Measurement of Queuing system Performance

The performance of a queuing system can be evaluated in terms of a number of response parameters, however the following fours are generally employed

1. Average number of customers in the queue or in the system.
2. Average waiting time of the customer in the queue or in the system.
3. System Utilization (server utilization)
4. The cost of waiting time and idle time

1. Average number of customers in the queue or in the system

If T_a and T_s be the inter arrival time and the mean service time then

Arrival rate $\lambda = \frac{1}{T_a}$

Service rate $\mu = \frac{1}{T_s}$

Average number of customer in the system $= \frac{\lambda}{(\mu - \lambda)}$

Average number of customers in the queue $= \frac{\lambda^2}{\mu(\mu - \lambda)}$

② Average waiting time in the system $= \frac{1}{(\mu-\lambda)}$

Average waiting time in the queue $= \frac{\lambda}{\mu(\mu-\lambda)}$

The ratio of the mean service time to the mean inter arrival time is called traffic intensity i.e $H = \frac{T_s}{T_a}$

→ What do you mean by server utilization?

③ Server utilization :

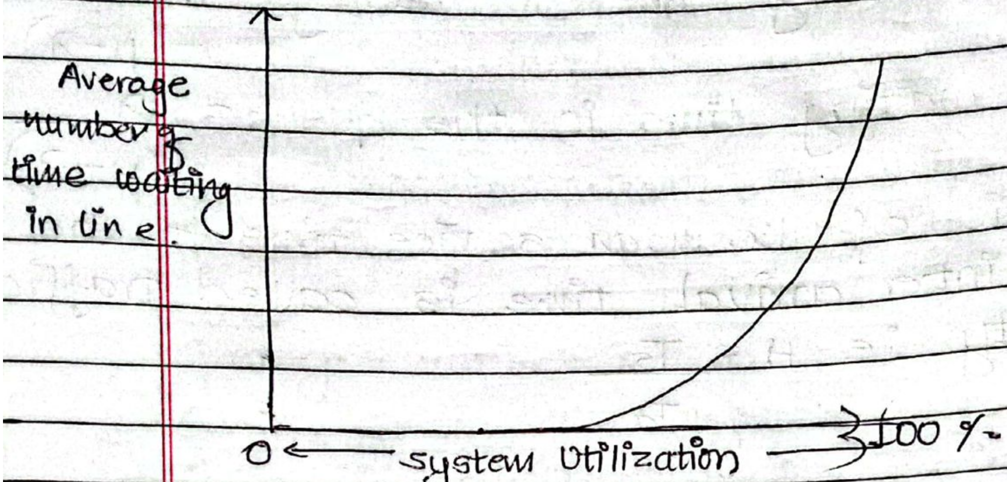
Server / system utilization is the percentage of the time that all servers are busy. System utilization factor (ρ) is the ratio of average arrival rate (λ) to the average service rate (μ).

$\rho = \frac{\lambda}{\mu}$ in the case of a single server model.

$\rho = \frac{\lambda}{\mu n}$ in the case of a "n" server model.

The system utilization can be increased by increasing the arrival rate which amounts to increasing the average queue length as well as the average waiting time as shown in fig. Under the normal circumstances 100% system utilization is not a realistic goal.

CamScanner



4) The cost of waiting time and idle time:
Probability of finding service counter free is $(1-\rho)$ i.e. there are zero customers in the server facility.

Congestion:
A congestion system is system in which there is a demand for resources for ~~a~~ system and when the resources become unavailable, those requesting the resources wait for them to become available. The level of congestion in such systems is usually measured by the waiting line or queue of resource requets (waiting line or queuing models.)

CSMP (Continuous System Modelling Program)

CSMP or Continuous System Modelling Program is an early scientific computer software designed for modelling and solving differential equations numerically. This enables real-world systems to be simulated or tested with a computer.

Types of statements in CSMP

(1) Structured Statement

(2) Data Statement

(3) Control Statement.

(1) Structural statements which defines the model. They consists of FORTRAN like, statement and functional block designed by operations that frequently occur in a model definition. Structural statement can make use of the operations of addition, substraction multiplication, division, exponentiation, using the same notation and rule as are used in FORTRAN.

If the model include the equation.

$$X = \frac{6Y}{W} + (Z-2)^2$$

Then the following statement would be used.

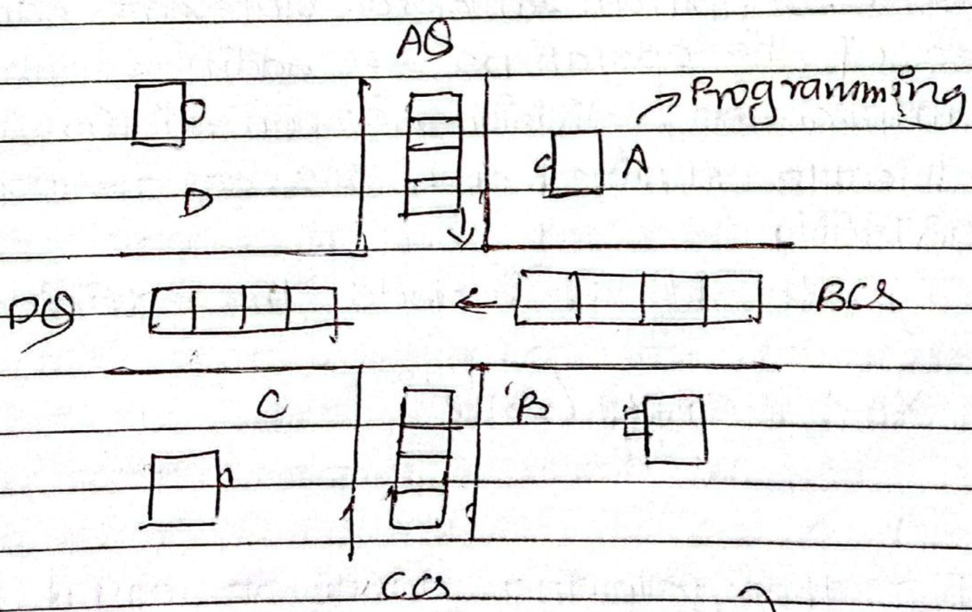
```
X=6.0*Y/W+(Z-2)**2.0
```

② Data statements which assign numerical value to parameters, constants and initial condition. For example one data statement called INCON can be used to set the initial value of integration function block.

③ Control statements which specify options in the assembly and execution of program and choices of input. For eg, if printed output in ~~remaining~~ required, control statements with PRINT and PRAEL are used followed by the names of variables to form the output.

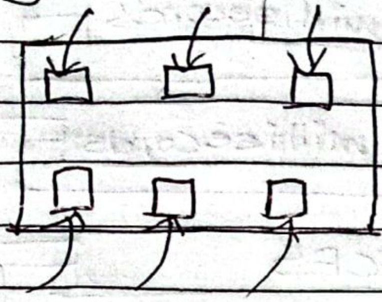
Application of queuing system.

① Simulation of traffic control system.



(A junction of four roads)
[Simulation of traffic control system]

② CPU scheduling in multiprogramming and Time sharing environment

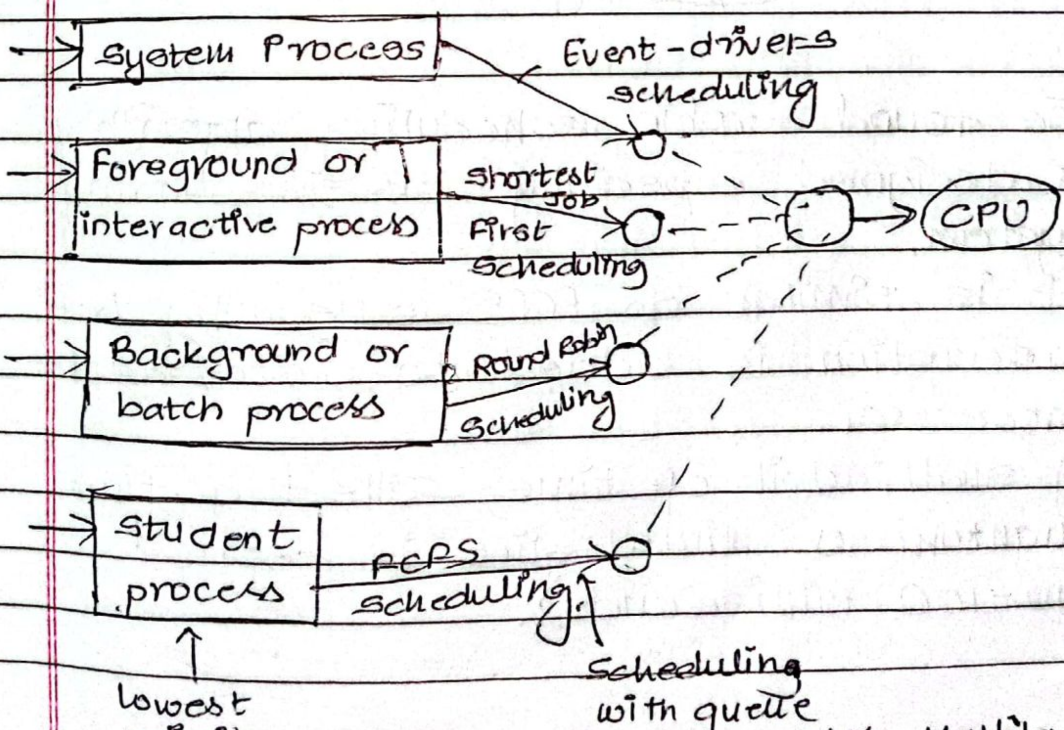


→ CPU scheduling is the basis of multipro-gramming operation systems

→ By switching the CPU among processes the OS can make the more productive

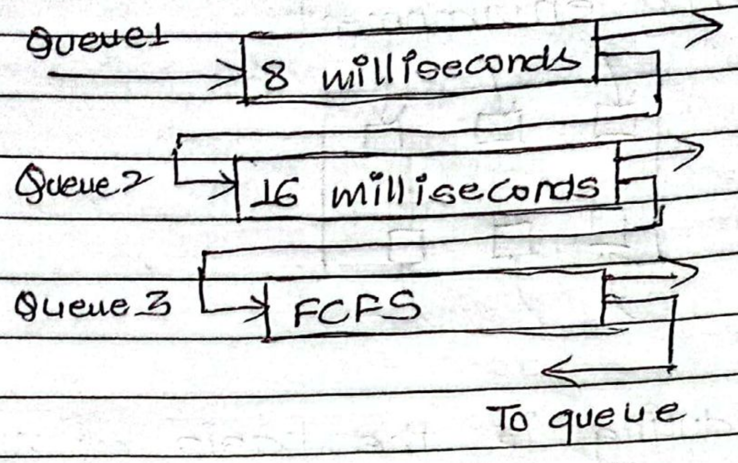
③ Multilevel queue scheduling

↳ highest priority process served first

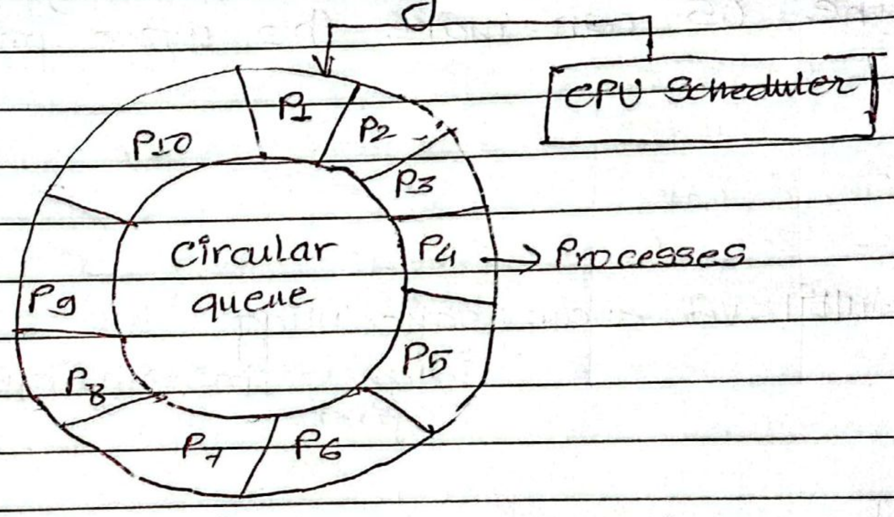


[Process scheduling with Multilevel Queues]

④ Multilevel feedback queue scheduling



⑤ Round Robin Scheduling



→ The round-robin scheduling algorithm is designed especially for timesharing system.

→ It is similar to FCFS scheduling but preemption is added to switch between processes.

→ A small unit of time, called a time quantum or time slice is defined (10-100 milliseconds).

Network of Queues

(1) Open Network
(2) Closed Network

→ Network of queues are systems in which a number of queues are connected by customer routing. When a customer is serviced at one node it can join another node and queue for service or leave the network.

(1) Open Network

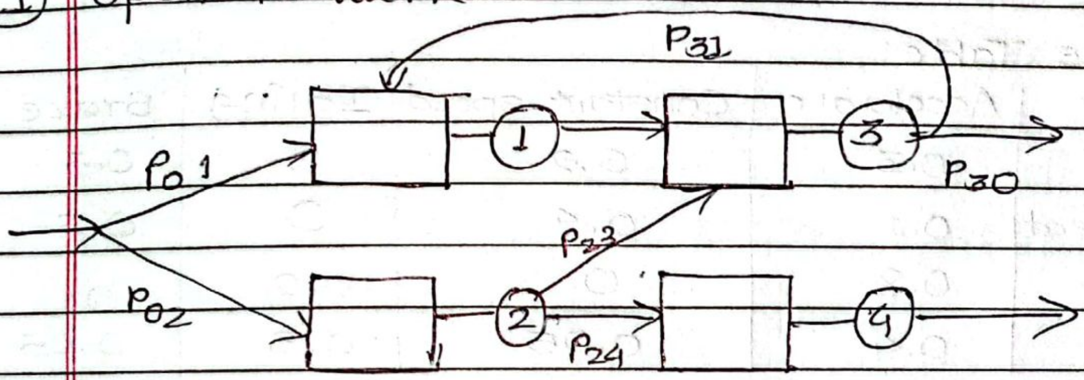


Fig: Open Network

(2) Closed Network

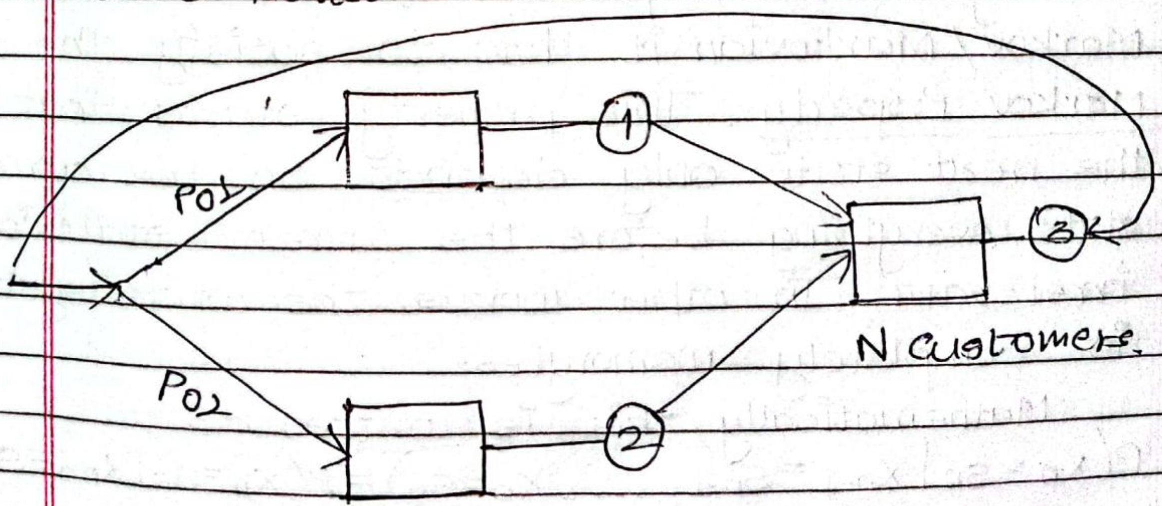
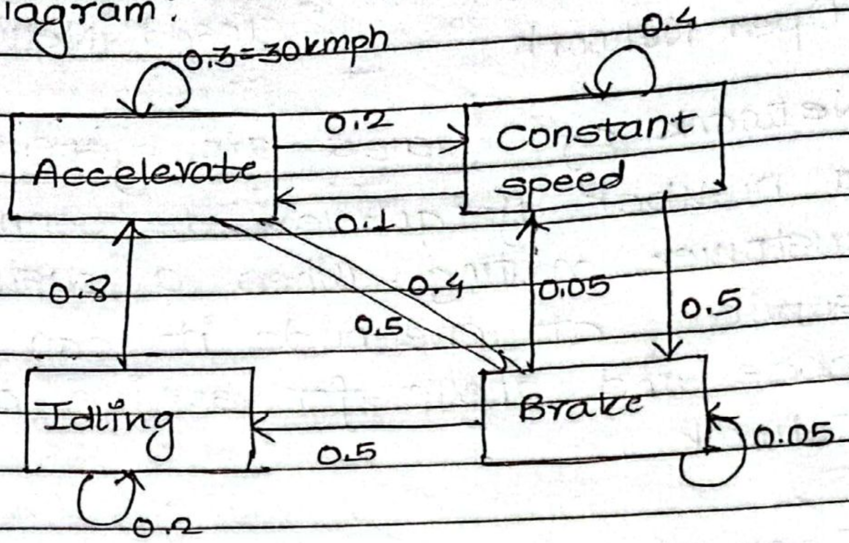


Fig: Closed Network.

eg:

State Diagram:



State Table

	Accelerate	Constant speed	Idling	Brake
Accelerate	0.3	0.2	0	0.5
Constant speed	0.1	0.4	0	0.5
Idling	0.8	0	0.2	0
Brake	0.4	0.05	0.5	0.05

Markov Property

For any modelling process to be considered Markov/Markovian it has to satisfy the Markov Property. This property states that the next state only depends on the current state, everything before the current state is irrelevant. In other words, the whole system is completely memoryless.

Mathematically, this is written as

$$P(X_n = S_n \mid X_{n-1} = S_{n-1}, \ldots X_0 = S_0) = P(X_n = S_n \mid X_{n-1} = S_{n-1})$$

where n is the time step parameter and X is a random variable that takes on a value in a given space S. The state space refers to all the possible outcome of an event.

For examples, a coin flip has two values in its state space :S = {Head, Tails} and the probability of transitioning from one state to the other is 0.5.

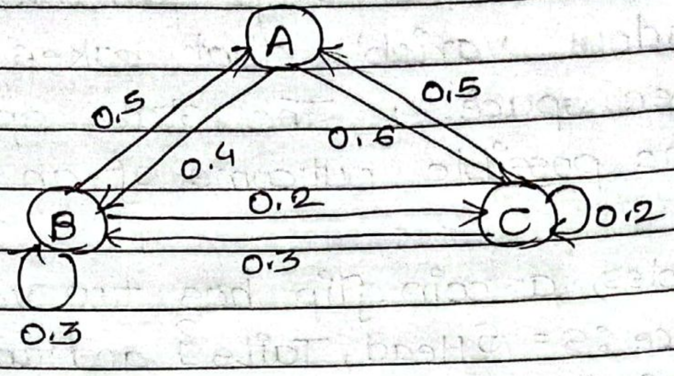
A process that uses the Markov Property is known as a Markov Process. If the state space is finite and we use discrete, time-steps this process is known as a Markov chain. In other words, it is a sequence of random variables that takes on state in the given state space.

Consider :→

→ Time-homogeneous discrete-time Markov chains as they are the easiest to work with.

→ Time-inhomogeneous Markov chains where the transition probability between state is not fixed & varies with time.

Fig below shows an examine Markov chain with state space ({A, B, C}) The numbers on the arrow indicate the probabitylity of transitioning between those two states.



For example, if we want to go from state B to C, then this transition has a 20% chance.

UNIT-5

Random Number

→ True Random Number → 4.1, 6.4, 5.6,

→ Pseudo Random Number

Q. Differentiate between true and pseudo Random Numbers. What are the basic properties of random numbers?

→ Pseudo Random numbers:

Pseudo Random numbers are the ~~com~~ random numbers that are generated by using some known methods (algorithms) so as to produce a sequence of numbers in [0,1] that can simulate the ideal properties of random numbers. They are not completely random as the set of random numbers can be replicated because of use of some known methods.

→ True Random numbers are gained from physical processes like radioactive decay or also rolling a dice and introduce it into a computer.

→ Pseudo Random numbers have fact response in generating numbers while true random number have slow response.

→ In pseudo random numbers, sequence of number can be reproduced whereas in True random numbers, sequence of numbers can't be reproduced.

CamScanner

→ In pseudo random numbers, sequence of number is repeated whereas in true random number, sequence of numbers will or will not be repeated

Properties of Random Number

1. Uniformity
2. Independence
3. The maximum Density
4. Maximum cycle

Use of Random Numbers

1) Science & Technology
2) Art
3) Gaming
4) Gambling
5) Cryptography — Random [AES, DES], Security
6) Various other field

Techniques to generate pseudo Random Numbers

1) Mid-Square Algorithm
2) Linear Congruential Generator
3) Implicit inversion Congruential
4) Shift Register
5) Additive lagged-Fibonacci
6) Combined
7) Multiplicative lagged-Fibonacci

Pseudo Random Number Generation (PRNG)

Methods of generation of Pseudo Random numbers

1) Mid-square method or Algorithm
2) Linear Congruential Generator (LCG)

1) Mid-square method :

It was given by John Von Neumann and Metropolis in 1940.

Algorithm

→ Take any integer (seed) of say a two digit numbers.

→ Square it and pick the middle two digits and discard the rest. Next seed

→ Now again square the selected middle two digits in the previous step.

→ Repeat

S.No	Random Number	Square	Middle digits
1 →	11 (seed)	→ 0121	→ 12
2 →	12	→ 0144	→ 14
3 →	14	→ 0196	→ 19
4 →	19	→ 0361	→ 36
5 →	36	→ 1296	→ 29
6 →	29	→ 0841	→ 84
7 →	84	→ 7056	→ 05
8 →	05	→ 0025	→ 02
9 →	02	→ 0004	→ 00
10 →	00	→ 0000	→ 00

Drawback

1) It terminates with a 0 but that is not the real case.

Example - 2

① Seed → 7184

2) Linear Congruential Generator (LCE)

Algorithm:

1) We can generate random numbers from random integers X_i of the LCM by

$$X_{i+1} = (aX_i + c) \bmod m.$$

2) Convert the integer X_i to random number:

$$R_i = \frac{X_i}{m}, \quad i = 1, 2, \ldots$$

$[0,1]$

Example 1

Use $X_0 = 0$ (→ Seed); $a = 5$, $c = 3$ & $m = 7$

The X_i and R_i values are:

$X_1 = (X_0 \cdot a + c) \bmod m$

$= (0 \times 5 + 3) \bmod 7$

$= (3)(\bmod 7)$

$= 3$

$R_1 = \frac{X_1}{m} = \frac{3}{7}$

$= 0.428571$

$X_2 = (3 \times 5 + 3) \bmod 7$

$= (18) \bmod 7$

$= 4$

$R_2 = \frac{X_2}{m} = \frac{4}{7}$

$= 0.571428$

$X_3 = (4 \times 5 + 3) \bmod 7$
$= 23 (\bmod 7)$
$= 2$ → $R_3 = \frac{2}{7} = 0.28571$

$X_4 = (2 \times 5 + 3) \bmod 7$
$= 13 (\bmod 7)$
$= 6$ → $R_4 = \frac{6}{7} = 0.857142$

$X_5 = (6 \times 5 + 3) \bmod 7$
$= 33 (\bmod 7)$
$= 5$ → $R_5 = \frac{5}{7} = 0.714288$

$X_6 = (5 \times 5 + 3) \bmod 7$
$= 28 (\bmod 7)$
$= 0$ → $R_6 = 0/7 = 0$

cycle length = 6

Example 2
Use $X_0 = 27$, $a = 17$, $c = 43$ and $m = 100$. Generate Random number using linear congruential method (LCM).

Test for Random Number

The desired properties of random number ① Uniformity and ② Independence can be checked or achieved by number of tests that can be performed on generated random number.

The test can be placed into two categories according to the ideal properties of randomness.

Random Number Tests

- Uniformity Test
  - Frequency Test
    - ① Kolomogorov Simorov (KS) Test
    - ② Chi-Square Test
- Independence
  - Autocorrelation Test
    - Test the correlation between numbers & compare the sample ~~correlation to the~~ expected correlation zero.

These two tests are used to compare the distribution of set of numbers generated to a uniform distribution

↳ Uniformity Test (Frequency test)

It is a basic test that should always be performed to validate a new generator for uniformity distribution of random integers, numbers.

Two different testing are available.

① KS Test (Kolmogorov Test)

② Chi-square Test

Both of these tests measure the degree of agreement between the distribution of samples of generated random numbers and the theoretical uniform distribution.

Both these test are based on the null hypothesis of no significant differences between the sample distribution and theoretical distribution.

(1) Kolmogorov - Smirov (KS) Test :
This test compares the continuous cdf, $F(x)$ of uniform distribution with empirical cdf, $S_N(x)$ of the sample of 'N' observation

Algorithm (KS Test)

Step 1: Rank the data from smallest to largest. Let $R_{(i)}$ denote the i^{th} smallest observation so that

$$\boxed{R_{(1)} \leq R_2 \leq R_3 \leq \ldots\ldots \leq R_{(N)}}$$

Step 2: Compare

$$D^+ = \max_{1 \leq i \leq N} \left\{ \frac{i}{N} - R_{(i)} \right\}$$

D^+ is largest deviation above $S_N(x)$ above $f(x)$.

$$\boxed{\frac{i}{N} = S_N(x) = \frac{\text{No. of } R_1, R_2, \ldots\ldots \leq x}{N}}$$

$$D^- = \max_{1 \leq i \leq N} \left\{ R_i - \frac{N_i - 1}{N} \right\}$$

D^- is largest deviation of $S_N(x)$ below $F(x)$

Step 3: Compute $D = max(D^+, D^-)$
where 'D' is largest standard deviation between uniform distribution and empirical distribution

Step 4: Locate critical value D_α for specified level of significance 'α' and given sample size 'N'.

Step 5: If sample statistical 'D' is greater than critical value D_α then the null hypothesis for a sample data is rejected.

Chi-Square Test

Q. Soln:
$N = 100$
Total number of intervals $= \frac{100}{10} = 10$

So, Interval, 1 - 10
11 - 20
21 - 30
⋮
91 - 100

Q. Problem → Assume 100 no. are distributed between [0,1] and the no. of samples observed in each interval is given in table below. 10 interval of equal length.
$\alpha = 0.05$

10	9	5	6	16	13	10	7	10	14

Test for uniformity distributed

Intervals	Upper Limit	O_i	E_i	$O_i - E_i$	$(O_i - E_i)^2$	$\frac{(O_i - E_i)^2}{E_i}$
1	0.1	10	10	0	0	0
2	0.2	9	10	-1	1	0.1
3	0.3	5	10	-5	25	2.5
4	0.4	6	10	-4	16	1.6
5	0.5	16	10	6	36	3.6
6	0.6	13	10	3	9	0.9
7	0.7	10	10	0	0	0
8	0.8	7	10	-3	9	0.9
9	0.9	10	10	0	0	0
10	1.0	14	10	4	16	1.6
	$X_0^2 = \sum_{i=1}^{n} \frac{(O_i - E_i)^2}{E_i} = 11.2$	100	100	0		11.2

Example with with 100 numbers from [0, 1]

$\alpha = 0.05$ 10 intervals

$X^2_{0.05, 9} = 16.9 \rightarrow$ obtained from tab.

$\rightarrow X^2_{0.05, 9} = 11.2 < 16$

Accept since $X_0^2 = 11.2 < 16.9$

So, 100 no. are uniformly distributed.

Gap Test

Consider the following sequence of 120 digit

2 3 6 5 6 0 0 1 3 4 5 6 7 9 4 3 1

0 3 . . . . 0 5 7 8 0

Gaplen.	Freq	Rel Freq	Accum Rel freq $S_N(x)$	$F(x)$ $1-(0.9)^{x+1}$	$\lvert F(x)-S_N(x) \rvert$
0-3	34	309	309	3439	

$$D = Max\left(F(x) - S_N(x)\right)$$
$$= 0.034$$
$$D_{0.05} = \frac{1.36}{\sqrt{N}} = \frac{1.36}{\sqrt{110}} = 0.129$$

$$D = 0.0349 < D_{0.05} = 0.113$$

Ho is not rejected. They are independent

Random variate Generation

1. Inverse Transform Technique
2. Acceptance & Rejection Technique

Flowchart