MONTE CARLO METHOD FOR ESTIMATING π

- E. kenneth

1 Introduction

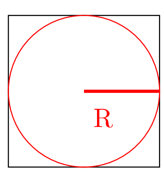
1.1 Problem: The area of a circle is defined as πr^2. Estimate π to 3 decimal places using a Monte Carlo method.

The "Monte Carlo Method" is a technique that can be used to solve mathematical or statistical problems, and a Monte Carlo simulation uses repeated sampling to obtain the statistical properties of some phenomenon (or behavior). Given the probability, P, that an event will occur in certain conditions, a computer can be used to generate those conditions repeatedly. The number of times the event occurs divided by the number of times the conditions are generated should be approximately equal to P.

$$P = \frac{number\ of\ times\ the\ event\ occurs}{number\ of\ times\ the\ conditions\ are\ generated} \tag{1}$$

1.2 Monte Carlo explained

If a **circle** of radius R is inscribed inside a **square** with side length 2R, then the area of the circle will be πr^2 and the area of the square will be $(2R)^2$. So the ratio of the area of the circle to the area of the square will be $\frac{\pi}{4}$.



$$Area\ of\ square = (2R)^2 \tag{2}$$

$$Area\ of\ circle = (\pi * R^2) \tag{3}$$

$$P_{ratio} = \frac{Area\ of\ circle}{Area\ of\ square} = \frac{\pi}{4} \tag{4}$$

for N points selected at random inside the square, approximately $\frac{N*\pi}{4}$ of those points should fall inside the circle.

$$P_{ratio} = \Pi_i^N \frac{\pi}{4} = P_{ratio} = \frac{N*\pi}{4} \qquad for\ i\ in\ 1...N \tag{5}$$

This procedure picks points at random inside the square. It then checks to see if the point is inside the circle.

If $(x_{(i)}, y_{(i)})$ are points in the square, and this point is also in the circle, then it should obey the equation of a circle such that

$$x_{(i)}^2 + y_{(i)}^2 <= 1 \qquad for\ i\ in\ 1...N \tag{6}$$

The procedure keeps track of all points within the circle (C_r) for N iterations and Finally, Pi is approximated as:

$$\pi = 4\frac{P_{ratio}}{N} \tag{7}$$

2 Algorithm implementation

ALGORITHM PROCEDURE

initialize count $\leftarrow 0$
Repeat for N iterations
  x := random()
  y := random()
  if $x^2 + y^2 \leq 1$
   count $\leftarrow +1$
  else
   continue
return $4 * \frac{count}{N}$

PYTHON IMPLEMENTATION

```
import numpy as np
count = 0
for ii in range(100):
    x, y = np.random.rand(), np.random.rand()
    if np.square(x) + np.square(y) ≤ 1:
        count += 1
        else: continue
print('{}'.format((4 * count/100)))
```

ALGORITHM PROCEDURE — Average π estimate

```
initialize count ← 0
list of pis ← []
Repeat for M number of times
  Repeat for N iterations
      x := random()
      y := random()
      if x^2 + y^2 ≤ 1
        count ← +1
      else
        continue
  return 4 * count/N
return average(list of pi's)
```

PYTHON IMPLEMENTATION — Average π

```
import numpy as np
count = 0
pi = []
for ij in range(20):
    for ii in range(100):
        x, y = np.random.rand(), np.random.rand()
        if np.square(x) + np.square(y) ≤ 1:
            count += 1
            else: continue
  print('{}'.format((4*count/100)))
print('{}'.format(np.mean(pi)))
```