretty sure a number of people will duplicate or improve on my original solutions, will you be one of them ? :-)

> Thanks for your continued interest and your excellent postings, and

Best regards from V.

Re: HP-15C Mini-challenge: Speeding it up !

Message #29 Posted by Kiyoshi Akima on 12 July 2006, 3:53 p.m.,
in response to message #1 by Valentin Albillo

It wasn't clear whether the challenge specifies the use of the series for arctan(1) or the general form for arctan(x). If we can assume the latter, then the series is

```
arctan(x) ~ x - x^3/3 + x^5/5 - x^7/7 + x^9/9 - ...
```

For tan(B)=1/5, we have

```
          2 tan(B)    5
tan(2B) = -------- = --
          1-tan²(B)  12
```

and

```
          2 tan(2B)   120
tan(4B) = --------- = ---
          1-tan²(2B)  129
```

This differs only by 1/119 from 1, whose arctangent is pi/4; in terms of angles, this difference is

```
tan(4B-pi/4) = [2tan(4B)-1]/[1+tan(4B) = -1/239]
```

and hence

```
arctan(1/239) = 4B - pi/4 = 4arctan(1/5)-pi/4
```

The derivation is due to John Machin, who computed pi to 100 decimal digits 280 years before V's wedding. (Congrats, V! I'll hit 17 this year.)

Both arctan series converge faster than arctan(1). How fast? If we consider the arctan series as a power series in x with nonexistent even terms, then the magnitude of the ith term is less than x^i, and since it's an alternating series the magnitude of the error is less than the magnitude of the first neglected term.

Log(1/5) ~ -0.699, -10/log(1/5) ~ 14.31, so fifteen terms suffice to give us ten decimal digits.

Log(1/239) ~ -2.378, -10/log(1/239) ~ 4.20, so five terms suffice to give us ten decimal digits.

My RPN is very rusty, my first effort ended up at 56 steps and runs in about 16 seconds (on the emulator), with an error of 1 ulp.

The first part of the program simply ignores the number of terms and computes the two arctangents and the final result:

```
001 LBL A      008 STO 5      015 *
002 2          009 5          016 RCL 0
003 3          010 ENTER      017 -
004 9          011 1          018 4
005 ENTER      012 6          019 *
006 8          013 GSB 0      020 RTN
007 GSB 0      014 4
```

The rest of the program computes the arctangent based on the series:

```
021 LBL 0      033 RCL 2      045 /
022 STO 5      034 *          046 -
023 Rdown      035 STO 4      047 ST- 3
024 1/x        036 RCL I      048 RCL 5
025 STO 3      037 /          049 RCL I
026 STO 4      038 RCL 4      050 4
027 x^2        039 RCL 2      051 +
028 STO 2      040 *          052 STO I
029 3          041 STO 4      053 x<=y
030 STO I      042 RCL I      054 GTO 1
031 LBL 1      043 2          055 RCL 3
032 RCL 4      044 +          056 RTN
```

A brute-force RPL program did 12 digits exactly on the HP 48 in under half a second.

For more on the history of computing pi (and a BCPL program), take a look here.

Re: HP-15C Mini-challenge: Speeding it up !

Message #30 Posted by Kiyoshi Akima on 13 July 2006, 1:10 p.m.,
in response to message #29 by Kiyoshi Akima

Oops.

If you tried this program and didn't get the expected results, that's because there was a typo. Step 008 should be "STO 0" and not "STO 5" as shown.

The same program works with almost no change on the 16C. Just change "047 ST-3" to "RCL 3" "x<>y" "-" "STO 3" and "027 x^2" to "ENTER" "*". It takes about 20 seconds on that machine.

Re: HP-15C Mini-challenge: Speeding it up !

Message #31 Posted by Walter B on 12 July 2006, 6:54 p.m.,
in response to message #1 by Valentin Albillo

Valentin,

congratulations on your 20 years! As you wrote, such long marriages are becoming rare nowadays.

"Besides, you wouldn't expect me to issue a clunker to commemorate my 20th Wedding Annyversary, right ? My wife would kill me if I did ! ;-)"

Appears you even have a mathematics aficionada? That's really rare! So all the best to you and your wife!!

Although we cannot compete mathematically (it's past midnight here, and I have to work tomorrow, so I do have a good excuse ;-)), we can compete on the original field: we had our 26th anniversary a month ago. :)

Best regards, Walter

Re: HP-15C Mini-challenge: Speeding it up !

Message #32 Posted by Valentin Albillo on 13 July 2006, 4:18 a.m.,
in response to message #31 by Walter B

Hi, Walter:

Walter posted:

"Congratulations on your 20 years! As you wrote, such long marriages are becoming rare nowadays."

> Thank you very much and yes, they are. I think this is mostly due to the fact that newer generations are a lot more hedonistic than we were, and they won't suffer anything that contradicts their wishes and expectations. So, as soon as some non-trivial disagreements begin to happen, they simply cut the relationship for good and look elsewhere, no serious attempt to bend their own views a little to try and adapt to their partner's.

"Appears you even have a mathematics aficionada? That's really rare! So all the best to you and your wife!!"

> So why did you think I married her, in the first place ? ;-) Thank you very much for your kind wishes, much appreciated.

"... we can compete on the original field: we had our 26th anniversary a month ago. :)"

> Why, that's impressive ! 26 years !! :-) My very best wishes to you and your wife as well, it's nice to see that indeed there are a number of long-wed couples among this forum's frequent visitors.

Thanks again and best regards from V.

Re: HP-15C Mini-challenge: Speeding it up !

Message #33 Posted by Antoine M. Couëtte on 17 July 2006, 6:02 a.m.,
in response to message #32 by Valentin Albillo

Congratulations Valentin on your Wedding anniversary.

I am always extremely impressed at your mathematical challenges in which I have no time to study because so busy : twice Milano-Madras in five and half days (42 flight hours) and Milano-Chicago the day after to-morrow. So I do need some rest too, even if I LOVE flying long haul aircraft.

I always read with a lot of interest your superb contributions. And I am - again - extremely curious to read about your "Pie Recipe"

Best Regards from

Antoine

PS

Everything you wrote about Marriage - even if difficult to hear or read to a number of people - is certainly true in many respects. Still every situation is a UNIQUE case and we can only try to help out and show compassion to those who go through the terrible painful and tragic event of Marriage rupture.

... and by the way, my only Wife Sabine and I will hit our 29 th Anniversary this fall, after a extremely severe 7 year storm which is currently resolving into a new life of Love together. Why did our Marriage did not fall apart ? Partly because we both decided that we would do everything in our power to avoid such a tragedy to both ourselves and our 7 Children and partly also thanks to the presence of true friends standing by and praying for us.

Re: HP-15C Mini-challenge: Speeding it up !

Message #34 Posted by Bram on 13 July 2006, 4:53 a.m.,
in response to message #31 by Walter B

Quote:

(...) we can compete on the original field: we had our 26th anniversary a month ago. :)

Looks like I'm in good company.
24 (and counting :-)

I second Giancarlo´s wishes.

Message #35 Posted by Vieira, L. C. (Brazil) on 12 July 2006, 7:52 p.m.,
in response to message #1 by Valentin Albillo

Be well, Valentin!

Luiz (Brazil)

Re: I second Giancarlo´s wishes.

Message #36 Posted by Valentin Albillo on 13 July 2006, 4:04 a.m.,
in response to message #35 by Vieira, L. C. (Brazil)

Hi, Luis:

Thanks a lot ! :-)

By the way, I've noticed your frequency of posting has diminished appreciably as of lately (unless it's just me ...), so I hope it's simply due to a tremendous amount of other more urgent things to do and all's well on your side.

Anyway, the more you post, the nicer this forum will feel :-)

Best regards from V.

Re: I second Giancarlo´s wishes.

Message #37 Posted by Vieira, L. C. (Brazil) on 14 July 2006, 4:12 p.m.,
in response to message #36 by Valentin Albillo

Hi, Valentin;

thanks for your kind words, I appreciate.

You are correct, I´ve been too busy these last two months due to some extra activity at the University. As you said, everythgn else is going right.

One extra acivity of mine is setting my own lab up. I think I´ll have some good news, soon...

In time: I was about to post a message thanking you, Valentin, for your continuous and valuable contributions, mainly the S&S challenges, but other contributor(s) took the head of it. I read all of the cahlenges, try to reason briefly over their contents, and as I see that I cannot contribute the way I´d like to, I wait till the answers and save a copy of them for future reading. Also, your 'Live Long...' series are a 'must read' pack of good stuff.

Thanks, Valentin. Your posts are inspiring.

Best regards from Brazil.

Luiz

Re: HP-15C Mini-challenge: Speeding it up !

Message #38 Posted by Paul Dale on 13 July 2006, 7:04 a.m.,
in response to message #1 by Valentin Albillo

Using the series expansion for arctan:

```
atan(x) = x - (x^3)/3 + (x^5)/5 - ...
```

which might be close enough to the requested series to be permitted.

Then relying on the identity:

```
atan(a) + atan(b) = PI/4 when b = (1-a) / (1+a)
```

Then take advantage of the special case a = b = sqrt(2) - 1 gives us:

```
PI = 8 * atan(sqrt(2) - 1)
```

which is rather nice because we now only have to calculate a single series expansion for arctan and the expansion is at a more rapidly converging point than atan(1).

Here is my attempt at a HP-15C program:

```
01  LBL A
02  8
03  STO 2
04  2
05  SQRT
06  1
07  STO 1
08  -
09  STO 0
10  *
11  LBL 0
12  ENTER
13  ENTER
14  RCL 0
15  2
16  RCL+ 1
17  STO 1
18  Y^X
19  RCL/ 1
20  RCL 2
```

```
21  CHS
22  STO 2
23  *
24  +
25  x != y
26  GTO 0
```

I've never used a 15c before so I've likely missed a few optimisations. I'm a bit concerned about the two ENTERs needed on lines 12 and 13, I really only want to have one there but that doesn't work. I can also leave off the initial label to save a single step. Still not quite good enough.

Run time is about 25 second, it evaluates 12 terms before it reaches its answer (actually it does evaluate the 13th but the result doesn't change). Accuracy is suspect being out by 3 in the last digit.

Factoring the * 8 out of the loop doesn't change the program's size or runtime and hurts accuracy by another 1 in the last digit.

Finally, as an aside, the initial series:

```
1 - 1/3 + 1/5 - 1/7 + 1/9 - ...
```

looks rather like being conditionally convergent and as such the summation is anything I want it to be (according to the Riemann series theorem). I haven't had this much fun with analysis since my undergradute days!

- Pauli

Re: HP-15C Mini-challenge: Speeding it up !
Message #39 Posted by Paul Dale on 13 July 2006, 4:59 p.m.,
in response to message #38 by Paul Dale

I thought about this some more overnight and I think accuracy might be improved by summing the series backwards. Unfortunately, this makes the program longer.

Re: HP-15C Mini-challenge: Speeding it up ! (and a special message)
Message #40 Posted by Andrés C. Rodríguez (Argentina) on 13 July 2006, 1:14 p.m.,
in response to message #1 by Valentin Albillo

Valentin:

It was difficult for me to find an earlier time to post this short comment in response to your message dated yesterday. You mentioned your anniversary (please receive my congratulations for MJ and you), and I noticed it was also my anniversary… well, not a wedding anniversary but an important date nevertheless; because in the same day of July, many years ago, I met my one and only wife, CV, for the first time. And we celebrated the anniversary of our first encounter every year from then on.

I was 11 years old... and just now I noticed that the next year will mark the 20 anniversary of that event. We married 8 years later, in October, after both of us graduated. She is a History professor, which brings some equilibrium to a rather scientific/ technical/ mathematical/ nerdy family. In fact, her view is usually complementary of mine, and so is a much needed and appreciated counterweight; so to speak.

My story may be sweet, and it is short indeed; and while it falls short of one of those "short and sweet" challenges of yours, it nevertheless has some mathematical riddle in. A trick I took advantage from when I realize that my current age is no greater, but also no less, than 30; and when such an undeniable fact makes me behave as pretending I am somewhat younger…

It is in such moments when I reach for my loved HP scientific programmable calculator (a high-end Pioneer model, one which was discontinued too early), and press the second key of the BASE menu…

PS: No, the initials of my wife's younger sister are not CX.

Re: HP-15C Mini-challenge: Speeding it up ! (and a special message)

Message #41 Posted by Valentin Albillo on 14 July 2006, 4:35 a.m.,
in response to message #40 by Andrés C. Rodríguez (Argentina)

Hi, Andrés:

Andrés posted:

"You mentioned your anniversary (please receive my congratulations for MJ and you) ..."

Why, thank you very much, indeed, much appreciated ! :-)

"... and I noticed it was also my anniversary... well, not a wedding anniversary but an important date nevertheless; because in the same day of July, many years ago, I met my one and only wife, CV, for the first time. And we celebrated the anniversary of our first encounter every year from then on."

How nice, and what a coincidence ! :-) Please accept my congratulations to both of you, as well. Talking about "coincidences", that "CV" seems a case of predestination :-)

"In fact, her view is usually complementary of mine, and so is a much needed and appreciated counterweight; so to speak."

That's also my case; we're both complementary and it's my personal opinion that couples where each partner complements the other tend to be more balanced and stable than the opposite, when both partners have very similar traits, because in that case none of them ever tries to restrain the other when he/she's going the wrong way, as they would do the same, and at times a kind of 'sibling rivalry' does develop when both think they're pretty good at what they do, and won't easily accept that the other might do better.

"PS: No, the initials of my wife's younger sister are not CX."

LOL ! :-) Even "CV" seems somewhat puzzling: does "C" stand for "Cristina" or "Carla" ? Does "V" stand for "Valeria" or "Verónica" ? Anyway, nice initials for an HP-calc aficionado's wife !

Thanks for sharing your fond memories with us, really nice and really much appreciated.

Best regards from V.

Re: HP-15C Mini-challenge: Speeding it up ! (and a special message)

Message #42 Posted by Andrés C. Rodríguez (Argentina) on 14 July 2006, 7:46 a.m.,
in response to message #41 by Valentin Albillo

a) Claudia Viviana. She uses no calc at all (except when doing grade averages for her students).

b) ... But my first daughter initials are MJ (Maria Julieta), and she is pursuing an industrial engineering degree at the same university I attended. She inherited the 4-level stack RPN bug from me, have a 32 S ii all the time in her student backpack, but also sometimes uses a 48G+ I have at home (of course, I don't like nor use the 48, except for playing Minehunt).

c) And my younger daughter Ana Ines attends the secondary school. She likes her algebric HP 30S, and I managed not to make her dependent on a close-to-extintion system as RPN is, no matter how much I regret it.

d) I assume you solve the riddle en-passant, didn't you?

Edited: 14 July 2006, 7:58 a.m.

Re: HP-15C Mini-challenge: Speeding it up !

Message #43 Posted by Paul Dale on 13 July 2006, 6:01 p.m.,
in response to message #1 by Valentin Albillo

This "solution" might be pushing the limits of acceptability...

Recognising that the series we're interested in:

    1 - 1/3 + 1/5 - 1/7 + 1/9 - ...

just happens to be the solution to:

    integral(0, 1, dx/(1+x^2))

This brief program results:

    01 LBL A
    02 0
    03 FIX 9
    04 1
    05 intgerate 0
    06 4
    07 *
    08 RTN
    09 LBL 0
    10 x^2
    11 1
    12 +
    13 1/x

This takes about two and a half minutes to execute and produces an answer correct to ten digits.

Pauli

Re: HP-15C Mini-challenge: Speeding it up !

Message #44 Posted by Valentin Albillo on 14 July 2006, 4:18 a.m.,
in response to message #43 by Paul Dale

Hi again, Paul:

Paul posted:

"This "solution" might be pushing the limits of acceptability..."

Indeed ! As I mentioned before, the mini-challenge requires that you *do-actually-use-the-series* as a basis. The fact that the series adds up to Pi, or that it happens to equal some definite integral, or the Taylor expansion of some function at a particular value is irrelevant and does not allow you to use some other algorithms to compute Pi, or the definite integral, or evaluate the Taylor expansion at some other argument.

Else, if the series happened to add up to 5, say, you could argue that 5 = 2+3 and then LBL A, 2, ENTER, 3, +, RTN would do nicely.

This is why considering the series as 4*ATAN(1) was not a legal entry (HP-15C's ATAN's internal algorithm does *not* use the series) and your present entry, integrating $1/(1+x^2)$ (which by the way is yet again atan(x) in disguise) fails as well, as HP-15C's INTEG algorithm *doesn't* use the original series either, but rather evaluates $1/(1+x^2)$ at a number of arguments and does something with the results. Tthe original series is *never* evaluated or used.

Anyway, thanks for your continued interest and your brave attempts to find imaginative solutions. There's still 12 to 24 hours left, go for it ! :-)

Best regards from V.

Re: HP-15C Mini-challenge: Speeding it up !

Message #45 Posted by Paul Dale on 14 July 2006, 4:36 a.m.,
in response to message #44 by Valentin Albillo

I thought it would be rejected :-(

At least I do have one solution that uses the proper series even if it isn't as short or accurate as yours.

Pauli

Re: HP-15C Mini-challenge: Speeding it up !

Message #46 Posted by Bruno Ferard on 16 July 2006, 7:53 a.m.,
in response to message #44 by Valentin Albillo

I did try, I can swear, but could not find a solution that matches the criteria. The countdown has elapsed; I am looking forward to reading your solution. You may want to let us ponder over it a little longer. But don't forget to post it, I am delighted in advance...

Re: HP-15C Mini-challenge: Speeding it up !

Message #47 Posted by Kiyoshi Akima on 13 July 2006, 7:24 p.m.,
in response to message #1 by Valentin Albillo

Could someone familiar with the 15C list the tests available with the "g TEST" sequence for the benefit of those of us without manuals? Or point us to a free manual (not the one on the CD)?

thx

Re: HP-15C Mini-challenge: Speeding it up !

*Message #48 Posted by **Paul Dale** on 13 July 2006, 7:34 p.m.,*
in response to message #47 by Kiyoshi Akima

Try the picture of the back panel http://www.hpmuseum.org/15cbk.jpg.

Re: HP-15C Mini-challenge: Speeding it up !

*Message #49 Posted by **Kiyoshi Akima** on 13 July 2006, 7:52 p.m.,*
in response to message #48 by Paul Dale

Yeah, that works too. Should've thought of looking at the back when looking at the front didn't show the ones I wanted...

thx for the quick response.

Re: HP-15C Mini-challenge: Speeding it up !

*Message #50 Posted by **Kiyoshi Akima** on 13 July 2006, 7:49 p.m.,*
in response to message #47 by Kiyoshi Akima

To answer my own request: I found a quick-reference card here

Re: HP-15C Mini-challenge: Speeding it up !

*Message #51 Posted by **Valentin Albillo** on 14 July 2006, 4:02 a.m.,*
in response to message #47 by Kiyoshi Akima

Hi, Kiyoshi Akima:

You posted:

"Or point us to a free manual (not the one on the CD)?"

I don't own the "CD" so I don't know if this HP-15 manual in PDF format is the same or not, but you may download it and compare both, if desired.

Best regards from V.

Re: HP-15C Mini-challenge: Speeding it up !

Message #52 Posted by Crawl on 14 July 2006, 10:20 a.m.,
in response to message #1 by Valentin Albillo

> Quote:
>
> Your program must accept as input the number of terms to use, and you must specify how many terms your program needs to achieve 10-digit accuracy, (give or take one or two units in the last place, to cater for error accumulation), and the time it takes to deliver the goods.

I'm not sure what you mean by "Your program must accept as input the number of terms to use".

However, this should be a program that accepts a *parameter* (the number 1, 2 or 3 on the stack) and from that *calculates* the number of terms to use.

Entering 1 gives 4 digits of accuracy in under 18 seconds.

Entering 2 gives 7 digits of accuracy in 2 minutes, 36 seconds.

Entering 3 gives 9 digits of accuracy in 26 minutes, 14 seconds. (3.141592655, meeting the condition of "10-digit accuracy, give or take one or two units in the last place..")

The mathematical principle this is based on is that the error for this formula is known, based on the number of terms taken.

For 10^x terms taken, it turns out that the first error in pi is very simple: Just 10^{-x}!

This program could be made to give more accurate results in the same time by taking into account more error terms. Just two error terms would probably suffice.

Here's the program:

```
LBL 1    001 -    42,21,  1
STO 1    002 -    44      1
10^x     003 -            13
1        004 -            1
-        005 -            30
3        006 -            3
CHS      007 -            16
```

```
10^x    008 -            13
x       009 -            20
STO 2   010 -    44       2
0       011 -             0
LBL 2   012 -    42,21,   2
1       013 -             1
CHS     014 -            16
RCL 2   015 -    45       2
INT     016 -    43      44
y^x     017 -            14
RCL 2   018 -    45       2
INT     019 -    43      44
2       020 -             2
x       021 -            20
1       022 -             1
+       023 -            40
/       024 -            10
+       025 -            40
ISG 2   026 -    42,6,    2
GTO 2   027 -    22       2
4       028 -             4
x       029 -            20
RCL 1   030 -    45       1
CHS     031 -            16
10^x    032 -            13
+       033 -            40
RTN     034 -    43      32
```

Edited: 14 July 2006, 10:21 a.m.

Re: HP-15C Mini-challenge: Speeding it up !

*Message #53 Posted by **Kiyoshi Akima** on 14 July 2006, 5:46 p.m.,*
in response to message #1 by Valentin Albillo

My latest attempt uses the Euler transform and divided differences to speed up the convergence. I knew from the start it was too complicated a scheme to make a compact program, but I wanted to try the algorithm just to see how well it works. The RPL program all but fell together and the 48GX delivers all twelve digits in under two seconds.

On the 15C I can get +/- 1 ulp in about 58 seconds, or all ten digits in just under two minutes. However the program is nearly a hundred steps long, thus I won't bore you by listing it here (unless somebody really wants to see it). I know it's not close to optimal, but I also know I'm not going to shrink it down to a quarter of its size.

Just what does V have in his devious (meaning circuitous, roundabout, as opposed to straightforward, obvious) mind?

BTW, thx for pointing me at the manual.

Re: HP-15C Mini-challenge: Speeding it up !

Message #54 Posted by Dave Shaffer on 16 July 2006, 7:30 p.m.,
in response to message #1 by Valentin Albillo

If you group adjacent pairs of terms, you can get the summation to converge as $1/n^2$, which is better than $1/n$, but still not very fast ($1/9999^2$ and $1/10000^2$ aren't that much different and are helping you only at the eighth decimal place). I tried grouping even more terms (four at a time), but the algebra was getting pretty bad, and the convergence didn't seem to be getting any faster.

Are there better ways to group terms?

PS my wife (a confirmed 11C and 32S user and giver of my 41CX) and I will celebrate 27 years in August.

Re: HP-15C Mini-challenge: My Original Solutions & Comments

Message #55 Posted by Valentin Albillo on 16 July 2006, 7:33 p.m.,
in response to message #1 by Valentin Albillo

Hi all,

Thanks again for your interesting and highly imaginative efforts to solve this mini-challenge, even if at times they were somewhat derivative (or 'integrative', come to think of it ...), I hope you enjoyed it all and here are my original solutions with a thorough explanation of how can one empirically get to the correct idea and the theory behind it.

As stated in the challenge's description, the task is to find a way to use the well-known series:

```
Pi = 4 * (1 - 1/3 + 1/5 - 1/7 + ... )
```

to compute Pi to 10 correct places while keeping program size and running time small. A direct approach seems doomed to failure as this series converges so incredibly slowly that millions of terms must be added up to get no more than 6 or 7 correct digits, let alone 10. To clearly demonstrate it, this simple 15-step HP-15C program, which will serve as the basis for my solutions, will add up any specified *even* number of terms from the series:

```
01*LBL A      06  0        11 STO 0
02 STO I      07*LBL 0     12 RCL/I
03 STO+I      08 DSE I     13  +
04  4         09 RCL 0     14 DSE I
05 STO O      10 CHS       15 GTO 0
```

To improve accuracy, the program begins adding up the smallest terms and goes *backwards* until it reaches the largest term, 1. Upon running it, you'll see that, as expected, the convergence is *awfully slow*. Let's try to add 4 terms, then 44, then 444:

```
4  , GSB A ->  2.895238096  (barely one correct digit)
44 , GSB A ->  3.118868314  (barely three correct digits)
444, GSB A ->  3.139340404  (barely four correct digits)
```

This last result took almost *7 minutes*, yet we've got no more than *four* not-so-correct digits, so the situation seems hopeless. At this point, it seems we can do no better than try some relatively complicated techniques, such as the Euler-McLaurin formula or extrapolation mechanisms for summation of infinite, alternating series such as this one. This would incur in a serious penalty in vastly increased *complexity* and program *size*, as seen in several working programs posted by contributors.

A bit of sleuthing:

However, math is full of surprises and serendipitous findings are bound to happen when and where you least expect them, as we'll immediately see. Let's use our basic program to add up exactly 50 terms:

```
50, GSB A ->  3.121594653
```

Now, this has a fairly large error, as we're getting 3.12+ instead of 3.14+, so that the 3rd digit is alreay 2 units off. But, don't you notice something truly eerie ?

Yes, we get a "2" where a "4" should be. But the following three digits (159) are correct ! Then we get another wrong digit, a "4" which should be a "2", but then the next three digits (653) are once again correct !!

Let's align our value and the correct Pi value and carefully examine the differences:

```
        Sum  ->  3.121594653
        PI   ->  3.141592653 (58979...)
                 ---------------------
                   +2  -2
```

which, in absolute values means:

```
        +0.02 -0.000002
```

Let's see if this is just a weird coincidence, or else it also happens for other numbers of terms being added up. Let's try 100 terms, for instance:

```
100, GSB A ->  3.131592904
               3.141592654
               -----------------
                   +1    -25
                 +0.01  -0.00000025
```

and we see that our initial impression does hold, because after one wrong digit, the subsequent four digits (1592) are indeed correct, then another a couple of wrong digits, and once again another correct digit follows.

Let's call these two 'corrections' C1 and C2 (i.e: +0.02 and -0.000002 for 50 terms, +0.01 and -0.00000025 for 100 terms, respectively) and see how they relate to the number of terms being used. A little insight or a little data-fitting will allow us to issue the following plausible, tentative hypothesis, where N is the number of terms:

```
C1 = 1/N
C2 = -0.25/N³ = -1/4N³
```

which do indeed work for N = 50 and N = 100 terms. Now we'll put our hypothesis to the test, by using it to predict the values of C1 and C2 for N=200 terms:

```
Prediction for N = 200 -> C1 = 1/200        = 0.005
                          C2 = -1/(4*200³) = -0.000000031
```

and we'll now check if they agree with actual results, by running our basic program with 200 as the input value:

```
200, GSB A -> 3.136592685
              3.141592654
              -----------------
                   +5    -31
```

which indeed do exactly agree with our predicted corrections, +0.005 and -0.000000031.

At this point, we can be fairly sure that our empirical finding holds, and can then proceed to make use of it by simply computing one or both correction terms, C1 and C2, and using them to refine the sum provided by our basic program, as follows:

First version, using just one correction term, C1 = 1/N:

Just two little changes to our basic program will compute and add the correction term C1, resulting in a program *just a single step longer*, at 16 steps, yet much faster and accurate:

```
01*LBL A
02 STO I
03 STO+I          50, GSB A ->  3.141594653  in 55"
04 1/X
05  4                   error = 2E-6, actually all digits are correct except the "4" !
06 STO O
07 X<>Y          100, GSB A ->  3.141592904  in 1'50"
08*LBL 0
09 DSE I                error = 2.5E-7
```

```
       10 RCL 0
       11 CHS            400, GSB A ->  3.141592658  in 7'45"
       12 STO 0
       13 RCL/I               error = 4E-9
       14  +
       15 DSE I
       16 GTO 0
```

so this simple version, with just the one correction term C1 does achieve a 10-digit correct value (within 4 ulps) while using just 400 terms, in less than 8 minutes. That's many orders of magnitude better than the basic program could achieve, but we can do still *much* better:

Second version, using two correction terms, C1=1/N and C2=-1/4N^3:

A few stack manipulations will allow us to compute and use both correction terms, C1 and C2 while using *just 5 additional steps*, for a very small total of just *21 steps*:

```
       01*LBL A
       02 STO I
       03 STO+I    40, GSB A ->  3.141592651  in 40"  (error = 3E-9)
       04 1/X
       05 ENTER
       06 ENTER    50, GSB A ->  3.141592653  in 50"  (error = 1E-9)
       07  3
       08 Y^X
       09  4       62, GSB A ->  3.141592654  in 60"  (error = 0)
       10 STO O
       11  /
       12  -
       13*LBL 0
       14 DSE I
       15 RCL 0
       16 CHS
       17 STO 0
       18 RCL/I
       19  +
       20 DSE I
       21 GTO 0
```

so this improved version needs to add up just *62 terms* to return a full *10 correct-digit value* within *60 seconds*. Here's a table summarizing the different degrees of approximation using 0, 1, and 2 correction terms, for up to 60 terms added up:

N	bare series	+C1	+C1 +C2	t
10	3.041839619	3.141839619	3.141589619	10"
20	3.091623807	3.141623807	3.141592557	20"

30	3.108268567	3.141601900	3.141592641	30"
40	3.116596557	3.141596557	3.141592651	40"
50	3.121594653	3.141594653	3.141592653	50"
60	3.124927144	3.141593811	3.141592653	60"

Further empirical confirmation:

As we've been able to indeed get 10 correct digits by using our empirically discovered corrections, we can be fairly confident that they are no mere coincidences but hold for greater number of terms added up and thus greater precision. To test this, just out of curiosity, these are the results for N = 500, 5000, 50000, 500000, and 5 million terms added up:

```
N = 500 terms added up


    3.13959265558978323858464...
    3.14159265358979323846264...
       +2    -2   +10  -122


N = 5,000 terms added up


    3.14139265359179323836264339547950...
    3.14159265358979323846264338327950...
       +2      -2     +10     -122


N = 50,000 terms added up


    3.14157265358979523846264238327950410419716...
    3.14159265358979323846264338327950288419716...
          +2         -2        +10         -122


N = 500,000 terms added up


    3.14159065358979324046264338326950288419729139937510...
    3.14159265358979323846264338327950288419716939937510...
           +2             -2           +10           -122


N = 5,000,000 terms added up


    3.14159245358979323846464338327950278419716939938730582097494...
```

```
        3.141592653589793238462643383279502884197169399375100582097494...
           +2          -2              +10            -122
```

Notice in particular the values for N = 5,000,000 terms: the 7th decimal is already in error by 2 units. But the next 13 digits are all correct ! Then, the following digit is also 2 units wrong. But the next 12 digits are again correct !! All in all, *among the first 47 digits, only 3 of them are a few units wrong* !

In other words, the original series converges incredibly slowly, granted, but the errors when you stop at N terms are *extremely predictable* and *easy to compute*, so you can increase your accuracy 3-fold or 5-fold by using just one or two easily derived correction terms.

Final notes

This empirical discovery, once made, can be substantiated by theory, and a nifty expression is arrived at which results in an asymptotic approximation to Pi based on the sum of the original series truncated to N terms plus a 'correction' series (the asymptotic component) in negative powers of N ($1/N$, $1/N^3$, etc) where the so-called Euler numbers are the coefficients.

Similar phenomena occur for constants other than Pi, for example similarly truncating the series:

```
    Ln(2) = 1 - 1/2 + 1/3 - 1/4 + 1/5 - ...
```

results in:

```
    Sum   = 0.6931470805599553094172321212581765680755161343436025525...
    Ln(2) = 0.6931471805599453094172321214581765680755001343436025525...
                 1      -1            2            -16
```

and another asymptotic series can be theoretically substantiated, the required coefficients being now the so called *"tangent numbers*" instead: 1, -1, 2, -16, ...

Thanks for your interest and many excellent posted contributions, hope you enjoyed yourselves while working them out.

Best regards from V.

Re: HP-15C Mini-challenge: My Original Solutions & Comments
Message #56 Posted by Bram on 17 July 2006, 3:54 a.m.,
in response to message #55 by Valentin Albillo

And you called this a **mini**-challenge ??!
Wouldn't have discovered this in a megayear...
Beautiful.
Thanks.

Re: HP-15C Mini-challenge: My Original Solutions & Comments

Message #57 Posted by Namir on 17 July 2006, 11:48 a.m.,
in response to message #55 by Valentin Albillo

Valentin,

I read your solution with great fasciation. Now that's what I call a very clever solution. Hats off to you!!!

At the end I asked myself, "Why don't we all work on imroving popular algorithms in numerical analysis?" Many of these algorithms were created by mathematicans who would kill for PCs, let alone programmable calculators OR even simple calculators!!! We are so fortunate to have all the power tools!!!

I know I kno .. easier said than done! I don't think it's impossible though!

Namir

Re: HP-15C Mini-challenge: My Original Solutions & Comments

Message #58 Posted by Kiyoshi Akima on 17 July 2006, 12:58 p.m.,
in response to message #55 by Valentin Albillo

Wow. Nice.

Re: HP-15C Mini-challenge: My Original Solutions & Comments

Message #59 Posted by Bruno Ferard on 17 July 2006, 5:03 p.m.,
in response to message #55 by Valentin Albillo

This solution is accessible to everybody, even with limited mathematical background. That's really impressive. Thanks for sharing this with us.

Re: HP-15C Mini-challenge: My Original Solutions & Comments

Message #60 Posted by Les Wright on 18 July 2006, 10:35 p.m.,
in response to message #55 by Valentin Albillo

Absolutely gorgeous!

An HP42S adaptation with 50,000 iterations takes at most 30 or 40 seconds in Free42 running on a Palm TX to give full 25 digit precision, the best it gets with Free42.

Les

Re: HP-15C Mini-challenge: My Original Solutions & Comments

Message #61 Posted by Valentin Albillo on 19 July 2006, 3:50 a.m.,
in response to message #60 by Les Wright

Hi, Les:

Would you post the HP42S version, please, for the sake of completeness when this thread is archived ?

BTW, I think you'll agree that getting 25 correct digits from that ultra-slow convergent series in non-geological times is utterly unexpected, isn't it ? :-)

Thanks and best regards from V.

Edited: 19 July 2006, 3:54 a.m.

Re: HP-15C Mini-challenge: My Original Solutions & Comments

Message #62 Posted by Paul Dale on 19 July 2006, 4:26 a.m.,
in response to message #61 by Valentin Albillo

Here is a HP 12c version...

```
01  STO 1
02  STO+ 1
03  1/x
04  ENTER
05  ENTER
06  3
07  y^x
08  4
09  STO 0
10  /
11  -
12  1
13  STO- 1
```

```
14  Rv          Roll down
15  RCL 0
16  CHS
17  STO 0
18  RCL 1
19  /
20  +
21  1
22  STO-1
23  Rv
24  RCL 1
25  x=0?
26  GTO 29
27  Rv
28  GTO 12
29  Rv
30  GTO 00
```

Timings are:

```
N   t
-------
10  11"
20  20"
30  29"
40  39"
50  48"
```

Pauli

Re: HP-15C Mini-challenge: My Original Solutions & Comments

Message #63 Posted by Les Wright on 20 July 2006, 3:44 a.m.,
in response to message #61 by Valentin Albillo

> Quote:
>
> Would you post the HP42S version, please, for the sake of completeness when this thread is archived ?

Sure. It is identical to yours, with the the obvious syntactical differences--HP42S can store to alpha-named variables, for example. This can work just as well on an HP41 by using numbered storage registers of choice instead of the variables "I" and "O".

```
01 LBL "PIE"
02 STO "I"
03 STO+ "I"
04 1/X
05 ENTER
06 ENTER
07  3
08 Y^X
09  4
10 STO "O"
11  /
12  -
13 LBL 00
14 DSE "I"
15 RCL "O"
16 -/+
17 STO "O"
18 RCL/ "I"
19  +
20 DSE "I"
21 GTO 00
```

I know not everyone in this forum is fond of emulators and simulators, but if there was ever a good case for the speed and high precision internal arithmetic of Tom Okken's Free42, this challenge is a case in point. These days, for all of my RPN programming, I seem to be using nothing else, and I own a 12C, 41CV, 41CX and 42S. Indeed, once I collect my thoughts, I plan to post a little message of praise for Free42, with a little narrative of the numerical analysis meanderings that lead me to this conclusion.

Les

P.S. When you did your high-iteration tests (i.e. out to 5 million terms, etc.) of your discovery, what did you use? The actual 15C would take hours or days, wouldn't it?

Re: HP-15C Mini-challenge: My Original Solutions & Comments

Message #64 Posted by Valentin Albillo on 20 July 2006, 4:26 a.m.,
in response to message #63 by Les Wright

Hi, Les:

Les posted:

"When you did your high-iteration tests (i.e. out to 5 million terms, etc.) of your discovery, what did you use? The actual 15C would take hours or days, wouldn't it? "

If it could do the multiprecision computations with the number of digits shown (which it can't) at the same speed as it does when using just 10-digit values (which it wouldn't), it would take an HP-15C more than 2 months of continuous running to sum up five million terms. Even if it could, I seriously doubt the batteries would last that long, and there's no AC adapter to save the day anyway :-)

Actually, I did the high-iteration tests with Mathematica, which takes very little time.

Best regards from V.

Re: HP-15C Mini-challenge: My Original Solutions & Comments
Message #65 Posted by Les Wright on 20 July 2006, 4:58 a.m.,
in response to message #64 by Valentin Albillo

Quote:

If it could do the multiprecision computations with the number of digits shown (which it can't) at the same speed as it does when using just 10-digit values (which it wouldn't)

Boy, I feel silly! I should've thought of this before asking such a silly question!

It would be interesting to plot -log(precision) vs. the number of terms to discern the relationship.

Here is a little wrinkle I discovered in Maple--whether one ADDS or SUBTRACTS the correction depends on the parity of N. Try to run the program with odd vs. even values of N and see what happens! Me thinks a factor of $(-1)^N$ or $(-1)^{(N-1)}$ is needed somewhere....

Les

Re: HP-15C Mini-challenge: My Original Solutions & Comments
Message #66 Posted by Valentin Albillo on 20 July 2006, 5:27 a.m.,
in response to message #65 by Les Wright

Hi again, Les:

Les posted:

"Here is a little wrinkle I discovered in Maple--whether one ADDS or SUBTRACTS the correction depends on the parity of N. Try to run the program with odd vs. even values of N and see what happens!"

That's not allowed. In my posted solution, I said:

```
"To clearly demonstrate it, this simple 15-step HP-15C
 program, which will serve as the basis for my solutions,
 will add up any specified even number of terms from the
 series:"
```

so your conversion, if exactly based in my program, is also valid only for **even** values of N and so no sign correction factor is needed anywhere.

Best regards from V.

Re: HP-15C Mini-challenge: My Original Solutions & Comments

Message #67 Posted by [Les Wright](#) on 20 July 2006, 5:41 a.m.,
in response to message #66 by Valentin Albillo

You already thought of it!

And I thought I was being clever ;)

That said, I still would like to plot -log(error) vs. N (including a parity corrector), to see what the curve looks like.

Les

Re: HP-15C Mini-challenge: My Original Solutions & Comments

Message #68 Posted by [RDunn](#) on 19 July 2006, 2:25 p.m.,
in response to message #55 by Valentin Albillo

I've just finished digesting the solution and have a problem which I hope somebody can help me with.

In response to Paul Dale, Valentin Albillo said:

Quote:

Indeed ! As I mentioned before, the mini-challenge requires that you do-actually-use-the-series as a basis. The fact that the series adds up to Pi, or that it happens to equal some definite integral, or the Taylor expansion of some function at a particular value is irrelevant and does not allow you to use some other algorithms to compute Pi, or the definite integral, or evaluate the Taylor expansion at some other argument.

Then, in the solution, Valentin Albillo said:

> Quote:
>
> However, math is full of surprises and serendipitous findings are bound to happen when and where you least expect them, as we'll immediately see. Let's use our basic program to add up exactly 50 terms:
>
> ```
>     50, GSB A ->  3.121594653
> ```
>
> Now, this has a fairly large error, as we're getting 3.12+ instead of 3.14+, so that the 3rd digit is alreay 2 units off. But, don't you notice something truly eerie ?
>
> Yes, we get a "2" where a "4" should be. But the following three digits (159) are correct ! Then we get another wrong digit, a "4" which should be a "2", but then the next three digits (653) are once again correct !!

My problem is this: Without knowing that the series adds up to Pi, how do we know that the second decimal digit should be a "4", or that the following three digits are correct?

Granted, there is always an asymptotic series in the negative powers of N, but how do you derive the coefficients without either knowing the sum or carrying out enough terms to get sufficient accuracy?

Re: HP-15C Mini-challenge: My Original Solutions & Comments
Message #69 Posted by Paul Dale on 19 July 2006, 8:11 p.m.,
in response to message #68 by RDunn

One could even go as far as saying that the inclusion of the correction terms isn't summing the series. Taken to an extreme, just eval one or two terms in the series and then use the correction series which converges far faster...

Still an interesting piece of mathematics I'd not seen before. I did have a quick look at estimating the error terms but the functional form didn't click. Not too surprising if you don't stop at a N where the error has a recognisable or finite decimal expansion :-(

- Pauli

Re: HP-15C Mini-challenge: My Original Solutions & Comments

Message #70 Posted by Valentin Albillo on 20 July 2006, 4:14 a.m.,
in response to message #69 by Paul Dale

Hi, Paul:

Paul posted:

"One could even go as far as saying that the inclusion of the correction terms isn't summing the series"

My wording for the challenge is absolutely precise in that respect:

```
"Using the above series for Pi as a basis [...]
 You may use whatever techniques you deem appropriate as
 long as you *do* use the above series as the
 basis for your computation and adequately justify
 the theoretical background for the techniques you're using"
```

and, as can be seen, my basic program and both of my solutions *do add up N terms of the series*, where N is specified by the user: 400 to 500 terms suffice for near 10-digit accuracy in the simpler version, 62 terms suffice for full 10-digit accuracy in the second version.

Once you've summed up a number of terms from the series, as required, whether afterwards you do use Euler-McLaurin, or correction terms, or whatever other refinement techniques, is up to you, that's not the part that does sum up the original series. By the time you consider refinements, the summation must have previously been completed up to N terms.

"Taken to an extreme, just eval one or two terms in the series and then use the correction series which converges far faster..."

You got it completely wrong, the correction series does *not* converge far faster, matter of fact *it does not converge at all*, but rather it actually **diverges** for all N, that's why I said that it is an **asymptotic** series.

You've probably encountered asymptotic series before. One such is Stirling's series to approximate large factorials or gamma function values. Being asymptotic, you get fast but limited convergence to the correct value, up to a *finite* maximum accuracy that depends on the value of N. You *can't* improve that maximum accuracy for a given finite N no matter how many terms of the asymptotic series you do use. After a point, adding any more terms results in *decreased* accuracy and ultimately *divergence*. However, as N goes larger and larger, the maximum (but finite for a finite N) obtainable accuracy does increase without limit.

Same here. After adding a certain number of terms of the asymptotic correction series, which depends on N, you get a maximum accuracy. Adding any more terms after that, your accuracy decreases and the 'correction' gets worse and worse, ultimately diverging. You can't increase

the obtainable accuracy by using more terms from the correction series but only by increasing N itself.

Best regards from V.

Re: HP-15C Mini-challenge: My Original Solutions & Comments

*Message #71 Posted by **RDunn** on 20 July 2006, 1:41 p.m.,*
in response to message #70 by Valentin Albillo

I'm new to this forum, so forgive me if I'm misunderstanding the rules here.

> Quote:
>
> Once you've summed up a number of terms from the series, as required, whether afterwards you do use Euler-McLaurin, or correction terms, or whatever other refinement techniques, is up to you, that's not the part that does sum up the original series. By the time you consider refinements, the summation must have previously been completed up to N terms.

Let S(N) be the partial sum of the first N terms and let T(N) be the error in S(N) such that

```
S(Infinity)=S(N)+T(N)
```

In his solution, Valentin asserts

```
T(50)=3.141592563-S(50)
```

By induction one can easily show that

```
T(N)=3.141592653-S(N) (+/- 1E-9) for N>=0
```

Thus

```
S(Infinity)=S(N)+3.141592653-S(N)
```

A program to add up N terms, add the magic number 3.141592563, and subtract N terms, should take about twice as long as Valentin's program for the same N. By computing S(N) once and storing it for later subtraction, we should match his speed. From his timing table, the run time is about N seconds.

There's a further optimization possible to reduce the run time to a constant, but that probably violates the "the summation must have previously been completed up to N terms" clause.

Re: HP-15C Mini-challenge: My Original Solutions & Comments

Message #72 Posted by Paul Dale on 20 July 2006, 7:49 p.m.,
in response to message #71 by RDunn

But you're achieving full accuracy after but a single term of S(N). I'd be willing to wear the overhead of computing the '1' term twice :-)

- Pauli

Re: HP-15C Mini-challenge: My Original Solutions & Comments

Message #73 Posted by Paul Dale on 21 July 2006, 12:33 a.m.,
in response to message #70 by Valentin Albillo

> Quote:
>
> [ul]You got it completely wrong, the correction series does *not* converge far faster, matter of fact *it does not converge at all*, but rather it actually **diverges** for all N, that's why I said that it is an **asymptotic** series.

I'll have to take your word on this my real analysis is a bit rusy these days.

The first four terms in the error series from the last example you gave are:

```
 0.0000002
-0.00000000000000000002
 0.000000000000000000000000000000001
-0.0000000000000000000000000000000000000000000000122
```

I think I can be forgiven for assuming that this series converges :-)

Bring on the next puzzle :-)

Regards,

- Pauli

(Re: HP-15C Mini-challenge ...): Asymptotic series made 'visible'

Message #74 Posted by Valentin Albillo on 22 July 2006, 12:38 p.m.,
in response to message #73 by Paul Dale

Hi, Paul:

"The first four terms in the error series from the last example you gave are: [...] I think I can be forgiven for assuming that this series converges :-)"

Certainly, it's understandable that when you see the apparently very fast convergence you'd think it continues and even increases when more and more terms are used.

That's not the case because of the *asymptotic* nature of this correction series. The term's numerators are proportional to the Euler numbers wich (roughly) increase as n^n, say, while the denominators increase as 10^n only, so the numerators' magnitude will eventually *exceed* that of the denominators and the terms will be even greater than 1 from then on.

This small program for the HP-71B, which just takes a few seconds to run, will make it visually clear. It uses a very good *approximation* to compute the numerators (exact values of the Euler numbers are not so easy to compute) while the denominators are computed exactly, for assorted number of terms.

As the values eventually grow incredibly large and far exceed the normal range ($>= 10^{500}$), the program switches to base-10 logarithms from a certain point on, and finally computes and outputs the approximate number of terms necessary for the numerators to equal or exceed the denominators:

```
10 DISP "# terms              App. Numerator     Denominator     Sign(N-D)"
20 DISP "-----------------------------------------------------------------"
30 FOR I=1 TO 13 @ IF I<6 THEN N=I ELSE N=10*N
40 IF I<6 THEN A=IROUND(10^FNN(N)) @ B=10^FND(N) ELSE A=FNN(N) @ B=FND(N)
50 IF I=6 THEN DISP @ DISP "(decimal logarithms from now on)" @ DISP
60 DISP N,A,B,SGN(A-B) @ NEXT I
70 DISP @ DISP "Approximate # terms when numerator >= denominator: ";
80 DISP IROUND(FNROOT(10000,100000000,FNN(FVAR)-FND(FVAR)))
90 !
100 DEF FND(N)=(2*N-1)*7
110 DEF FNN(N) @ N=N-1 @ IF N=0 THEN FNN=LGT(2) @ END
120 FNN=LGT(16)+LGT(N/PI)/2+2*N*LGT(4*N/PI/EXP(1))


>RUN


# terms              App. Numerator     Denominator     Sign(N-D)
-----------------------------------------------------------------
 1                   2                  10000000        -1
 2                   2                  1.E21           -1
 3                   10                 1.E35           -1
```

```
4                    120                 1.E49              -1
5                    2741                1.E63              -1


(decimal logarithms from now on)


50                   135.160191328       693                -1
500                  2366.28334531       6993               -1
5000                 33691.0537239       69993              -1
50000                436952.261381       699993             -1
500000               5369577.83834       6999993            -1
5000000              63695847.1079       69999993           -1
50000000             736958553.304       699999993           1
500000000            8369585628.77       6999999993          1


Approximate # terms when numerator >= denominator:   21349339
```

So you see, even though the numerators start very modestly (2, 2, 10, 120, ...) while the denominators start *very* srong (10^7, 10^{21}, 10^{35}, 10^{49}, ...), after some **21,350,000 terms** they manage to catch on, and from then on they're *exponentially greater*, thus all subsequent terms are greater than 1 and exponentially increasing. By then, both numerators and denominators will be integer numbers around 10 raised to the power of:

```
>FND(21349339)


   298890739
```

which means they are nearly *300 million-digit integers* !

Finally, it's also very easy to determine the approximate number of terms necessary for the terms to *stop diminishing*, which is where you should stop adding them up to achieve maximum (but finite) accuracy. That's left as an (easy) exercise to the reader :-)

Best regards from V.

Edited: 22 July 2006, 2:36 p.m.

Re: HP-15C Mini-challenge: My Original Solutions & Comments

Message #75 Posted by Les Wright on 20 July 2006, 4:30 a.m.,
in response to message #55 by Valentin Albillo

> Quote:
>
> ---
>
> But, don't you notice something truly eerie ?
>
> ---

I have been playing around with this in Maple to a few dozen digits for a few hours now (iterations over a million terms takes a while....), and I have to tell you the recurrence of the pattern is downright spooky..... ;)

I know there have been some skeptical responses in the thread, but I have to tell you this find impresses me. I wonder if it is documented in the formal literature anywhere?

Les

Re: HP-15C Mini-challenge: My Original Solutions & Comments

Message #76 Posted by Paul Guertin on 20 July 2006, 2:25 p.m.,
in response to message #75 by Les Wright

Les Wright wrote:

> Quote:
>
> ---
>
> I know there have been some skeptical responses in the thread, but I have to tell you this find impresses me. I wonder if it is documented in the formal literature anywhere?
>
> ---

Yes. I read about it for the first time in a wonderful book called *Mathematics by Experiment* by Jonathan Borwein and David Bailey (co-discoverers, with Simon Plouffe, of a famous formula for computing individual binary digits of pi). It is subsection 2.2 (p.48 in my hardcover edition), entitled "A Curious Anomaly in the Gregory Series". Credit for the discovery is attributed to Joseph Roy North of Colorado Springs in 1988. A web search of "Joseph Roy North" turns up more information.

Wonderful: there is an online PDF containing long excerpts from the book, including "A Curious Anomaly". Download it here, you won't be disappointed.

Paul Guertin

Re: HP-15C Mini-challenge: My Original Solutions & Comments

Message #77 Posted by Les Wright on 20 July 2006, 4:21 p.m.,
in response to message #76 by Paul Guertin

> Quote:
>
> ---
>
> Mathematics by Experiment by Jonathan Borwein....
>
> ---

I am waving the maple leaf with approval!

The Borwein name (Peter and Jonathan) is big in Canadian mathematics and computer science. Borwein is also an important academic family in London, Ontario, home of the University of Western Ontario, of which I am an alumnus.

Valentin made no mention of 1988 North discovery. Was he already aware of it, or did he come across the pattern himself from first principles?

Les

Re: HP-15C Mini-challenge: My Original Solutions & Comments

Message #78 Posted by Trent Moseley on 20 July 2006, 11:49 p.m.,
in response to message #76 by Paul Guertin

Question for Paul:

In all this fuss about pi to a zillion places is there any interest about the precision of the natural number e ? Both numbers are of such great interest.

tm

Re: Anything about "e"? (from HP-15C Mini-challenge regarding pi)

Message #79 Posted by Karl Schneider on 23 July 2006, 3:01 a.m.,
in response to message #78 by Trent Moseley

Hello, Trent --

Here's a book through Amazon for a low price that probably gives a value to many decimal places, along with the history.

Of course, the denominators of the terms of the series that Valentin addressed for ($\ln 2$) and ($\tan^{-1} 1$) are sequential, while the denominators of terms for the e^x series are factorials. So, the error-correction series doesn't apply. However, error-correction isn't really needed, as e^x converges rapidly and accurately once the magnitude of terms begins to decrease.

-- KS

Edited: 23 July 2006, 3:38 a.m.

Re: V's Solution to HP-15C Mini-challenge + HP-11C/34C program

Message #80 Posted by [Karl Schneider] on 22 July 2006, 4:46 p.m.,
in response to message #55 by Valentin Albillo

Valentin --

Let me offer my belated applause for your Solutions & Comments provided for your "mini-challenge". I finally had an opportunity to print, digest, and run the solution program. The results and presentation are very impressive.

I can offer a very modest supplement for those who don't have an HP-15C, but would like to try it on a fairly-comparable HP-11C or HP-34C. These models don't have all the features used in your solution program, so several workarounds must be used.

Note to readers -- the applicable differences are as follows:

- *No register arithmetic on the I-register.*
- *DSE works <u>only</u> on the I-register, so "DSE" is a complete command.*

Best regards,

-- KS

Valentin's solution program with both correction factors, ported to *and tested on* the HP-11C and HP-34C:

```
01*LBL A
02 ENTER
03 +
04 STO I
05 LASTx
06 1/X
07 ENTER
08 ENTER
09  3
10 Y^X
11  4
12 STO 0
13  /
14  -
15*LBL 0
```

```
16 DSE
17 RCL 0
18 CHS
19 STO 0
20 RCL I
21  /
22  +
23 DSE
24 GTO 0
25 RTN
```

NOTE: The RTN instruction is necessary on all three of these calculators if other programming follows this program in memory.

Edited: 23 July 2006, 2:37 a.m. after one or more responses were posted

Re: V's Solution to HP-15C Mini-challenge + HP-11C/34C program

Message #81 Posted by Valentin Albillo on 22 July 2006, 8:53 p.m.,
in response to message #80 by Karl Schneider

Hi, Karl:

Karl posted:

"Let me offer my belated applause for your Solutions & Comments provided for your "mini-challenge" [...] The results and presentation are very impressive [...] I can offer a very modest supplement for those who don't have an HP-15C, but would like to try it on a fairly-comparable HP-11C or HP-34C"

Thanks a lot for your kind comments and nice version for the HP-11C/34C, much appreciated.

Just in case you hadn't already noticed, check my message #74 above in this very thread where I discuss the asymptotic nature of the correction series and give an additional program to clearly demonstrate why, against all appearances of very fast convergence, it ultimately diverges big time ! :-)

It may come as a nice surprise to people not familiarized with the seemingly paradoxical concept of asymptotic approximations, which usually produce extremely accurate results despite actually being divergent.

Best regards from V.