rubyによるπの近似値計算

情報班：森川大翔　市村奉紀

１．研究内容

rubyというプログラミング言語を用いて円周率を近似した。今回は、ライプニッツの公式とガウス＝ルジャンドルのアルゴリズムを用いて、二つの方法でπの近似値を求めた。rubyとは、日本で作成されたプログラミング言語の一つで、日本のWEB系の企業でも使用されている。

２．ライプニッツの公式について

ライプニッツの公式とは次の式である。

$$\sum_{n=1}^{\infty}\frac{(-1)^n}{2n+1}=4\pi$$

上の式を、数Ｂの数列の単元で学習した、数列の和の公式を用いてπについて解くと

$$\pi=\sum_{n=0}^{\infty}\frac{8}{16n^2+16n+3}$$

となる。これを（A）とする。

３．ガウス＝ルジャンドルのアルゴリズムについて

【初期値の設定】初期値（a_0,b_0,t_0,p_0）を設定する…(B)

$$a_0=1,b_0=\frac{1}{\sqrt{2}},t_0=\frac{1}{4},p_0=1$$

【反復式】反復式を次のような漸化式によって定める…(C)

$$a_{n+1}=\frac{a_n+b_n}{2}\qquad b_{n+1}=\sqrt{a_nb_n}\qquad t_{n+1}=t_n-p_n(a_n-a_{n+1})^2\qquad p_{n+1}=2p_n$$

【πの算出】円周率πはa, b, tを用いて次のように近似される…(D)

$$\pi\fallingdotseq\frac{(a_n+b_n)^2}{4t_n}$$

以上をガウス＝ルジャンドルのアルゴリズムという。小数点以下n桁まで求めたいときはおおよそ$\log_2 n$回の反復でよいとされている。また１回の反復で小数点以下２桁（$\pi=3.14\ldots$）まで求められることからわかるように非常に収束が早いので、「スーパーπ」というソフトのプログラムに用いられていて、2009年に2兆6000億桁まで計算された。

４．プログラミング言語化

２．と３．で紹介したものを ruby で書き換えると次のようになる。ただし、左側の数字は行数を表している。

・ライプニッツの公式を言語化したもの

```ruby
# Your code here
def pi(x)
    p = 0.0
    n = 0.0
    while(n <= x)
        p = P + (8.0 / (16 * n * n + 16 * n + 3))
        n = n + 1.0
    end
    puts puts
end
puts pi(10000000)
```

6 行目が 2. の（A）にあたる。11 行目の()の中で 6 行目の式の n に何を代入するかを決める。

５．結果と考察

一千万を代入することで 1 秒足らずで 7 桁目まで求めることができたが、それ以降の桁数を求めることはできなかった。

・ガウス＝ルジャンドルのアルゴリズムを言語化したもの

```ruby
require 'benchmark'
require 'begdecimal'
require 'bigdecimal/math'

def pi(n)
    k = 1
    a = 1
    b = 1 / BigMath.sqrt(BigDecimal(2), 100)
    t = 1 / 4.0
    p = 1
    for k in 1.. n do
        k = k + 1
        x = a * b
        t = t - p * (((a - b) / 2) ** 2)
        a = (a + B) / 2
        b = BigMath.sqrt(BigDecimal(x), 100)
        p = 2 * p
    end
    ((a + b) ** 2) / (4 * t)
end

for m in 1.. 5 do
    for n in 1.. 10 do
        result = Benchmark.realtime do
            puts  pi(n).to_s
        end
        puts "Time: #{result} [s]"
    end
end
```

３．の(B)(C)(D)はそれぞれ 6-10 行目、11－18 行目、19 行目にあたる。また、22－29 行目は(B)-(D)のプログラミングを実際に動かす部分である。22－29 行目で円周率の計算をし、同時に計算にかかる時間を計っている。

５’．結果結果と考察

・10 回目まで計算を行うと、6 回目で小数点以下 108 桁まで求められたがそれ以降求められた桁数に変化はなかった。

・$\log_2 n$の値が n とほとんど一致していた。

５．考察と今後

ガウス＝ルジャンドルのアルゴリズムを用いるとかなり効率が良いことが分かった。将来的にはより効率のいい円周率の近似値の求め方を言語化したい。