Monte Carlo Methods

I, at any rate, am convinced that He does not throw dice.

Albert Einstein

Pseudo Random Numbers: 1/3

- **Random numbers are numbers occur in a "random" way,**
- **If they are generated by an algorithm, they are not actually very random. Hence, they are usually referred to as *pseudo random* numbers.**
- **In Fortran 90, two subroutines help generate random numbers: `RANDOM_SEED()` and `RANDOM_NUMBER()`.**
- **The generated random numbers are uniform because the probability to get each of these numbers is equal.**

Pseudo Random Numbers: 2/3

- **`RANDOM_SEED()` must be called, with or without actual arguments, before any use of `RANDOM_NUMBER()` or before you wish to "re-seed" the random number sequence.**
- **`RANDOM_NUMBER(x)` takes a `REAL` actual argument, which is a variable or an array element. The generated random number is returned with this argument.**
- **The generated random number is in [0,1). Scaling and translation may be needed.**

Pseudo Random Number: 3/3

- **Simulate the throwing of two dice *n* times.**
- **Array `count()` of 12 elements should be initialized to 0, and `p` and `q` are the "random" numbers representing throwing two dice.**
- **What does `INT(6*x)+1` mean?**

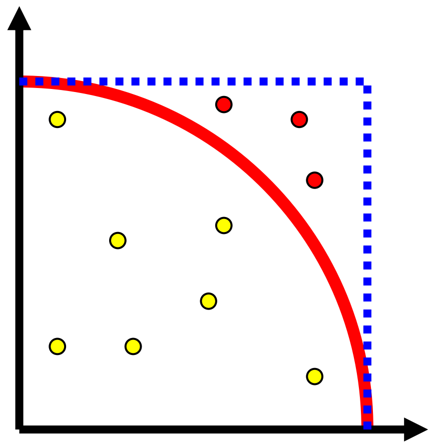
```
CALL RANDOM_SEED()
DO i = 1, n
  CALL RANDOM_NUMBER(x)
  p = INT(6*x) + 1
  CALL RANDOM_NUMBER(x)
  q = INT(6*x) + 1
  count(p+q) = count(p+q) + 1
END DO
WRITE(*,*) (count(i), i=1, 12)
```

Monte Carlo Methods

- **Monte Carlo techniques have their origin in WW2. Scientists found out problems in neutron diffusion were intractable by conventional methods and a probabilistic approach was developed.**
- **Then, it was found that this probabilistic approach could be used to solve deterministic problems. In particular, it is useful in evaluating integrals of multiple dimensions.**

Computing π: 1/3

- **The unit circle (*i.e.*, radius = 1) has an area of π.**
- **Consider the area in the first quadrant as shown below. Its area is π/4 ≈ 0.785398…**
- **If we generate *n* pairs of random numbers (*x*,*y*), and count the pairs in the circle, say *k*, the area is approximately *k*/*n*.**



Computing π: 2/3

- **In the following, `n` is the number of random number pairs to be generated, `count` counts the number of pairs in the circle, and `r` is the ratio.**
- **Hence, `r` ≈ π/4 if enough number of (x,y) pairs are generated.**

```
count = 0
CALL RANDOM_SEED
DO i = 1, n
  CALL  RANDOM_NUMBER(x)
  CALL  RANDOM_NUMBER(y)
  IF (x*x + y*y < 1.0)  count = count + 1
END DO
r = REAL(count)/n
```

Computing π: 3/3

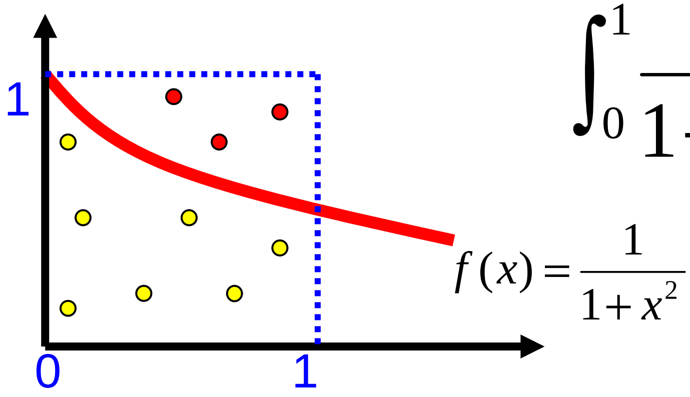
- **The following shows some results.**
- **Due to randomness, the results may be different if this program is run again.**

n	**in circle**	**ratio**
10	**9**	**0.9**
100	**72**	**0.72**
1000	**804**	**0.804**
10000	**7916**	**0.7916**
100000	**78410**	**0.7841**
1000000	**785023**	**0.7850**

ratio $\approx \pi/4$ = 0.785398…

Integration: 1/2

- **The same idea can be applied to integration.**
- **Let us integrate $1/(1+x^2)$ on [0,1]. This function is bounded by the unit square.**
- **We may generate *n* random number pairs and count the number of pairs *k* in the area to be integrated. The ratio *k*/*n* is approximately the integration.**



$$\int_0^1 \frac{1}{1+x^2}\,dx = \tan^{-1}(1) - \tan^{-1}(0) = \frac{\pi}{4}$$

Integration: 2/2

- The following shows the results.

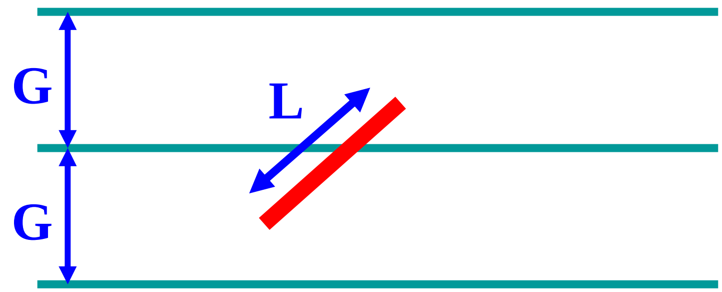
```
CALL RANDOM_SEED()
count = 0
DO i = 1, n
  CALL RANDOM_NUMBER(x)
  CALL RANDOM_NUMBER(y)
  fx = 1/(1 + x*x)
  IF (y <= fx) count=count+1
END DO
r = REAL(count)/n
```

ratio ≈ $\pi/4$ = 0.785398…

n	in area	ratio
10	9	0.9
100	77	0.77
1000	781	0.781
10000	7940	0.794
100000	78646	0.786
1000000	784546	0.785

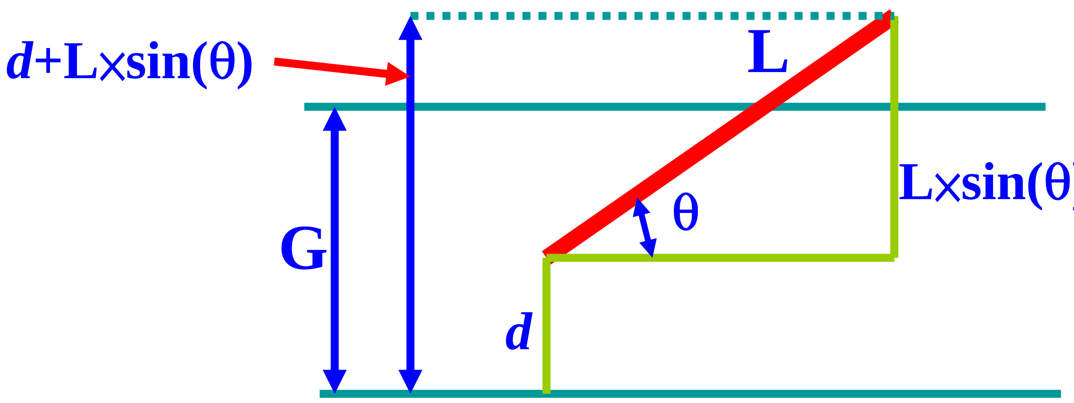
Buffon Needle Problem: 1/4

- **Suppose the floor is divided into infinite number of parallel lines with a constant gap G.**
- **If we throw a needle of length L to the floor randomly, what is the probability of the needle crossing a dividing line?**
- **This is the *Buffon needle problem*. The exact probability is $(2/\pi)\times(L/G)$.**
- **If $L = G = 1$, the probability is $2/\pi \approx 0.63661\ldots$.**



Buffon Needle Problem: 2/4

- We need two random numbers: θ for the angle between the needle and a dividing line, and d the distance from one tip of the needle to the lower dividing line.
- If $d + L \times \sin(\theta)$ is less than 0 or larger than G, the needle crosses a dividing line.



Buffon Needle Problem: 3/4

- `gap` and `length` are gap and needle length.
- The generated random number is scaled by `gap` and the angle by 2π.

```
count = 0
DO i = 1, n
  CALL RANDOM_NUMBER(x)
  distance = x*gap             ! distance in [0,gap)
  CALL RANDOM_NUMBER(angle)
  angle = angle*2*PI           ! angle in [0,2π)
  total = distance + length*sin(angle)
  IF (0 < total .AND. total < gap) count = count + 1
END DO
ratio = REAL(n-count)/n
```

Buffon Needle Problem: 4/4

- **The following has the simulated results with gap and needle length being 1.**

n	**in area**	**ratio**
10	**8**	**0.8**
100	**61**	**0.61**
1000	**631**	**0.631**
10000	**6340**	**0.634**
100000	**63607**	**0.63607**
1000000	**636847**	**0.63685**

Exact value = $2/\pi \approx 0.63661\ldots$.

The End