28th December, 2020, 13:20 **Post: #1**

EdS2
Senior Member

Posts: 525
Joined: Apr 2014

Gerson's Pi Program

Did anyone else have a go at de-obfuscating and running Gerson's short pi-calculating program from the festive greetings thread?

> **Gerson W. Barbosa Wrote:** (26th December, 2020 01:45)
>
> 
>

I'd be interested to know how it works!

I made a plain copy in BBC Basic, which you can run in your browser here.

And here it is below anti-spoiler scroll filler:

Code:

```
.
..
.
...
.
..
.
....
.
..
```


28th December, 2020, 15:15 (This post was last modified: 28th December, 2020 15:15 by Allen.) **Post: #2**

Allen
Member

Posts: 201
Joined: Aug 2014

RE: Gerson's Pi Program

> **Quote:**
> I'd be interested to know how it works!
>
> **Code:**
>
> ```
> 50 m=4*n*n - 1
> 60 w=2
> ...
> 80 w=w+w/m
> ```

Looks like a nicely compact convergence based on Wallis' Product

PM FIND QUOTE REPORT

29th December, 2020, 03:59 **Post: #3**

Gerson W. Barbosa
Senior Member

Posts: 1,500
Joined: Dec 2013

RE: Gerson's Pi Program

> **EdS2 Wrote:** (28th December, 2020 13:20)
> I'd be interested to know how it works!

Hello, Ed!

Thank you very much for your interest! That's an MSX BASIC listing. The MSX computer was very popular in Brazil and England in the mid 80's (but not in the USA). That was my second computer. I liked its 14 significant digits, much better than my first computer, an Apple II clone.

The program is based on this formula:

$$\pi \approx \left(\frac{4}{3}\times\frac{16}{15}\times\frac{36}{35}\times\frac{64}{63}\times\cdots\times\frac{4n^2}{4n^2-1}\right)\left(2+\cfrac{4}{8n+3+\cfrac{3}{8n+4+\cfrac{15}{8n+4+\cfrac{35}{8n+4+\cfrac{63}{8n+4+\cfrac{\ddots}{\cdots\ \cfrac{4n^2-1}{8n+4}}}}}}}\right)$$

That's the third Wallis-Wasicki formula, as I called it, rather jokingly. Notice the reuse of the denominators in the Wallis product as the numerators in the continued fraction.

Here's the thread in which I presented it, in case you missed it:

https://www.hpmuseum.org/forum/post-1394...#pid139434

Best regards,

Gerson.

EMAIL PM FIND QUOTE REPORT

29th December, 2020, 11:15 **Post: #4**

EdS2
Senior Member

Posts: 525
Joined: Apr 2014

RE: Gerson's Pi Program

Thanks! I will surely have seen that post before, but evidently failed to recognise the approach. I think I might blame my ever-increasing age.

EMAIL PM FIND QUOTE REPORT

29th December, 2020, 14:53 (This post was last modified: 29th December, 2020 14:53 by Gerson W. Barbosa.) **Post: #5**

Gerson W. Barbosa
Senior Member

Posts: 1,500
Joined: Dec 2013

RE: Gerson's Pi Program

> **EdS2 Wrote:** (29th December, 2020 11:15)
> Thanks! I will surely have seen that post before, but evidently failed to recognise the approach. I think I might blame my ever-increasing age.

My pleasure! Time goes forward for us all. I have to update my avatar so that it reflects my current age, but I don't remember how I did that twenty years ago.

Actually, this is the post where I first presented that formula, along with an explanation how I found it (not a proof, though).

https://www.hpmuseum.org/forum/post-1345...#pid134502

It appears to produce 25*n/12 correct digits (I have tested the algorithm to 1000 digits only).

The previous program is optimized for speed as it avoids two extra multiplications per iteration. For the few digits provided by most computer languages, it should be better to use this even more compact program:

Code:

```
10 INPUT N
15 C=0
20 D=8*N+4
25 W=2
30 FOR I=N TO 1 STEP -1
35 T=4*I*I-1
40 W=W+W/T
45 C=T/(C+D)
50 NEXT I
55 PRINT W*(2/(C+D-1)+1)
```

EMAIL PM FIND QUOTE REPORT

28th February, 2022, 23:59 (This post was last modified: 10th March, 2022 22:39 by Gerson W. Barbosa.) **Post: #6**

Gerson W. Barbosa
Senior Member

Posts: 1,500
Joined: Dec 2013

RE: Gerson's Pi Program

Here is an arbitrary precision version for the MSX computer. It shouldn't be difficult to convert it to the HP-BASIC (HP-75C and HP-71B), but the base should be lowered to 1E6 or 1E5.

Code:

```
10 REM  ***  PI BY WALLIS-WASICKI FORMULA  ***
20 CLS: KEYOFF: WIDTH 40
30 DEF FNFRAC(X) = X-INT(X)
40 DEF FNRMDR(X,Y)=X-Y*INT(X/Y)
50 DEFINT D,E,G-N
60 DIM A(50), B(50), C(100), Q(50), X(50), Y(100), W(50)
70 LINE INPUT "Number of decimal digits: ";N$: ND=VAL(N$): PRINT
80 B=10000000!: N=ND\7+2: K=ND*12\25+1
90 TIME=0
100 D=8*K+4: E=D-8: G=4*K*K-1
```

Because full-precision multiplications and inversions are required in the main loop, this is not meant to be fast. It took eight minutes and eleven seconds on an emulator running ten times as fast compared to the real machine for just the first 100 digits. The program may return a few extra digits, but only the ones you ask for can be trusted.

```
Number of decimal digits: 100

       3 1415926 5358979 3238462 6433832
 7950288 4197169 3993751  582097 4944592
 3078164  628620 8998628  348253 4211706
 7982103

512.45s
```

Edited to fix a typo in line 720 and to update the new timings.

EMAIL PM FIND QUOTE REPORT

1st March, 2022, 11:39 (This post was last modified: 12th March, 2022 11:13 by EdS2.) **Post: #7**

EdS2
Senior Member

Posts: 525
Joined: Apr 2014

RE: Gerson's Pi Program

Very nice! I'd like to port it to BBC Basic, where we have 4 byte integers and 5 byte floats. But it's not working... any ideas?

Minor notes:

The DEFINT are ignored, because in BBC Basic we need to decorate integer variables with % suffix. I was hoping that would merely be a speed advantage as the 5 byte floats are very nearly as good as the 4 byte ints for accuracy and range.

I took the ! off the end of the B=100000000 but I don't know what it means.

I changed the \ operators to DIV, guessing that this is what's meant.

I changed the printing to print an integer rounded version of the number.

Here's the link.

(Edit: typo for 4 byte ints)

EMAIL PM FIND QUOTE REPORT

1st March, 2022, 13:41 (This post was last modified: 1st March, 2022 18:12 by Gerson W. Barbosa.) **Post: #8**

Gerson W. Barbosa
Senior Member

Posts: 1,500
Joined: Dec 2013

RE: Gerson's Pi Program

> **EdS2 Wrote:** (1st March, 2022 11:39)
> Very nice! I'd like to port it to BBC Basic, where we have 4 byte integers and 5 byte floats. But it's not working... any ideas?
>
> Minor notes:
>
> The DEFINT are ignored, because in BBC Basic we need to decorate integer variables with % suffix. I was hoping that would merely be a speed advantage as the 5 byte floats are very nearly as good as the 5 byte ints for accuracy and range.
>
> I took the ! off the end of the B=100000000 but I don't know what it means.
>
> I changed the \ operators to DIV, guessing that this is what's meant.

B stands for base. Default variable types in MSX are double precision (8 bytes, 14 significant digits), which allows for bases as high as 10000000. With 9-digit precision, you should choose base = 10000 or perhaps 100000. The \ operator is the integer division, as you have correctly guessed. The exclamation sign says it's a single precision number.

Try changing lines 80 and 440:

```
80 B = 10000: N=ND DIV 4 + 2: K=ND*12 DIV 25 + 1
```

```
440 M=10*LOG(ND)/LOG(1000)
```

For 30 digits the output should be

```
3 1415 9265 3589 7932 3846 2643 3832 7941
```

P.S.: For a much faster program, you might want to try Katie Wasserman's program for the HP-71B:

CALCULATING MANY DIGITS OF PI

EMAIL PM FIND QUOTE REPORT

2nd March, 2022, 13:17 **Post: #9**

EdS2
Senior Member

Posts: 525
Joined: Apr 2014

RE: Gerson's Pi Program

Hmm, thanks for the suggested edits, but unfortunately something isn't quite right...
12 digits gives 3 1416 448 2514 8212

(Thanks also the link back to the earlier thread. I'll see about porting that too.)

(Sorry my link to owlet is not working well: something about url-escaping. My posted text works, the rendered link doesn't. If you open a draft which quotes my post you'll get the good link text.)

EMAIL PM FIND QUOTE REPORT

3rd March, 2022, 05:01 **Post: #10**

Gerson W. Barbosa
Senior Member

Posts: 1,500
Joined: Dec 2013

RE: Gerson's Pi Program

> **EdS2 Wrote:** (2nd March, 2022 13:17)
> Hmm, thanks for the suggested edits, but unfortunately something isn't quite right...
> 12 digits gives 3 1416 448 2514 8212

I have a working version for the BeebEm emulator, but I don't know how to get a listing. Anyway, basically you should delete line 40 and use MOD instead:

```
520 T=(B-BR%-C(I)) MOD B
```

```
890 W(I)=INT(X/G): Y=X MOD G
```

```
960 W(I)=X MOD B: Y=INT(X/B)
```

```
1030 C(I)=X MOD B: Y=INT(X/B)
```

Also, lines 30, 740 and 740 should be changed to

```
30 DEF FNFRAC(X) = X+5E-6 - INT(X+5E-6)
```

```
740 AC=INT(B*FNFRAC(CT))
```

```
810 C(I)=INT(B*FNFRAC(T))
```

I've used only real variables, including the one in line 520, but you can change the types of the variables accordingly for speed and memory saving. On the MSX I cannot define AC and C() to integer, because its integer type is only 2-byte long. For the same reason MOD cannot be used on the MSX for bases greater than 10000.

For 12 digits you should get `3 1415 9265 3589 8142`

EMAIL PM FIND QUOTE REPORT

4th March, 2022, 18:50 (This post was last modified: 4th March, 2022 18:51 by EdS2.) **Post: #11**

EdS2
Senior Member

Posts: 525
Joined: Apr 2014

RE: Gerson's Pi Program

Thanks! With your changes, my suboptimal port produces the 12 correct digits as it should, in 53 seconds.

For reference, my simplistic and inefficient port of Katie's program produced 92 correct digits in 71 seconds.

EMAIL PM FIND QUOTE REPORT

6th March, 2022, 03:25 **Post: #12**

Gerson W. Barbosa
Senior Member

Posts: 1,500
Joined: Dec 2013

RE: Gerson's Pi Program

> **EdS2 Wrote:** (4th March, 2022 18:50)
> With your changes, my suboptimal port produces the 12 correct digits as it should, in 53 seconds.
>
> For reference, my simplistic and inefficient port of Katie's program produced 92 correct digits in 71 seconds.

As stated earlier, the WWF is not intended to be fast. According to an article by the Borwein Brothers in Scientific American (February 1988), not even 100 years of computing on a supercomputer programmed to evaluate the Wallis Product or the Gregory's Series would yield 100 digits of π. So, getting 100 digits in 100 minutes on an 8-bit personal computer of that era using the Wallis Product as a basis is an unexpected surprise.

EMAIL PM FIND QUOTE REPORT

8th March, 2022, 19:17 (This post was last modified: 8th March, 2022 19:19 by Gerson W. Barbosa.) **Post: #13**

Gerson W. Barbosa
Senior Member

Posts: 1,500
Joined: Dec 2013

RE: Gerson's Pi Program

> **EdS2 Wrote:** (4th March, 2022 18:50)
> Thanks! With your changes, my suboptimal port produces the 12 correct digits as it should, in 53 seconds.

When porting the program to the HP-75C I noticed an error in line 720.

```
720 FOR J=N-1+2 TO -1 STEP -1
```

The first **1** should be an **I**:

```
720 FOR J=N-I+2 TO -1 STEP -1
```

Please correct it if you haven't done so yet. This should make the program a little less slow.
I am away from home and have brought only the HP-75C along. Also, I can't make your links work. Sorry for the mistake!

EMAIL PM FIND QUOTE REPORT

8th March, 2022, 19:42 **Post: #14**

Valentin Albillo
Senior Member

Posts: 970
Joined: Feb 2015
Warning Level: 0%

RE: Gerson's Pi Program

.
Hi, **Gerson**,

> **Gerson W. Barbosa Wrote:** (6th March, 2022 03:25)
> According to an article by the Borwein Brothers in Scientific American (February 1988), **not even 100 years of computing on a supercomputer programmed to evaluate the Wallis Product or the Gregory's Series would yield 100 digits of π**.

Perhaps, but then again there's this:

Speeding it up !

Regards.
V.

.
.

PM WWW FIND EDIT QUOTE REPORT

9th March, 2022, 03:40 **Post: #15**

Gerson W. Barbosa
Senior Member

Posts: 1,500
Joined: Dec 2013

RE: Gerson's Pi Program

> **Valentin Albillo Wrote:** (8th March, 2022 19:42)
>
> > **Gerson W. Barbosa Wrote:** (6th March, 2022 03:25)
> > According to an article by the Borwein Brothers in Scientific American (February 1988), **not even 100 years of computing on a supercomputer programmed to evaluate the Wallis Product or the Gregory's Series would yield 100 digits of π**.
>
> Perhaps, but then again there's this:
>
> **Speeding it up !**

Hello, Valentin,

Yes, I remember that particular HP-15C Mini-challenge of yours. It was linked by Thomas Klemm in the beginning of a thread I started. As the thread progressed, I ended up with what could have been another solution to your challenge, which I didn't participate in:

https://www.hpmuseum.org/forum/post-9194.html#pid9194

This is a correction term to the Gregory's Series in continued-fraction form, similar to the correction factor to the Wallis Product.

Best regards,

Gerson.

P.S.: This is a closer quotation of the Scientific American Article:

> **Quote:**
> WALLIS PRODUCT (1665)
>
> **Code:**
> ```
> π   2×2   4×4   6×6   8×8
> - = --- x --- x --- x --- x …
> 2   1×3   3×5   5×7   7×9
> ```
>
> GREGORY'S SERIES (1671)
>
> **Code:**
> ```
> π       1   1   1
> - = 1 - - + - - - + …
> 4       3   5   7
> ```
>
> TERMS OF MATHEMATICAL SEQUENCES can be summed or multiplied to yield values for pi (divided by a constant) or its reciprocal. The first two sequences, discovered respectively by the mathematicians John Wallis and James Gregory, are probably among the best-known, but they are practically useless for computational purposes. Not even 100 years of computing on a supercomputer programmed to add or multiply the terms of either sequence would yield 100 digits of pi. The formula discovered by John Machin made the calculation of pi feasible, since calculus allows the inverse tangent (arc tangent) of a number, x, to be expressed in terms of a sequence whose sum converges more rapidly to the value of the arc tangent the smaller x is.

EMAIL PM FIND QUOTE REPORT

9th March, 2022, 10:12 (This post was last modified: 9th March, 2022 10:14 by Ángel Martin.) **Post: #16**

Ángel Martin
Senior Member

Posts: 1,393
Joined: Dec 2013

RE: Gerson's Pi Program

Hi Gerson (and All),

Obviously pi continues to be a top-subject on the forum, with fascinating contributions being added every few months - and that happening for years already. Your latest ones have made me decide to put together a PI/E ROM as an attempt to gather in a common place (i.e. an HP-41 Module) some of the material the forum members have been producing. Since my math prowess is limited and won't allow for more substantial contributions here, this ROM is my tribute to the forum activity on the pi subject.

Obviously the selection is biased by my own preferences, but hopefully I didn't miss the more original or intriguing ones. Besides your latest articles, the sources used obviously include Valentín's wondrous challenges and frequent articles, a treasure trove on the subject. The hyperlinks and forum archives have also been very helpful to compile the material, of course.

To add some value to the project I've made MCODE versions of some algorithms, making them faster and slightly more accurate (due to the 13-digit routines) - though this is a negligible feature in the transcendental pi-world...

Anyway, be as limited as it may, the ROM should be ready in a few more days - ideally on the 14th. but no assurances are made ;-)

Thanks for the contributions, a lot of fun came from working on it.
Best,
ÁM

PM FIND QUOTE REPORT

10th March, 2022, 09:47 **Post: #17**

EdS2
Senior Member

Posts: 525
Joined: Apr 2014

RE: Gerson's Pi Program

> **Gerson W. Barbosa Wrote:** (8th March, 2022 19:17)
> Please correct it if you haven't done so yet. This should make the program a little less slow.

Thanks! There is indeed a speedup. I'm now going through to change to integer variables to see what difference that can make.

> **Quote:**
> Also, I can't make your links work.

Sorry about that - I will publish some updated ones, perhaps on the 14th!

> **Ángel Martin Wrote:** (9th March, 2022 10:12)
> Hi Gerson (and All),
>
> Obviously pi continues to be a top-subject on the forum, with fascinating contributions being added every few months - and that happening for years already. Your latest ones have made me decide to put together a PI/E ROM...

Excellent news! (I hope you find a spigot approach to include: such a thing will produce digits successively from left to right.)

> **Valentin Albillo Wrote:** (8th March, 2022 19:42)
> ... there's this:
>
> **Speeding it up !**

Thanks for the link, Valentin, and again to Gerson for all the ideas and links.

EMAIL PM FIND QUOTE REPORT

10th March, 2022, 13:23 **Post: #18**

Gerson W. Barbosa
Senior Member

Posts: 1,500
Joined: Dec 2013

RE: Gerson's Pi Program

> **EdS2 Wrote:** (10th March, 2022 09:47)
>
> > **Gerson W. Barbosa Wrote:** (8th March, 2022 19:17)
> > Please correct it if you haven't done so yet. This should make the program a little less slow.
>
> Thanks! There is indeed a speedup. I'm now going through to change to integer variables to see what difference that can make.

Thank you! I've updated the listing (will update the timings later).
Times on the HP-75C:
100 digits: 8083.154 s (~ 135 min)
152 digits: 24386.212 s (~ 406 min)

EMAIL PM FIND QUOTE REPORT

10th March, 2022, 13:48 **Post: #19**

Valentin Albillo
Senior Member

Posts: 970
Joined: Feb 2015
Warning Level: 0%

RE: Gerson's Pi Program

.
Hi, **EdS2**,

> **EdS2 Wrote:**
> Excellent news! (I hope you find a **spigot** approach to include: such a thing will produce digits successively from left to right.)
> [...]
> **Thanks** for the link, **Valentin**, [...]

You're welcome. I know you like spigot algorithms for π, for instance you ported my multiprecision *BASIC* spigot program for a *SHARP* pocket computer to the BBC micro, which ran flawlesly (and faster when using integer variables throughout), while I also ported it to the HP-71B, which could produce many thousands of digits (one at a time, mind you) if you had enough RAM and time.

Curiously enough, using **INTEGER** variables on the *71B* results in a *slower* program than using **REAL** variables.

As for what **Ángel** might include in his forthcoming **PI/E ROM**, who knows ... 🙂

Best regards.
V.

PM WWW FIND EDIT QUOTE REPORT

11th March, 2022, 14:42 (This post was last modified: 11th March, 2022 15:38 by EdS2.) **Post: #20**

EdS2
Senior Member

Posts: 525
Joined: Apr 2014

RE: Gerson's Pi Program

Indeed I do like a pi program, and a spigot program, and anything multiprecision.

About porting to BBC Basic - in the case of your spigot, a very naive initial port would produce 50 digits in 118 seconds. With a change to integer variables, it becomes 66 seconds, and then by changing / to DIV, it becomes 54 seconds. We can even save another second by setting D% to 10 and using that instead of the literal in a couple of places.

So by recoding in the fastest idiom for BBC Basic, we get a 2x speedup. (The above timings are for Basic 2 on a 2MHz 6502 - later Basics were faster, and of course CPU clock speeds also improved.)

Other changes which can be beneficial:
- using only single-letter variables, especially A% to Z% which are special-cased
- removing the loop variable from NEXT
- using FOR, REPEAT, PROC and FN instead of line numbers
- inlining functions and subroutines
- removing blank spaces and consolidating multiple lines

Of course, some of these re-codings make the program a bit less clear, and one gets diminishing returns. It's a pity that all the % signs for integer variables are a bit ugly especially if you're not used to them: BBC Basic has no DEF INT.

In the case of Gerson's program, the version below produces 12 digits in 14 seconds, compared to 43 seconds for the initial port. This version produces 36 digits in 186s. (Edit: and 100 digits in 2510s.)

I was quite pleased to do without both FN definitions and especially the 5E-6 which I didn't understand or feel comfortable with. I could use DIV and MOD in a natural way.

Code:

```
10 REM  ***  PI BY WALLIS-WASICKI FORMULA  ***
20 REM CLS: REM KEYOFF: WIDTH 40
60 DIM A%(50), B%(50), C%(100), Q%(50), X%(50), Y%(100), W%(50)
70 INPUT "Number of decimal digits: ";N$: ND=VAL(N$): PRINT
80 B% = 10000: N%=ND DIV 4 + 2: K%=ND*12 DIV 25 + 1
85 M%=10*LOG(ND)/LOG(1000)
90 TIME=0
100 D%=8*K%+4: E%=D%-8: G%=4*K%*K%-1
110 W%(1)=4: B%(1)=0
120 FOR I%=2 TO N%
```

EMAIL PM FIND QUOTE REPORT

11th March, 2022, 18:29 (This post was last modified: 19th December, 2022 17:35 by Gerson W. Barbosa.) **Post: #21**

Gerson W. Barbosa
Senior Member

Posts: 1,500
Joined: Dec 2013

RE: Gerson's Pi Program

> **EdS2 Wrote:** (11th March, 2022 14:42)
>
> Other changes which can be beneficial:
> - using only single-letter variables, especially A% to Z% which are special-cased
> - removing the loop variable from NEXT
> - using FOR, REPEAT, PROC and FN instead of line numbers
> - inlining functions and subroutines
> - removing blank spaces and consolidating multiple lines
>
> Of course, some of these re-codings make the program a bit less clear, and one gets diminishing returns. It's a pity that all the % signs for integer variables are a bit ugly especially if you're not used to them: BBC Basic has no DEF INT.
>
> In the case of Gerson's program, the version below produces 12 digits in 14 seconds, compared to 43 seconds for the initial port. This version produces 36 digits in 186s. (Edit: and 100 digits in 2510s.)
>
> I was quite pleased to do without both FN definitions and especially the 5E-6 which I didn't understand or feel comfortable with. I could use DIV and MOD in a natural way.

Very good work! Thank you.
5E-6 was added to avoid errors due to binary representation. For instance, I notice a number that looked like 4482 on the BBC Micro, but its integer part was 4481. When I took its fractional part the result was 0.9999996185. So, INT(4481 + 0.999999 + 0.000005) will return 4482 as it should. Of course, this can be fixed by working with integer variables and functions whenever possible. One hundred digits in 42 minutes is quite good, considering I estimate it will take twice as long on the MSX computer (Z80 @ 3.57 Mhz). Perhaps I should improve mine also.

The multiprecision program is basically equivalent to the obfuscated one in the card, except that the expression for π is 4*W*(1/(C+D-1)+1/2) instead of W*(2/(C+D-1)+1). Also, long divisions are performed by multiplying by the inverse (for example M/(C + D) = M*1/(C + D).
I wrote the LMULT routine (multiply two n-digit numbers giving an n-digit result) many years ago. Usually if I can do something by hand I don't have trouble programming it. But when I tried to do a long division by hand, I discovered I don't remember how to do it anymore. As LDIVIDE was needed, I did 1/y using Newton's method: $x_n = x_{n-1}(2 - x_{n-1}.y)$. The initial guess, x_0, is given with the first few exact digits to ensure a faster convergence (lines 440 and

450 in the HP-75C program below). ADD, MULT and DIVID were taken from Katie Wasserman's program.

HP-75C program:

Code:

```
10 REM  **   PI BY WALLIS-WASICKI FORMULA   ***
20 ASSIGN # 1 TO 'PI'
30 OPTION BASE 0
40 INTEGER B0,D,E,G,I,I0,J,K,L,M,N,N0
50 DIM A(55),B(55),C(110),Q(55),X(55),Y(110),W(55)
60 INPUT "Number of decimal digits: "; N$ @ N0=VAL(N$)
70 B=1000000 @ N=N0 DIV 6+2 @ K=N0*12 DIV 25+1
80 T0=TIME
90 D=8*K+4 @ E=D-8 @ G=4*K*K-1
100 FOR I=0 TO N+2
```

The digits of π are written to a BASIC file with DATA lines suitable for occasional further processing (like insertion of the missing zeros at left, for example).

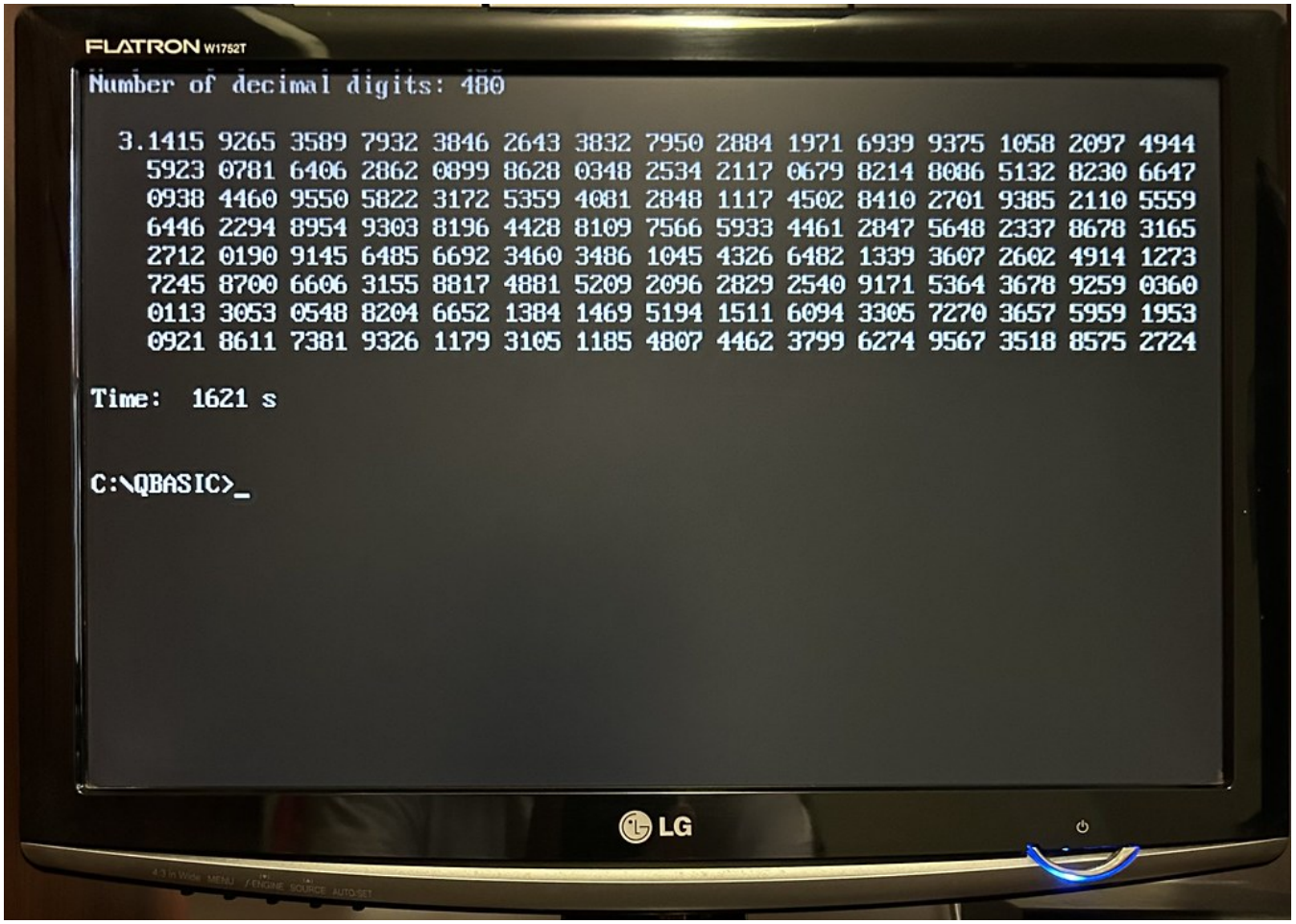
```
1 DATA 3
2 DATA 141592
3 DATA 653589
4 DATA 793238
5 DATA 462643
6 DATA 383279
7 DATA 502884
8 DATA 197169
9 DATA 399375
10 DATA 105820
11 DATA 974944
12 DATA 592307
13 DATA 816406
14 DATA 286208
15 DATA 998628
16 DATA 34825
17 DATA 342117
18 DATA 67967
19 DATA '8083.154 s'
```

100 correct digits in 135 minutes, but a really competent HP-75C programmer can surely improve on that.

P.S.: Here's a QBASIC version based on you port. The number of digits is forced to the next multiple of 4. All digits are supposed to be exact, no rounding in the last one. Because of the choice for 32-bit integers, the number of digits is limited to 480. One hundred digits in 71 seconds (DOSBox with default settings). 480 digits in about 27 minutes (when compiled). I would like to try it on my HP-200LX, but its screen has gone dark.

Code:

```
10 REM  ***  PI BY WALLIS-WASICKI FORMULA  ***
30 DEFLNG A-Y
40 DEFSNG Z
60 DIM A(125), B(125), C(250), Q(125), X(125), Y(250), W(125)
70 CLS : LINE INPUT "Number of decimal digits: "; N$: ND = VAL(N$): IF ND > 480 THEN 70
75 ND = 4 * (ND \ 4 + SGN(ND MOD 4)): LOCATE 1, 26: PRINT ND: PRINT
80 B = 10000: N = ND \ 4 + 2: K = ND * 12 \ 25 + 1
85 M = 10 * LOG(ND) / LOG(1000)
90 TM = TIMER
100 D = 8 * K + 4: E = D - 8: G = 4 * K * K - 1
```



EMAIL PM FIND QUOTE REPORT

14th March, 2022, 03:33 (This post was last modified: 23rd March, 2022 18:29 by Gerson W. Barbosa.) **Post: #22**

Gerson W. Barbosa
Senior Member

Posts: 1,500
Joined: Dec 2013

RE: Gerson's Pi Program

EdS2 Wrote: (11th March, 2022 14:42)

Code:

```
760     P%=T%DIVB%
770   NEXT
780 NEXT
790 FOR I%=N%+2 TO 2 STEP -1
800   T%=C%(I%)
810   C%(I%)=T%MODB%
820   C%(I%-1)=C%(I%-1)+T% DIV B%
830 NEXT
840 RETURN
850 REM  ***   DIVIDE   ***
```

On the MSX LOG is the natural logarithm whereas on the BBC Micro it is the decimal logarithm. This won't change the result for M% in line 85 above, but it can be simplified to

```
85 M%=10/3*LOG(ND)
```

Also, matrices are initialized to zero on both the MSX and the BBC micro. Thus, we can eliminate the lines 120 through 140 and edit the line 110 to

```
110 W%(1)=4
```

My mistake, especially because more matrices must be initiliazed to zero as you can see from the HP-75C listing above.

It's already Pi Day in countries which use MM/DD date format, so I decided to turn on my old MSX computer after 15 years to check the running time for 100 digits. The 2-byte system variable TIME allows for timings up to 1092 seconds only, but this limitation can be overcome by using an interruption to updated a timing variable every 10 seconds.

Same results on the BlueMSX emulator, in one tenth of the time.

Code:

```
10 REM PI BY WALLIS-WASICKI FORMULA
20 KEYOFF: CLS
30 MAXFILES=1: OPEN "PI" FOR OUTPUT AS #1
40 DEFFN FRAC(X) = X-INT(X)
50 DEFFN RMDR(X,Y)=X-Y*INT(X/Y)
60 DEFINT D,E,G-N
70 DIM A(50), B(50), C(100), Q(50), X(50), Y(100), W(50)
80 LINE INPUT "Number of decimal digits: ";N$: ND=VAL(N$):PRINT
90 B=10000000!: N=ND\7+2: K=ND*12\25+1
100 M=10*LOG(ND)/LOG(1000)-1
```

The matrices are dimensioned for 300 digits or so, but G will overflow for ND around 180, unless the line 60 is changed to

```
60 DEFINT D,E,H-N
```

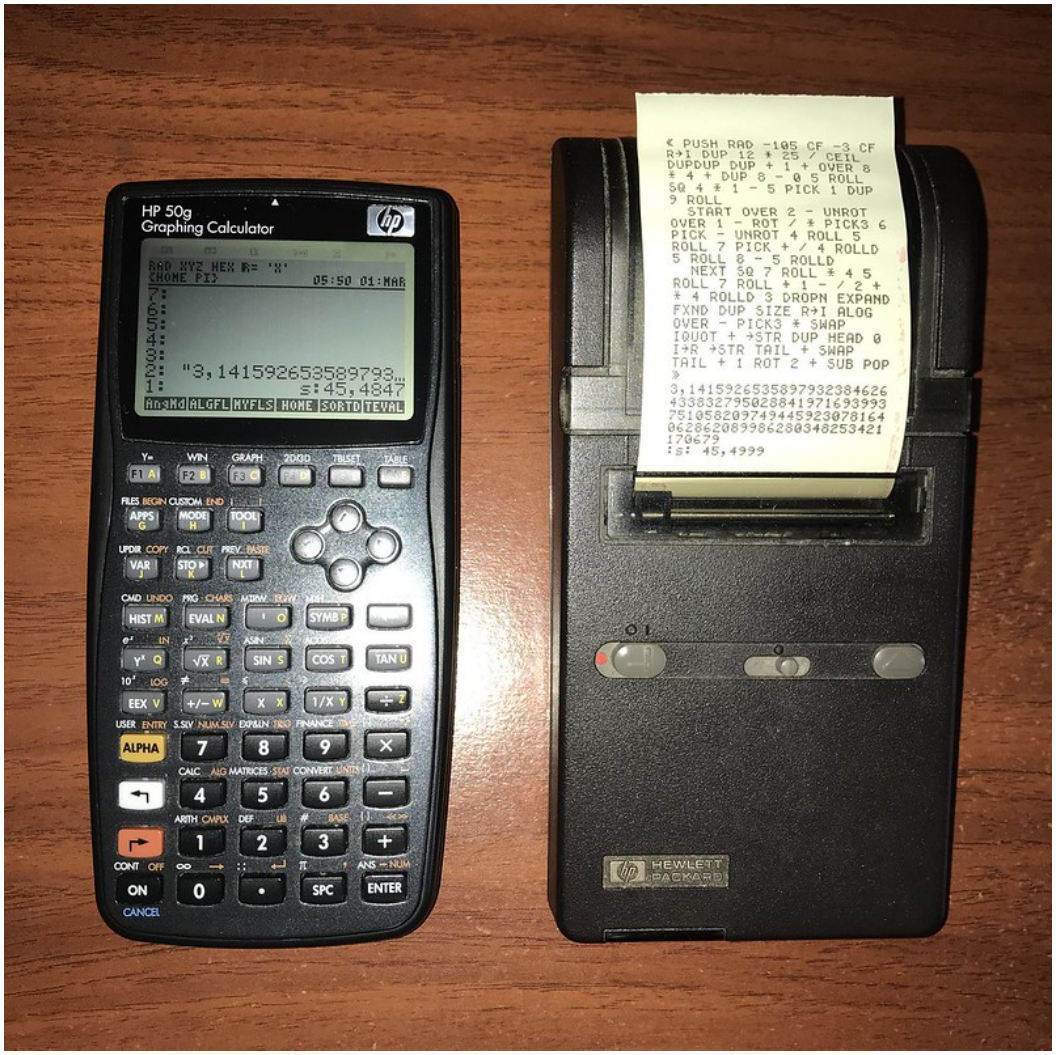
In this case, the constant in line 10 below should be changed to 1000.

Code:

```
10 CLEAR 500: MAXFILES=1: OPEN "PI" FOR INPUT AS #1: INPUT N%: INPUT #1,A$: A$=A$+".":P$=A$
20 IF EOF(1) THEN 60
30 INPUT #1,A$: L%=LEN(A$):Z$=""
40 IF L%<7 THEN FOR I%=1 TO 7-L%: Z$=Z$+"0":NEXT: A$=Z$+A$
50  P$=P$+A$: GOTO 20
60 FOR I%=1 TO N%+2: PRINT MID$(P$,I%,1);: NEXT: CLOSE #1
```

———

PS. Sure 86 minutes for just the first 100 digits is no big deal, but considering the most powerful supercomputer of the late eighties wouldn’t do it in even 100 years using the plain Wallis Product, this looks great.
The HP-49G calculator, released in 1999, does it in 137.3 seconds. In the picture, the slightly more recent HP 50g computes 100 decimal digits of π in one third of that time using an equivalent User-RPL program:



Edited to fix a couple of elusive typos.

EMAIL PM FIND QUOTE REPORT

15th March, 2022, 09:44 **Post: #23**

EdS2
Senior Member

Posts: 525
Joined: Apr 2014

RE: Gerson's Pi Program

Very nice to run it on a real machine!

EMAIL PM FIND QUOTE REPORT

« Next Oldest | Next Newest »

Enter Keywords Search Thread

NEW REPLY

View a Printable Version
Send this Thread to a Friend
Subscribe to this thread

User(s) browsing this thread: Valentin Albillo*