PIM+PIMP

Display



(HP-41CX, Hewlett Packard 1983 and DM41X, SwissMicros 2020)

Overview[1]

Two programs calculate π (pi) to N decimals with the formula of John Machin (1706): PIM and PIMP. PIM stores 5 decimals in every data register (base 100 000). PIMP is the packed version: it stores 10 decimals (two groups of 5) in every register, so about twice as many decimals fit in the same memory. Both programs set the SIZE themselves, show the maximum number of decimals when the memory is too small, measure the calculation time with the stopwatch and show π 10 decimals per line. The formulas used are:

$$\pi = 16 \cdot \mathrm{atan}(1/5) - 4 \cdot \mathrm{atan}(1/239)$$

$$\mathrm{atan}(1/x) = 1/x - 1/3x^3 + 1/5x^5 - 1/7x^7 + \ldots$$

The series for 1/5 and 1/239 are added term by term until the term becomes zero. Both programs use the built-in X-Functions (PSIZE, SIZE?, ALENG) and Time functions (SETSW, RUNSW, STOPSW, RCLSW, ATIME24) of the HP-41CX and DM41X. More about the method: Machin-like formula (Wikipedia), Machin's Formula (Wolfram MathWorld) and John Machin (Wikipedia).

Description

PIM keeps two multi-precision numbers in the data registers: T, the current term of the series, and S, the sum that becomes π. Each number takes n = INT(N/5)+3 registers of 5 decimals (SIZE = 2n+12); the first register holds the integer part and the last registers are guard digits, so N is rounded down to a multiple of 5. Subroutine LBL 01 adds ±T·atan(1/x) to S: T starts with the coefficient (16 or 4) and is divided by x (5 or 239), from then on by x^2 for every term. Each term is also divided by 2k+1 and added to or subtracted from S, register by register, with the remainder carried to the next register. Registers of T that have become zero are skipped, so every term is faster than the previous one, and the series stops when T is zero. Finally the carries in S are normalised from right to left (MOD 100 000), the time is shown and π is displayed in groups of 5 with leading zeros.

PIMP works the same way, but packs two groups of 5 decimals into one register (R = hi·100 000 + lo). It needs n = INT((N+9)/10)+2 registers per number (SIZE = 2n+18), and N is rounded up to a multiple of 10. Every register is unpacked, divided group by group and packed again, so all intermediate values stay below 10^{10} and remain exact. S is kept in a "balanced" form (the high group may be negative) and a small carry goes to the register on the left; at the end one pass from right to left (MOD 10^{10}) turns S into normal digits. The program is longer than PIM, but the number of decimals that fits in memory is almost doubled.

[1] *This program is copyright and is supplied without representation or warranty of any kind. The author assumes no responsibility and shall have no liability, consequential or otherwise, of any kind arising from the use of this program material or any part thereof*

 ©2006-2026 Auke Hoekstra

Metrics

METRIC	PIM V41	PIM HP-41CX	PIMP V41	PIMP HP-41CX
Dec./register	5	5	10	10
Program size	375 bytes	375 bytes	498 bytes	498 bytes
Max decimals	615	615	1120	1120
Time 20 dec.	0:00:24	≈10 min	0:00:52	≈22 min
Time 100 dec.	≈0:03:52	≈1.6 hours	≈0:07:46	≈3.2 hours
Time 600 dec.	1:33:13	≈1.6 days	≈2:55:20	≈3.0 days
Time max dec.	≈1:37:42	≈1.7 days	≈9:37:13	≈10.0 days

All V41 times were measured with V41 at top speed and maximum turbo: PIM with 20 and 600 decimals and PIMP with 20 decimals. The other times (≈) are estimates: the run time grows with N^2, and PIMP needs about 1.8 times as many program steps per decimal as PIM. The V41 at maximum turbo runs about 200 program steps per second and a real HP-41CX about 8, so the HP-41CX times are estimated as 25 times the V41 times. The DM41X has the same memory as the HP-41CX, so the maximum number of decimals is the same; it runs faster than the HP-41CX, but its times have not been measured yet.

Example (1)

KEYSTROKES	DISPLAY	COMMENTS
		Calculate π with 20 decimals
[XEQ] [ALPHA] PIM [ALPHA]	DEC=?	Number of decimals N
20 [R/S]	T=00:00:24	Calculation time (stopwatch)
[R/S]	PI=3,	Integer part of π
[R/S]	14159 26535	Decimals 1-10
[R/S]	89793 23846	Decimals 11-20
[R/S]	0,00000	End: FIX 5 and flag 29 restored
		Calculate π with 600 decimals
[XEQ] [ALPHA] PIM [ALPHA]	DEC=?	Number of decimals N
600 [R/S]	T=0 01:33:13	Calculation time (stopwatch)
[R/S]	PI=3,	Integer part of π
[R/S]	14159 26535	Decimals 1-10
[R/S]	89793 23846	Decimals 11-20
…		Decimals 21-580 (R/S each line)
[R/S]	84674 81846	Decimals 581-590
[R/S]	76694 05132	Decimals 591-600
[R/S]	0,00000	End: FIX 5 and flag 29 restored
		Not enough memory for N decimals
[XEQ] [ALPHA] PIM [ALPHA]	DEC=?	Number of decimals N
2000 [R/S]	MAX=615	Maximum decimals (depends on memory)

©2006-2026 Auke Hoekstra

Example (2)

KEYSTROKES	DISPLAY	COMMENTS
		Calculate π with 20 decimals (packed)
[XEQ] [ALPHA] PIMP [ALPHA]	DEC=?	Number of decimals N
20 [R/S]	T=00:00:52	Calculation time (stopwatch)
[R/S]	PI=3,	Integer part of π
[R/S]	14159 26535	Decimals 1-10 (register 1)
[R/S]	89793 23846	Decimals 11-20 (register 2)
[R/S]	23.846,00000	End: last group in X, FIX 5, SF 29
		Calculate π with 1120 decimals (maximum)
[XEQ] [ALPHA] PIMP [ALPHA]	DEC=?	Number of decimals N
1120 [R/S]	T=09:41:23	Calculation time (stopwatch)
[R/S]	PI=3,	Integer part of π
[R/S]	14159 26535	Decimals 1-10
[R/S]	89793 23846	Decimals 11-20
…		Decimals 21-1100 (R/S each line)
[R/S]	55748 57242	Decimals 1101-1110
[R/S]	45415 06959	Decimals 1111-1120
[R/S]	6.959,00000	End: last group in X, FIX 5, SF 29
		Not enough memory for N decimals
[XEQ] [ALPHA] PIMP [ALPHA]	DEC=?	Number of decimals N
2000 [R/S]	MAX=1120	Maximum decimals (depends on memory)

Program Listing (PIM)

The listing of PIM is given below. The decimals are calculated first; the time is measured with the stopwatch. Only the global label PIM is used.

```
01▪LBL "PIM"
02 FIX 0
03 CF 29
04 "DEC=?"
05 PROMPT
06 ,
07 SETSW
08 RDN
09 RUNSW
10 5
11 /
12 INT
13 3
14 +
15 ENTER
16 ST+ X
17 12
18 +
19 SF 25
20 PSIZE
21 FS?C 25
22 GTO 10
23 SIZE?
24▪LBL 11
25 RCL Y
26 RCL Y
27 +
28 2
29 /
30 INT
31 X=Y?
32 GTO 12
33 SF 25
34 PSIZE
35 FS?C 25
36 GTO 13
37 X<>Y
38 GTO 11
39▪LBL 13
40 X<>Y
41 RDN
42 GTO 11
43▪LBL 12
44 PSIZE
45 2
46 /
47 INT
48 9
49 -
50 5
51 *
52 "MAX="
53 ARCL X
54 AVIEW
55 GTO 14
56▪LBL 10
57 CLRG
58 RDN
59 STO 06
60 11
61 +
62 1 E3
63 /
64 STO 11
65 1 E5
66 STO 10
67 16
68 STO 12
69 1
70 5
71 XEQ 01
72 4
```


©2006-2026 Auke Hoekstra

```
 73 STO 12
 74 -1
 75 239
 76 XEQ 01
 77 RCL 06
 78 12
 79 +
 80 1 E3
 81 /
 82 RCL 06
 83 2
 84 *
 85 +
 86 11
 87 +
 88 STO 00
 89 ,
 90 STO 01
 91▪LBL 06
 92 RCL IND 00
 93 RCL 01
 94 +
 95 RCL X
 96 RCL 10
 97 MOD
 98 STO IND 00
 99 -
100 RCL 10
101 /
102 STO 01
103 DSE 00
104 GTO 06
105 RCL 01
106 ST+ IND 00
107 STOPSW
108 FIX 4
109 RCLSW
110 "T="
111 ATIME24
112 PROMPT
113 FIX 0
114 CLA
115 "PI="
116 ARCL IND 00
117 >","
118 PROMPT
119 CLA
120 RCL 06
121 13
122 +
123 RCL 06
124 2
125 *
126 9
127 +
128 1 E3
129 /
130 +
131 STO 00
132▪LBL 07
133 RCL IND 00
134 1 E4
135 X>Y?
136 >"0"
137 RDN
138 1 E3
139 X>Y?
140 >"0"
141 RDN
142 100
143 X>Y?
144 >"0"
145 RDN
146 10
147 X>Y?
148 >"0"
149 RDN
150 ARCL X
151 ALENG
152 6
153 X>Y?
154 GTO 08
155 PROMPT
156 CLA
157 GTO 09
158▪LBL 08
159 >" "
160▪LBL 09
161 ISG 00
162 GTO 07
163 ALENG
164 X#0?
165 AVIEW
166 GTO 14
167▪LBL 01
168 STO 03
169 X^2
170 STO 09
171 RDN
172 STO 05
173 1
174 STO 04
175 12
176 STO 08
177▪LBL 02
178 ,
179 STO 01
180 STO 02
181 RCL 08
182 RCL 11
183 +
184 STO 00
185 RCL 08
186 RCL 06
187 +
188 ,999
189 +
190 STO 07
191▪LBL 03
192 RCL 01
193 RCL 10
194 *
195 RCL IND 00
196 +
197 RCL X
198 RCL 03
199 MOD
200 STO 01
201 -
202 RCL 03
203 /
204 STO IND 00
205 RCL 02
206 RCL 10
207 *
208 +
209 RCL X
210 RCL 04
211 MOD
212 STO 02
213 -
214 RCL 04
215 /
216 RCL 05
217 *
218 ST+ IND 07
219 ISG 07
220 ISG 00
221 GTO 03
222▪LBL 04
223 RCL IND 08
224 X#0?
225 GTO 05
226 1
227 ST+ 08
228 RCL 06
229 11
230 +
231 RCL 08
232 X>Y?
233 RTN
234 GTO 04
235▪LBL 05
236 RCL 09
237 STO 03
238 2
239 ST+ 04
240 RCL 05
241 CHS
242 STO 05
243 GTO 02
244▪LBL 14
245 FIX 5
246 SF 29
247 END
```

(375 bytes)

Program Listing (PIMP)

The listing of PIMP is given below. The decimals are calculated first; the time is measured with the stopwatch. Only the global label PIMP is used.

©2006-2026 Auke Hoekstra

```
01▪LBL "PIMP"
02 FIX 0
03 CF 29
04 "DEC=?"
05 PROMPT
06 ,
07 SETSW
08 RDN
09 RUNSW
10 9
11 +
12 10
13 /
14 INT
15 2
16 +
17 RCL X
18 ST+ X
19 18
20 +
21 SF 25
22 PSIZE
23 FS?C 25
24 GTO 10
25 SIZE?
26▪LBL 11
27 RCL Y
28 RCL Y
29 +
30 2
31 /
32 INT
33 X=Y?
34 GTO 12
35 SF 25
36 PSIZE
37 FS?C 25
38 GTO 13
39 X<>Y
40 GTO 11
41▪LBL 13
42 X<>Y
43 RDN
44 GTO 11
45▪LBL 12
46 PSIZE
47 18
48 -
49 2
50 /
51 INT
52 2
53 -
54 10
55 *
56 "MAX="
57 ARCL X
58 PROMPT
59 GTO 14
60▪LBL 10
61 CLRG
62 RDN
63 STO 06
64 16
65 +
66 1 E3
67 /
68 STO 11
69 1 E5
70 STO 10
71 5 E4
72 STO 13
73 16
74 STO 17
75 1
76 5
77 XEQ 01
78 4
79 STO 17
80 -1
81 239
82 XEQ 01
83 RCL 06
84 18
85 +
86 1 E3
87 /
88 RCL 06
89 2
90 *
91 17
92 +
93 +
94 STO 00
95 ,
96 STO 01
97 1 E10
98 STO 14
99▪LBL 06
100 RCL IND 00
101 RCL 01
102 +
103 RCL X
104 RCL 14
105 MOD
106 STO IND 00
107 -
108 RCL 14
109 /
110 STO 01
111 DSE 00
112 GTO 06
113 RCL 01
114 ST+ IND 00
115 STOPSW
116 FIX 4
117 RCLSW
118 "T="
119 ATIME24
120 PROMPT
121 FIX 0
122 CLA
123 "PI="
124 ARCL IND 00
125 >","
126 PROMPT
127 RCL 06
128 19
129 +
130 RCL 06
131 2
132 *
133 16
134 +
135 1 E3
136 /
137 +
138 STO 00
139▪LBL 07
140 CLA
141 RCL IND 00
142 RCL 10
143 /
144 INT
145 XEQ 09
146 >" "
147 RCL IND 00
148 RCL 10
149 MOD
150 XEQ 09
151 PROMPT
152 ISG 00
153 GTO 07
154 GTO 14
155▪LBL 09
156 1 E4
157 X>Y?
158 >"0"
159 RDN
160 1 E3
161 X>Y?
162 >"0"
163 RDN
164 100
165 X>Y?
166 >"0"
167 RDN
168 10
169 X>Y?
170 >"0"
171 RDN
172 ARCL X
173 RTN
174▪LBL 01
175 STO 03
176 X^2
177 STO 09
178 RDN
179 STO 05
180 1
181 STO 04
182 17
183 STO 08
184▪LBL 02
185 ,
186 STO 01
187 STO 02
188 RCL 08
189 RCL 11
190 +
191 STO 00
192 RCL 08
193 RCL 06
194 +
195 1
196 +
197 ,999
198 +
199 STO 07
200 1
201 -
202 STO 12
203▪LBL 03
204 RCL IND 00
205 RCL X
206 RCL 10
207 MOD
208 STO 14
209 -
210 RCL 10
211 /
212 RCL 01
213 RCL 10
214 *
215 +
216 RCL X
```


©2006-2026 Auke Hoekstra

```
217 RCL 03
218 MOD
219 STO 01
220 -
221 RCL 03
222 /
223 STO 15
224 RCL 01
225 RCL 10
226 *
227 RCL 14
228 +
229 RCL X
230 RCL 03
231 MOD
232 STO 01
233 -
234 RCL 03
235 /
236 STO 14
237 RCL 15
238 RCL 10
239 *
240 +
241 STO IND 00
242 RCL 02
243 RCL 10
244 *
245 RCL 15
246 +
247 RCL X
248 RCL 04
249 MOD
250 STO 02
251 -
252 RCL 04
253 /
254 STO 15
255 RCL 02
256 RCL 10
257 *
258 RCL 14
259 +
260 RCL X
261 RCL 04
262 MOD
263 STO 02
264 -
265 RCL 04
266 /
267 STO 14
268 RCL IND 07
269 RCL X
270 RCL 10
271 MOD
272 STO 16
273 -
274 RCL 10
275 /
276 RCL 15
277 RCL 05
278 *
279 +
280 STO 15
281 RCL 14
282 RCL 05
283 *
284 RCL 16
285 +
286 RCL X
287 RCL 10
288 MOD
289 STO 16
290 -
291 RCL 10
292 /
293 RCL 15
294 +
295 STO 15
296 RCL 13
297 +
298 RCL 10
299 MOD
300 RCL 13
301 -
302 RCL 15
303 RCL Y
304 -
305 RCL 10
306 /
307 ST+ IND 12
308 RDN
309 RCL 10
310 *
311 RCL 16
312 +
313 STO IND 07
314 ISG 12
315 ISG 07
316 ISG 00
317 GTO 03
318▪LBL 04
319 RCL IND 08
320 X#0?
321 GTO 05
322 1
323 ST+ 08
324 RCL 06
325 16
326 +
327 RCL 08
328 X>Y?
329 RTN
330 GTO 04
331▪LBL 05
332 RCL 09
333 STO 03
334 2
335 ST+ 04
336 RCL 05
337 CHS
338 STO 05
339 GTO 02
340▪LBL 14
341 FIX 5
342 SF 29
343 END
```

(498 bytes)

Registers, Labels and Flags (PIM)

REGISTERS	COMMENTS
R00	Loop pointer ISG/DSE
R01	Carry 1 (divide by x)
R02	Carry 2 (divide by 2k+1)
R03	Divisor x or x²
R04	2k+1
R05	Sign +1/-1
R06	n=INT(N/5)+3
R07	Pointer into S
R08	First non-zero T register
R09	x²
R10	100000 (base)
R11	(11+n)/1000

LABELS	COMMENTS
LBL01	S=S±T·atan(1/x)
LBL02	One term of the series
LBL03	Inner loop per register
LBL04	Skip leading zeros of T
LBL05	Next term
LBL06	Normalise S (carries)
LBL07	Display loop
LBL08	Space after 1st group
LBL09	Next group
LBL10	Memory fits: calculate
LBL11	Search largest SIZE
LBL12	Show MAX decimals

©2006-2026 Auke Hoekstra

R12...	T: term, n regs (R12=int)	LBL13	SIZE fits: lo=mid
R12+n...	S: π, n regs (2 guard)	LBL14	Exit: FIX 5, SF 29

FLAGS	COMMENTS
25	Error ignore: test if PSIZE fits
29	Cleared: no separators in ARCL of integers

Registers, Labels and Flags (PIMP)

REGISTERS	COMMENTS	LABELS	COMMENTS
R00	Loop pointer ISG/DSE	LBL01	S=S±T·atan(1/x)
R01	Remainder 1 / carry	LBL02	One term of the series
R02	Remainder 2 (2k+1)	LBL03	Inner loop per register
R03	Divisor x or x²	LBL04	Skip leading zeros of T
R04	2k+1	LBL05	Next term
R05	Sign +1/-1	LBL06	Normalise S (MOD 10^10)
R06	n=INT((N+9)/10)+2	LBL07	Display loop
R07	Pointer into S	LBL09	Group with leading zeros
R08	First non-zero T register	LBL10	Memory fits: calculate
R09	x²	LBL11	Search largest SIZE
R10	100000 (group base)	LBL12	Show MAX decimals
R11	(16+n)/1000	LBL13	SIZE fits: lo=mid
R12	Pointer to S on the left	LBL14	Exit: FIX 5, SF 29
R13	50000 (balance)		
R14	Scratch / 10^10		
R15	Scratch		
R16	Scratch		
R17...	T: term, n regs (R17=int)		
R17+n	Carry sink		
R18+n...	S: π, n regs (packed)		

FLAGS	COMMENTS
25	Error ignore: test if PSIZE fits
29	Cleared: no separators in ARCL of integers

Downloads

The RAW/TXT format of the programs is available via the website: PIM (in zip file).

©2006-2026 Auke Hoekstra