Computational Analysis of Physical Systems (Lecture 4)

Random numbers in Python

“random” module - 1

Function	Description
choice(seq)	A random item from a list, tuple, or string.
randrange ([start,] stop [,step])	A randomly selected element from range(start, stop, step)
random()	A random float r, such that 0 is less than or equal to r and r is less than 1
seed([x])	Sets the integer starting value used in generating random numbers. Call this function before calling any other random module function. Returns None.
shuffle(lst)	Randomizes the items of a list in place. Returns None.
uniform(x, y)	A random float r, such that x is less than or equal to r and r is less than y

“random” module - 2

random()

Description

The method **random()** returns a random float r, such that 0 is less than or equal to r and r is less than 1.

```
import random

# First random number
print "random() : ", random.random()

# Second random number
print "random() : ", random.random()
```

```
random() :  0.281954791393
random() :  0.309090465205
```

“random” module – 3

choice(seq)

Description

The method **choice()** returns a random item from a list, tuple, or string.

```
import random

print "choice([1, 2, 3, 5, 9]) : ", random.choice([1, 2, 3, 5, 9])
print "choice('A String') : ", random.choice('A String')
```

```
choice([1, 2, 3, 5, 9]) :  2
choice('A String') :  n
```

“random” module - 4

randrange ([start,] stop [,step])

Description

The method **randrange()** returns a randomly selected element from range(start, stop, step).

```
import random


# Select an even number in 100 <= number < 1000

print "randrange(100, 1000, 2) : ", random.randrange(100, 1000, 2)


# Select another number in 100 <= number < 1000

print "randrange(100, 1000, 3) : ", random.randrange(100, 1000, 3)
```

```
randrange(100, 1000, 2) :  976
randrange(100, 1000, 3) :  520
```

“random” module - 5

seed([x])

Description

The method **seed()** sets the integer starting value used in generating random numbers. Call this function before calling any other random module function.

```
import random


random.seed( 10 )

print "Random number with seed 10 : ", random.random()


# It will generate same random number

random.seed( 10 )

print "Random number with seed 10 : ", random.random()


# It will generate same random number

random.seed( 10 )

print "Random number with seed 10 : ", random.random()
```

```
Random number with seed 10 :  0.57140259469
Random number with seed 10 :  0.57140259469
Random number with seed 10 :  0.57140259469
```

“random” module - 6

shuffle(lst)

Description

The method **shuffle()** randomizes the items of a list in place.

```
import random



list = [20, 16, 10, 5];

random.shuffle(list)

print "Reshuffled list : ",  list



random.shuffle(list)

print "Reshuffled list : ",  list
```

```
Reshuffled list :  [16, 5, 10, 20]
Reshuffled list :  [16, 5, 20, 10]
```

“random” module - 7

uniform(x, y)

Description

The method **uniform()** returns a random float r, such that x is less than or equal to r and r is less than y.

```
import random

print "Random Float uniform(5, 10) : ",  random.uniform(5, 10)

print "Random Float uniform(7, 14) : ",  random.uniform(7, 14)
```

```
Random Float uniform(5, 10) :  5.52615217015
Random Float uniform(7, 14) :  12.5326369199
```

Random vectors & matrices

```
import numpy
a=numpy.random.rand(3,3)
print a
```

or

```
import numpy as np
a=np.random.rand(3,3)
print a
```

Exercise – 1-1

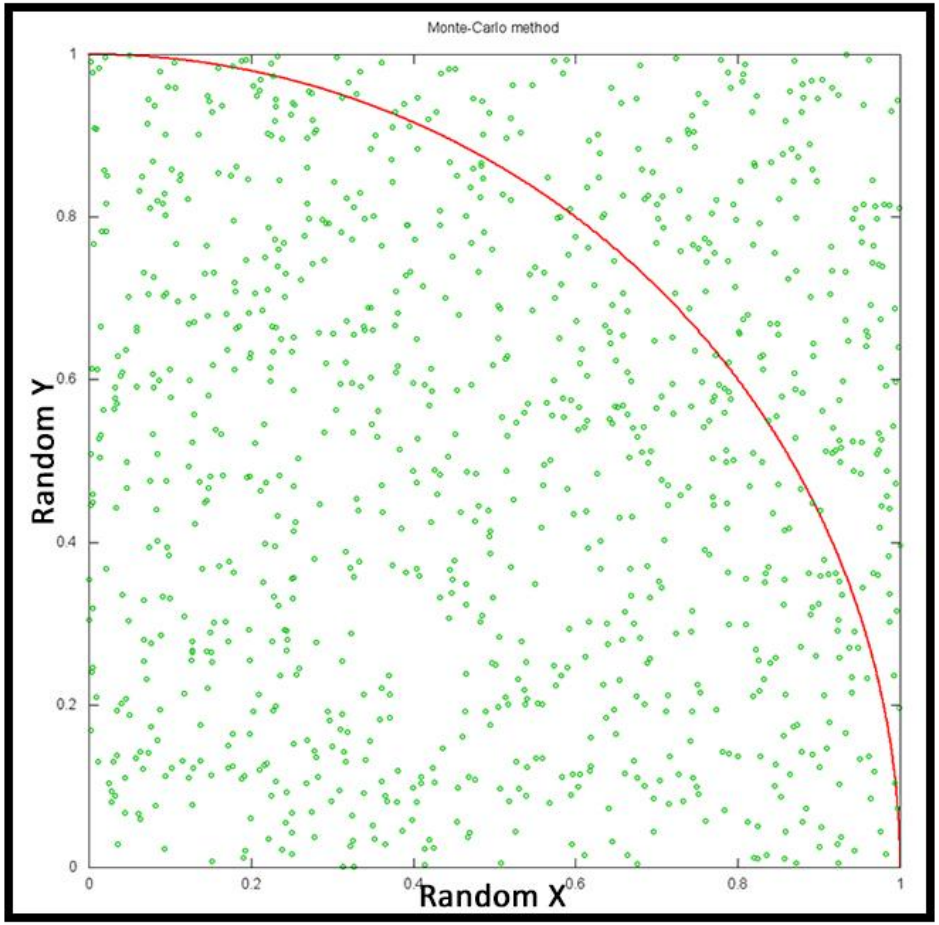
Radius of Circle = 1 Unit
Area of Quarter Circle = PI * R * R / 4
= PI * 1 * 1 / 4
= PI / 4 sq. units

Side of Square, a = 1 Unit
Area of Square = a * a
= 1 sq. unit

Probablity of Random Point, R(x,y) being inside the circle and square P=
Area of Quarter Circle / Area of Square
= PI / 4.

For N samples, M points lie within the circle and the square, and N-M points lie outside the circle but inside the square.

Using the probablitly, M for N samples
=> M = N*PI/4
=> PI = 4*M/N



Solution - 1

```
from random import random

all=input("How many points?... ")

inside=0

for i in range(all):
    x,y=random(),random()
    if (x**2+y**2)**(0.5)<1: inside=inside+1

mypi=4.0*(float(inside)/all)
print ("The value of pi for %d points is %f"%(all,mypi))
```

Solution – 1 with time measurement

```
from random import random
import time

all=input("How many points?... ")

starttime=time.clock()
inside=0

for i in range(all):
    x,y=random(),random()
    if (x**2+y**2)**(0.5)<1: inside=inside+1

mypi=4.0*(float(inside)/all)

elapsedtime=(time.clock() - starttime)
print ("The value of pi for %d points is %f and the time is %f seconds"%(all,mypi,elapsedtime))
```

The value of pi for 1000000 points is 3.142652 and the time is 1.104043 seconds

Solution - 2

```
import numpy as np
import time

all=input("How many points?... ")

starttime=time.clock()
inside=0

x=np.random.rand(all)
y=np.random.rand(all)

for i in range(all):
   if (x[i]**2+y[i]**2)**(0.5)<1: inside=inside+1

mypi=4.0*(float(inside)/all)

elapsedtime=(time.clock() - starttime)
print ("The value of pi for %d points is %f and the time is %f seconds"%(all,mypi,elapsedtime))
```

```
The value of pi for 1000000 points is 3.141996
and the time is 5.199235 seconds
```