数値計算法　演習課題 2　提出日：2024 年 4 月 28 日

202310330　長田悠生

課題 1

(1-1)

(1-1) のソースコード

```
module report1_1
    using Random
    Random.seed!(0)
    function MonteCarlo(n::Int64)
        if n <= 0
            throw("n have to larger than 0.")
        end
        x::Vector{Float64} = zeros(Float64, n)
        y::Vector{Float64} = zeros(Float64, n)
        for i::Int64 = 1:n
            x[i] = rand()
            y[i] = rand()
        end
        r::Vector{Float64} = zeros(Float64, n)
        for i::Int64 = 1:n
            r[i] = x[i]^2 + y[i]^2
        end
        m::Int64 = 0
        for i::Int64 = 1:n
            if r[i] <= 1
                m = m + 1
            end
        end
        p::Float64 = 4*m/n
        return p
    end
end
```

数値計算法　演習課題 2　提出日：2024 年 4 月 28 日

202310330　長田悠生

前ページの関数について具体的な n の値を入れてみたときのソースコードと出力結果です。前ページのソースコードと以下のソースコードは同一のファイルに書いています。

上記の関数に具体的な n の値を入れてみたときのソースコード

```
using .report1_1

p_1::Float64 = report1_1.MonteCarlo(1)
println("p_1 = $p_1")

p_100::Float64 = report1_1.MonteCarlo(100)
println("p_100 = $p_100")

p_0::Float64 = report1_1.MonteCarlo(0)
println("p_0 = $p_0")
```

実行結果

```
  p_1 = 4.0
  p_100 = 3.28
  ERROR: LoadError: "n have to larger than 0."
  Stacktrace:
   [1] MonteCarlo(n::Int64)
     @ Main.report1_1 C:\Users\admin\Documents\work-space\Julia\
        numeric-calculation\class2_report\src\report1-1.jl:9
   [2] top-level scope
     @ C:\Users\admin\Documents\work-space\Julia\numeric-
        calculation\class2_report\src\report1-1.jl:48
  in expression starting at C:\Users\admin\Documents\work-space\
     Julia\numeric-calculation\class2_report\src\report1-1.jl:48
```

数値計算法　演習課題 2　提出日：2024 年 4 月 28 日

202310330　長田悠生

(1-2)

(1-2) のソースコード

```
module monte_carlo
using Random
Random.seed!(0)
function MonteCarlo(n::Int64)
    x::Vector{Float64} = zeros(Float64, n)
    y::Vector{Float64} = zeros(Float64, n)
    for i::Int64 = 1:n
        x[i] = rand()
        y[i] = rand()
    end
    r::Vector{Float64} = zeros(Float64, n)
    for i::Int64 = 1:n
        r[i] = x[i]^2 + y[i]^2
    end
    m::Int64 = 0
    for i::Int64 = 1:n
        if r[i] <= 1
            m = m + 1
        end
    end
    p::Float64 = 4*m/n
    return p
end

function MonteCarloData(size::Int64)
    n_vec::Vector{Int64} = zeros(Int64, size)
    result_vec::Vector{Float64} = zeros(Float64, size)
    for i::Int64 = range(1, size, size)
        n::Int64 = 10^i
        result::Float64 = MonteCarlo(n)
        n_vec[i] = n
        result_vec[i] = abs(result - pi)
    end
```

数値計算法　演習課題 2　提出日：2024 年 4 月 28 日

202310330　長田悠生

```
        return n_vec, result_vec
    end
end

using .monte_carlo
using Plots

n_vec::Vector{Int64}, result_vec::Vector{Float64} = monte_carlo.
    MonteCarloData(6)

println("n_vec = $n_vec")
println("result_vec = $result_vec")

plot(n_vec, result_vec, xaxis=:log)

savefig("report1-2.png")
```

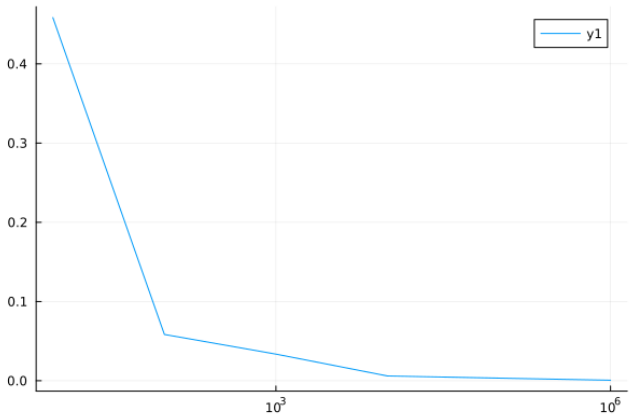

図 1　実行結果のグラフ

課題 2

(2-1)