Information Science
11: Monte Carlo Simulation

Naonori Kakimura
垣村尚徳
kakimura@global.c.u-tokyo.ac.jp

Monte-Carlo Method

(A city in Monaco, which is famous for gambling)

- Methods to use (pseudo)random numbers to solve problems

- Two main problems
  - Deterministic problems
    - Compute an integral using random numbers
      - Ex. Approximation of pi
  - Nondeterministic problems (not covered in the class)
    - Simulation in natural science/social science
      - To analyze a problem hard to solve exactly
      - For problem behaving in a probabilistic way

Approximation of pi

- put n points randomly in the square region

$$P = \{0 \le x \le 1, \quad 0 \le y \le 1\}$$

- count the number m of points in the quarter round

$$Q = \{x^2 + y^2 < 1, \quad 0 \le x \le 1, \quad 0 \le y \le 1\}$$

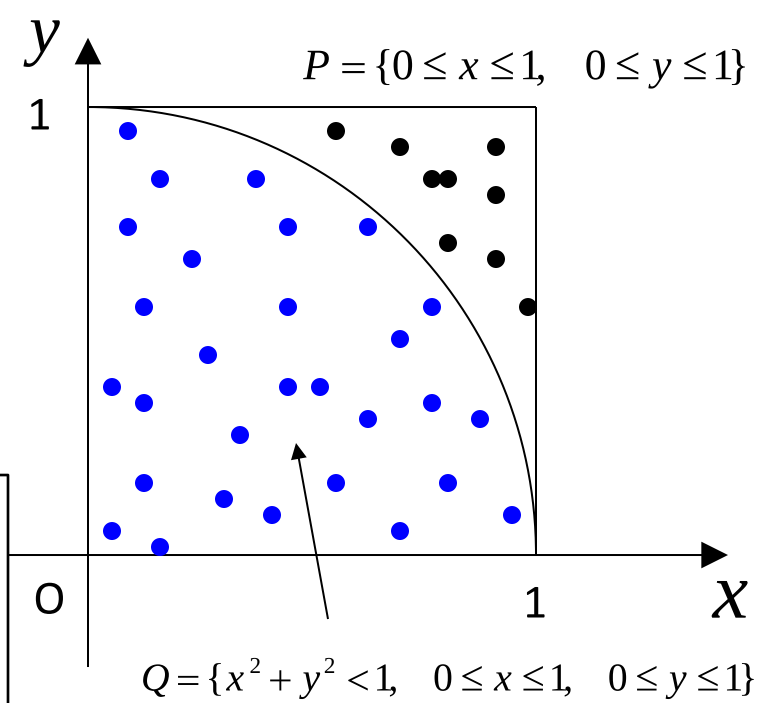
- Ratio m/n ≈ (prob. that a point is in Q) = Q/P

  (The law of large numbers)

$$m/n \approx Q/P = \pi/4$$

- pi is roughly equal to

$$\pi \approx 4m/n$$



$x_i \in [0,1) \quad y_i \in [0,1) \quad i = 1 \ldots n$

● satisfies $x_i^{\,2} + y_i^{\,2} \ge 1$

● satisfies $x_i^{\,2} + y_i^{\,2} < 1$

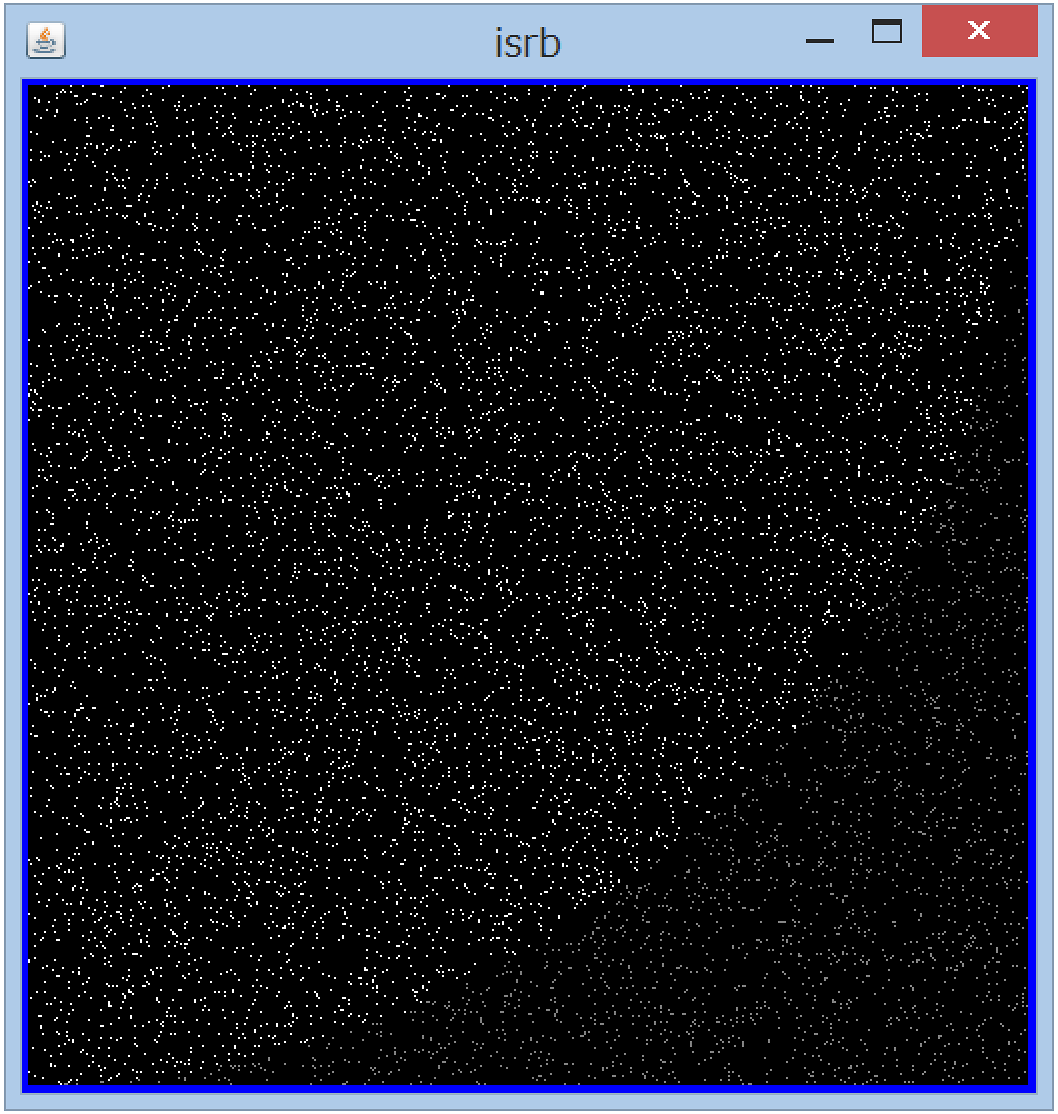
Scattering n points to approximate pi


```ruby
def montecarlo(n)
    m = 0
    for i in 1..n
        x = rand() # random number in [0,1)
        y = rand()
        if x*x + y*y < 1.0
            m = m + 1
        end
    end
    4*m*1.0/n
end
```

montecarlo.rb

Visualization: Make an image using Isrb

- mcplot(10000)



```
def mcplot(n)
  a = make2d(500,500)
  for i in 1..n
    x = rand() # random number in [0,1)
    y = rand()
    if x*x + y*y < 1.0
      a[y*500][x*500] = 1.0
    else
      a[y*500][x*500] = 0.5
    end
  end
  a
end
```

Remarks on Randomness

- Programs cannot generate a “random” number
  - can only do in a deterministic way

- Rand computes a pseudorandom number
  - Simulating random numbers in a deterministic way

Compute the n+1 th number from the last *m* numbers

$x_n, x_{n-1}, \ldots, x_{n-m+1}$ by the following recurrence formula

$$x_{n+1} = f(x_n, x_{n-1}, \ldots, x_{n-m+1})$$

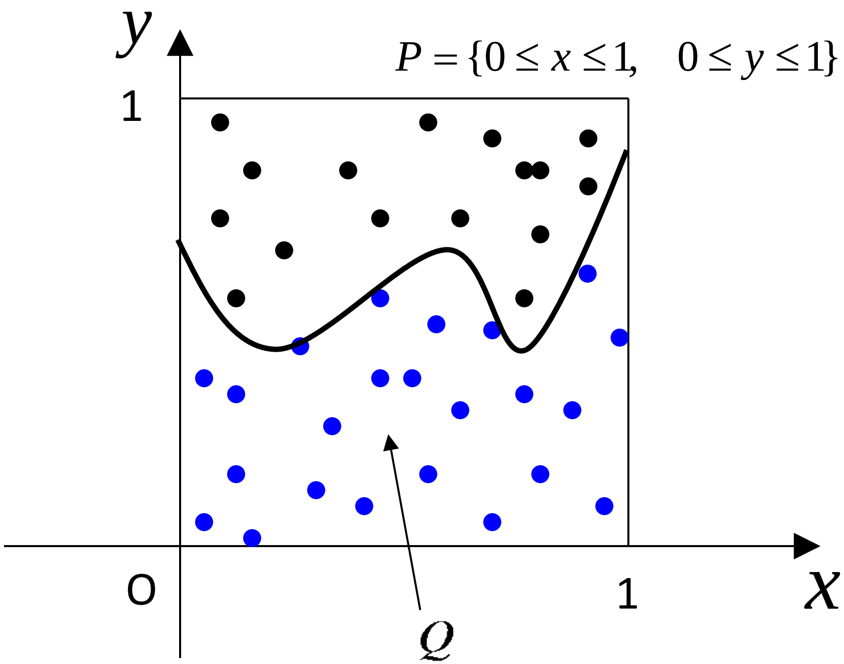
It is determined uniquely → has a cycle

Hope the length of periods is long

One More Remark

- Integral $\int_0^1 f(x)\mathrm{d}x$ can be computed using Monte-Carlo

- Ratio $m/n \approx$ (prob. that a point is in Q) = Q/P
  - = the area of the part under the curve
  - It is efficient if the curve is not explicitly given



One More Remark

- Integral $\int_0^1 f(x)\mathrm{d}x$ can be computed using Monte-Carlo
- Ratio m/n ≈ (prob. that a point is in Q) = Q/P
  - = the area of the part under the curve
  - It is efficient if the curve is not explicitly given

Rem. Trapezoidal approximation
Other methods to compute the integral
(out of the scope,
but within the April-entry course)

Approximate the area using
a set of trapezoids