

RUBYを使ってπを計算

(1) BigMathのPi関数を使って100桁まで計算

```
require 'bigdecimal/math'
require 'benchmark'
result = Benchmark.realtime do
  puts BigMath::PI(100).to_s
end
puts "Time: #{result}s"
```

計算結果:

```
0.314159265358979323846264338327950288419716939937510582097494459230781640628
620899862803482534211706798214808651328230 4863 86308 83606 928E1
Time: 0.000433847s
```

評価は、116桁まで真値に等しい。また、使い勝手は電卓並だ。

(2) ガウス＝ルジャンドルの反復アルゴリズムを、Bigmathを使い、Rubyでπを計算

a, b が希望する桁数になるまで以下の漸化式の計算を繰り返す。繰り返し回数は、 $\log_2 n$ 回。小数第100位まで求めるときで、 $\log_2 100 = 6.64$ 回程度。100万桁で20回程度の反復でよい。

$$\begin{aligned}a_0 &= 1 & b_0 &= \frac{1}{\sqrt{2}} & t_0 &= \frac{1}{4} & p_0 &= 1 \\ a_{n+1} &= \frac{a_n + b_n}{2} \\ b_{n+1} &= \sqrt{a_n b_n} \\ t_{n+1} &= t_n - p_n (a_n - a_{n+1})^2 \\ p_{n+1} &= 2p_n\end{aligned}$$

πは、以下の式で近似できる

$$\pi \approx \frac{(a + b)^2}{4t}$$

```

require "bigdecimal"
require "benchmark"

result = Benchmark.realtime do
  prec = 100
  conv = 6
  a = BigDecimal("1")
  b = BigDecimal("1") / BigDecimal("2").sqrt(prec)
  t = BigDecimal("1") / 4
  p = BigDecimal("1")

  for n in 1..conv do
    an = (a + b) / 2
    b = (a * b).sqrt(prec)
    t -= p * (an - a) * (an - a)
    p *= 2
    a = an
  end

  puts (a + b) * (a + b) / (4 * t)
end

puts "Time: #{result}s"

```

結果はお見事、簡単に、しかも約0.2秒と高速に計算できた

上位の桁を検証すると108桁まで正しい。

真値3.

14159 26535 89793 23846 26433 83279 50288 41971 69399 37510 58209 74944 59230 78164
06286 20899 86280 34825 34211 70679 82148 08651 32823

計算3.

14159 26535 89793 23846 26433 83279 50288 41971 69399 37510 58209 74944 59230 78164
06286 20899 86280 34825 34211 70679 82148 08674 76768 . . .

全桁計算結果:

0.314159265358979323846264338327950288419716939937510582097494459230781640628
6208998628034825342117067982148086747676780573856883629512666379599775013819
2766523475824380200769264901536893369839504428405340096343217291836790401541
8877044021855254487072688533839876768888861331503778126554683688821395087206
8810802511995606309472786003851657816477028430682528450312728477235288284252
8658914477164405698853275283901437649918024516559323749675500095162515213948
9811681034967119815596165286577567212517013895058239779042113694646097577661
8392621533882120274447335581231226279665102272599242753378508272522799023620
9522651605118837101856164559964259151218227474365434719711745383809195485525
5852915101273028455910975962845833759039143680321999519411406582655351391920

4983169653021366465516351336572378404721284407006321646783539940128878508418
4087449325755046265431969121025302969864860085419149175054435051439775375511
8600291979155365561782941710239200709190362969621066293013190008540188995474
0156203204024074185853380768872447493907089776735297005381314325559995181914
6366342858316970836134105209389585641916572995357115473479980311564731089545
4274111835097827536912568335361723596506943365973254513054680817835930453784
2326877957629859652534434366395770147521881969890467505837235982515220488004
0883148116460622501153772367468348366770941423719592898327865379158693616152
4451280536534003616576656001483433650390154335043392515295774039549546106580
4251882702054702586570745535553874962355564246997942477346384578408035770877
7048764011220797780793615322564648806999078585257916101538418219955508560589
6601372217327715164932533352155657088954268998145027149677268203929135869323
4620562839855556751697229600091041182819958463752701984589153790086429668459
2485773668479841867312404811020379445153140470799831436756427257920791011224
3730083764841592080037593418046101346487231171654245776606503992557425994976
5341667933850512761965774146170571545503729879750546851808658837793134540699
0557293480009212856517715349895869261585912797278006336660800050158906617686
5024430831810052536130021973824411872908294082861846420500465946611672697689
3470535928381247757909788748774340249449882041329361787392460896662590585152
2530367091386185021678974641259339015166421340857273198041877495184814395485
2435186245406042222482267715113117315496719288225872948534173021864603817264
3418001798092207730215257435481740033624391457943581604185446887158871199814
9761645718097968829982839003204544277056109466114981523549852249903064680224
11869770306328792099499463850646312655408138808826481113261690276339224211808
6553395968302252779801760869073412533879457208751590316536369691212816773209
6691358390540555003328770795775741532633387688855796948308486609046078086236
2596650700412251768593217006400354042142092817813380752137901946577351712525
2936572723129442421927776769372121427818292780204839832247237097373885013445
2075739167423579635963834546301444661886604414654725641824739599466455189953
8775609419679006476715114657314001751398740696852068272326798226850539279217
0926428164397067256698189316552455119492379616085674995701998037029796159969
2584621033512941886549961742560864778673081730547902249953912210899734037961
9973008110695323078876067318984051059020373918613252741145475106679564823555
1284315192145980051244300361818783505855408199780024251176527115696503935803
0196419185749126155234339571899387489514165353500131844341958167677855126237
4502730227823286892577963717242536529834869567574450342257339272038208028047
2416111382220809350838952132715584569049430024487569473014664871817036061536
7378477164971035684495381238631592822390530340692657010935120875107147118302
1612181609150235958812118513782800536888782186895609182978911451609283614203
2020795657333015906691293053844572537066782734589700282489714584718147709518
6849961079217962834142686182730197954353448198476749691578732170802133362993
7895538398039224342168232043901160853562099152246933873866042409203418842977
1814778182216083063582276318787527807772060129293369298660920553443318253858
5242219665917267975188263704428029639075271576917623642640152192236225682588
7205831830611058170646352006825917374881410461140536974354459963511062849109
9500045366467971389122412235467857599627847005185316224938784565237778806516
6940205671076615095194831855069011895280438079834837695883511718593212923807
1024262166742575719903054796133978973715123008838974624002460807413219693851
9618812309500722608493459190267778769176644805531272014719810367516284974676

```
5395473544707842830651023070405632324829223763157915883431644483830239414463
0101363983365868147785757900178511510756883274688340389458409082243870774812
1739323072504964467968933288854884189488693445716161436122617931526284059791
2174582692241528516732360985313904789102247115707767226925271987015997337913
4125923978636371611906820484524185631193469481189422602278176644253992972697
1386759274722806939086991935188593315724421093418881309112017494320371375091
1714526201387543545374413564723834536305530272088253385625959362911485289456
8469805127333894342297319714675098973870867583451679558174652663435786304899
2953603571628912111552045896691531277024922610353898730210189232692316003005
4544478598034949849286520289799536859402622757036785252995766153984174648687
4109504507242830164455340323047231217818E1
Time: 0.189022363s
```

参考までに、PYTHONで解いてみると、

```
## 円周率を求める
```

```
## arctan(1/n)をTaylor展開で求める
```

```
def arctan(p, n):
```

```
    x = p/n
    nn = n*n
    c = 1
    s = x
    k = 1
    while x > 0:
        x /= nn
        k += 2
        c = -c
        s += c*(x/k)
    return s
```

```
## 整数部分「3」も含むmantissa仮数表現の長整数で返します
```

```
def pi_mantissa(digit, redund=10):
```

```
    p = 10**digit
    q = 10**redund
    pi *= q
    pi = 4*(12*arctan(p,18)+8*arctan(p,57)-5*arctan(p,239))
    pi /= q
    return pi
```

```
print (pi_mantissa(100))
```

```
print (pi_mantissa(200))
```

計算結果:

```
3.141592653589794e+100
```

```
3.141592653589793e+200
```