Chapter 1

CONCEPTS OF SIMULATION

1.1 Introduction

Simulation is the representation of a real life system by another system, which depicts the important characteristics of the real system and allows experimentation on it. In other words, simulation is an imitation of the reality. Though the formal use of the simulation technique is not very old, simulation has long been used by the researchers, analysts, designers and other professionals in the physical and non-physical experimentations and investigations. In our day-to-day life, we use simulation, even without realizing it. Globe imitates some important geographical characteristics of the earth. Scale models of various machines and plants are used in laboratories to study their performance characteristics. Simple models of machines are used to simulate the plant layouts. A model aeroplane suspended in a wind tunnel simulates a real sized plane moving through the atmosphere, and is used to study the aerodynamic characteristics. A children cycling park, with various crossings and signals is a simulated model of the city traffic system. A planetarium represents a beautiful simulation of the planetary system. Environments in a geological park and in a museum of natural history are other examples of simulation.

In each of these examples, the real system has been substituted by physical model, which is not possible in case of complex and intricate problems of managerial decision making. The inventory management system, complicated waiting lines, manufacturing system etc., cannot be imitated physically. In such cases, a series of mathematical and logical statements are used to represent the system. Simulation of this type involves a huge amount of computations, which are possible only with the help of a powerful computing system, and hence the name computer simulation or system simulation.

It can be said that system simulation is mimicking of the operation of a real system, such as the day-to-day operation of a bank, or the running of an assembly or production line in a factory, or assignment of salesman to different sales areas. A simulation is the execution of a computer model of the system that is a computer program, to get information about the system. It is like conducting an experiment on the system, and as compared to purely theoretical analysis, simulation is much flexible and convenient. The readily available simulation softwares has made it possible for managers who are not computer programmers or expert analysts, to model and analyze the operation of real systems. Before going into further details, it will be appropriate to discuss the concept of system.

1.2 The System

The term 'System' is a word of everyday use. It is used in a large variety of situations and in such a variety of ways, that it has become very difficult to give a definition sufficiently broad to cover the variety of uses, and concise enough to serve the purpose. There are many definitions found in literature, but none can universally be applied. Klir* gives a collection of 24 definitions and one such definition is, "A system is a collection of components wherein individual components are constrained by connecting interrelationships such that the system as a whole fulfills some specific functions in response to varying demands".

---

* Klir, George J., *An Approach to General Systems Theory*, New York : Van Nostrand Reinhold Co., 1969

In simple, the system has also been defined as "an aggregation or assemblage of objects joined in some regular interaction or interdependency."

(Gorden Geoffrey, *System Simulation*, Prentice Hall of India, New Delhi, 1980)

These definitions seem to be more relevant to the physical and static systems comprising of components or objects, connected by some physical laws. A more comprehensive definition must include the dynamic effects, where the interactions cause changes over time.

A simple example of system can be taken of an engine governor, which is required to keep the speed constant within limits, at varying loads on the engine (Fig. 1.1). As the load is changed, speed changes, governor balls lift or lower the sleeve, which controls the fuel supply, and in turn the speed. It is an automatic control system.

A manufacturing system comprises of a number of departments such as production control department, purchase department, fabrication, assembly, finishing, packaging, inspection and quality control, shipping and personnel department etc. All of these departments are interlinked to maintain

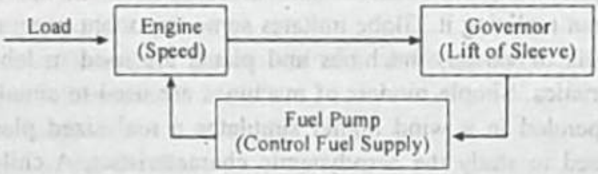


Fig. 1.1

the proper flow of the materials and information. These components of the manufacturing system are individually very complicated and the system as a whole becomes very complex.

In case of a governor system, the components of the system, the engine, governor, fuel pump etc., are all physical. Their interrelationships are based on well-known physical laws. Such systems are called physical systems.

The manufacturing system, on the other hand comprises of a large number of departments, with man made interrelationships, which cannot be represented by physical objects. Such systems are classified as non-physical systems. Management systems, social system, education system, political system etc. are all non-physical systems.

The behavior of one component of a system affects the performance of other components due to interconnections and hence, when a system is identified it has to be described and studied as a whole. This methodology of tackling the system is called *systems approach*. We may also call it the scientific approach. The term *system engineering* has been used for engineering employing systems approach.

The performance of a system is often affected by the activities occurring outside the system. The availability of raw materials, power supply, demand of goods etc., are activities, which will affect the performance of the manufacturing system. Similarly some activities may also produce changes that may not affect the system at all. All these changes, which occur outside the system, constitute the system environment. Thus, while describing a system, it is an important step to draw a boundary between the system and its environment. How and where the boundary is to be drawn depends upon the purpose of modeling the system. For example, if in a manufacturing system the effect of a particular work place lay out is to be studied, then activities like inspection, purchasing, packaging, inprocess inventory level etc., are all external to the system. On the other hand if the idle time of bank teller is to be studied, the arrival of customers is an activity internal to the system. Activities, which occur within the system, are called *endogenous activities*, while those occurring in environment are called *exogenous activities*. A system for which there is no exogenous activities, that is the system that is not affected by its environment, is said to be *closed system*, while a system, which is affected by the activities occurring outside its boundary, is called an *open system*.

In each system, there are some distinct components, which are of interest in a particular investigation. While the term *entity* is used to denote the component of interest, the property of interest of the entity is called its *attribute*. There can be many attributes to an entity. In a traffic system vehicles say buses, trucks and cars may be the entities while speed, distance moved, noise produced, exhaust emission, number of accidents etc., may be the attributes of each entity. Which of these are of interest will depend upon the purpose of investigation.

The process, which causes change in any attribute of an entity of a system, is called an *activity*. In case of traffic system, driving may be an activity. The term, state of the system at a possible time, is the description of all entities, attributes and activities, as they exist at that particular point in time.

Some other terms, which are frequently encountered while dealing with a system, are given below.

Queue: It is a situation, where entities wait for some thing to happen. It may be physical queue of people, or of objects, or a list of tasks to be performed.

Creating: Creating is causing an arrival of a new entity to the system at some point in time.

Scheduling: Scheduling is the process of assigning events to the existing entities, like the service begin and service and times.

Random variable: It is a variable with uncertain magnitude, *i.e.*, whose quantitative value changes at random. The inter arrival times of vehicles arriving at a petrol pump, or the life of electric bulbs are random variables.

Random variate: Random variate is a variable generated by using random numbers along with its probability density function.

Distribution: It is a mathematical law which governs and defines probabilistic features of the random variables. It is also called probability density function.

Let us take the example of a petrol pump, where vehicles arrive at random for getting petrol or diesel. In this case,

Entities are vehicles which will comprise of various types of four, three and two wheelers.

Events are arrival of vehicles, service beginning, service ending, departure of vehicles.

States: or state variables are the number of vehicles in the system at a particular time, number of vehicles waiting, number of workers busy, etc.

Attributes: Type of vehicle–four wheeler, two wheeler, petrol or diesel, size of fuel tank, filling rate of machine, etc.

Queue: Vehicles waiting in front of the pump.

Random variables: Inter arrival times of vehicles, service times and billing times, etc.

Distribution: The distribution may be one of the many statistical probability density functions. It is generally assumed to be exponential in case of queuing systems.

Table 1.1 lists the examples of entities, attributes, activities, events and state variables for a few systems. This table does not show a complete list of all entities, attributes, activities, events or states of the system, as the complete list can be made only when the system is properly defined.

Table 1.1

System	*Entities*	*Attributes*	*Activities*	*Events*	*State variables*
Banking	Customers	Balance, Credit status	Making deposits, withdrawls	Arrivals, departures	Number of customers waiting, number of busy tellers
Traffic control lights	Vehicles	Distance, speed, type of vehicle	Driving	Stopping, starting	Status of traffic signals, number waiting, time to green
Production	Machines, Work pieces	Processing rates, breakdown times	Machining, welding, sampling, moving of work pieces	Work arrives at machine, processing starts, ends	Machine busy, Work piece waiting, machine down
Super market	Customers, trolleys, baskets	Shopping list	Collecting items, checking out	Arrival in store, collect basket, end shopping	Availability of stock, variety, number of shoppers waiting for check out
Communication	Messages	Length, priority, destination	Transmitting	Sending time, arrival at destination	Massages waiting to be transmitted

1.3 Continuous and Discrete Systems

From the viewpoint of simulation, the systems are usually classified into two categories,

- Continuous systems
- Discrete systems

System in which the state of the system changes continuously with time are called continuous systems while the systems in which the state changes abruptly at discrete points in time are called discrete systems. The pure pursuit problem represents a continuous system since the state variables, the locations of target and pursuer, varies continuously with time. The inventory systems discussed in Chapter 7 and the queuing systems discussed in Chapter 6 are examples of discrete systems. In case of inventory system, the demand of items as well as the replenishment of the stock occur at discrete points in time and also in discrete numbers. Similarly in case of queuing systems the customers arrive and leave the system at discrete points in time. Generally the systems in which the relationships can be expressed by mathematical expressions as in engineering and physical sciences, turn out to be continuous systems, while the systems encountered in operations research and management sciences are generally discrete systems. In continuous systems, the variables are by and large deterministic while discrete systems generally deal with stochastic situations. Few systems are wholly continuous or discrete. In reservoir problem sudden heavy rains or sudden heavy release may result into sudden changes in the state of the system and may call for discrete treatment. In a factory system though the start and finish of a machine are discrete points but the machining process is continuous. Thus, no specific rules can be laid for the development of simulation models. However, effort should be made to reach a judicious balance between the simplicity of the model and the desired level of detail and accuracy of results.

1.4 Types of Models

The models used for the analysis of various types of systems have been classified in a number of ways, as shown in Fig. 1.2.

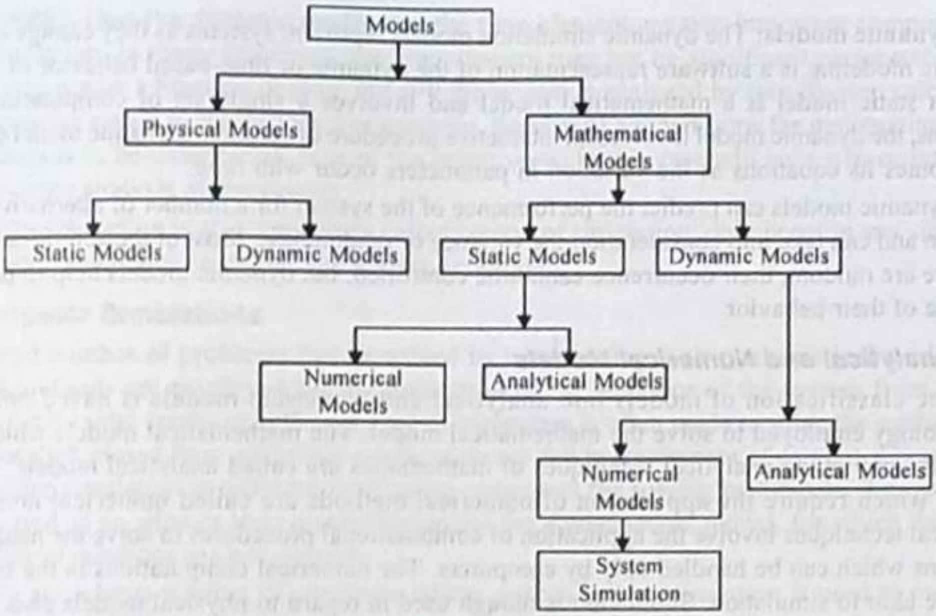


Fig. 1.2 Types of models

1.4.1 *Physical and Mathematical Models*

The first major classification of models is into physical models and mathematical models.

Physical models: In physical models, physical objects are substituted for real things. Physical models are based on some analogy between the model and simulated system. These models may comprise of only physical objects without any dynamic interaction between them, or physical objects where there is dynamic interaction between the components or the attributes of the system. Representation of mechanical systems by electrical systems, of hydraulic system by electrical systems, and vice-versa are examples of physical models. Here, the mechanical attributes like pressure, speed and load etc. are represented by properties like voltage, current and resistance etc. It can be said that in physical models, a physical representation is prepared for studying the system.

Mathematical models: The mathematical models comprise of symbolic notations and mathematical equations to represent the system. The system attributes are represented by the variables like the controlled and dependent variables, while the activities or the interaction between the variables are represented by mathematical functions. The mathematical models can be static as well as dynamic and further analytical, numerical or simulation models, depending upon the techniques employed to find its solution.

1.4.2 *Static and Dynamic Models*

Static models: A static model represents a system, which does not change with time or represents the system at a particular point in time. Static models describe a system mathematically, in terms of equations, where the potential effect of each alternative is ascertained by a single computation of the equation. The variables used in the computations are averages. The performance of the system is determined by summing the individual effects.

Since the static models ignore the time dependent variables, these cannot be used to determine the influence of changes which occur due to the variations in the parameters of the system. The static models, do not take into consideration the synergy of the components of the system, where the action of separate elements can have a different effect on the modeled system then the sum of their individual effect could indicate.

Dynamic models: The dynamic simulation models represent systems as they change over time. Dynamic modeling is a software representation of the dynamic or time-based behavior of a system. While a static model is a mathematical model and involves a single set of computations of the equations, the dynamic model involves an interactive procedure of solution. Dynamic model constantly re-computes its equations as the variation in parameters occur with time.

Dynamic models can predict the performance of the system for a number of alternative courses of action and can take into consideration the variance or randomness. Most of the activities occurring in nature are random, their occurrence cannot be controlled, but dynamic models help to predict the outcome of their behavior.

1.4.3 *Analytical and Numerical Models*

The classification of models into analytical and numerical models is based only on the methodology employed to solve the mathematical model. The mathematical models which can be solved by employing analytical techniques of mathematics are called analytical models, while the models which require the application of numerical methods are called numerical models. The numerical techniques involve the application of computational procedures to solve the mathematical equations which can be handled only by computers. The numerical computations in the interactive form are akin to simulation. Simulation is though used in regard to physical models also, but most of the applications have been found in mathematical modeling and further for modeling the dynamic and stochastic behavior of the systems.

1.4.4 *Deterministic and Stochastic Models*

Deterministic models: The deterministic models have a known set of inputs, which result into a unique set of outputs. In case of pure pursuit problem discussed in section 3-2, the initial position of the target and pursuer, their speeds and the path of flight are known and can be expressed by mathematical equations. For a given set of inputs there can be only one output. This model is deterministic. Occurrence of events is certain and not dependent on chance.

Stochastic models: In stochastic simulation models, there are one or more random input variables, which lead to random outputs. The outputs in such a case are only estimates of true characteristics of the system. The waiting line system, where the arrival of customers is random is a stochastic system. The working of a production line, where the operation times at different work station are random is a stochastic process. The random variables in a stochastic system may be represented by a continuous probability function or these may be totally discrete. Depending upon the distribution of random variables, stochastic models can be further divided into continuous models and discrete models. Because of the randomness of the variables, the results of stochastic simulations are always approximate. To obtain reliable results, a stochastic simulation model has to be run for a sufficiently long duration

1.5 System Simulation

In the introduction to the subject, a variety of examples have been given to explain the concept of simulation. In that context, forming of a physical model and experimenting on it is simulation. Developing a mathematical model and deriving information by analytical means is simulation. Where analytical methods are not applicable numerical methods or specific algorithms are applied to analyze the mathematical models, which again is simulation. These models, physical as well as mathematical can be of a variety of types. Thus the term simulation described as a procedure of establishing a model and deriving solution from it covers the whole gamut of physical, analogue, analytical and numerical investigations. To distinguish this interpretation of simulation from more general use of the technique the term system simulation is used. Geoffrey Gorden has defined system simulation as "the technique of solving problems by observation of the performance, over time, of a dynamic model of the system." Thus the dynamic model and the time element are two important components of system simulation. In many situations, the time element may not be significant parameter. System behavior may not be a function of time, but still the system is analyzed by step-by-step calculations of the successive stages of the system. For example, the size of a repair crew for maintaining a fixed fleet of buses is to be determined. Size of the repair crew, in this case will be a step or interval in the step-by-step analysis of the system.

System simulation is also sometimes called computer simulation, only because any simulation worth the name can only be carried out within a computer.

1.6 Computer Simulations

A good number of problems can be solved by constructing their mathematical models. The analytical methods are employed for the prediction of the behavior of the system from a set of parameters and initial conditions. Though, use of computers is often made to solve the mathematical models, but their closed form analytical solutions do not fall in the domain of computer simulations. The computer simulation is different from using computers for solving mathematical models. It is generally used as an adjunct to or a substitution for the mathematical model, for which the closed form analytical solutions are not possible.

There are a large number of situations where *discrete event simulation* is used to model the system of interest, like in manufacturing, commerce, defence, health etc. The complexities of such situations cannot be handled by any mathematical model. The IF and WHAT type models has to be constructed to study the influence of selected parameters on the performance of the system, *Theory of constraints*, *bottlenecks* and *management consulting* are other identical situations, which are too complex to be handled without the computer simulation.

Several software packages exist for running computer simulations or computer-based simulations. Computer simulations are also referred to as *human out of loop* simulations. These have become an important tool of modeling many natural systems in economics and social sciences as well as in engineering.

An interesting application of computer simulation is to simulate the computers using computers. The related software is called *architecture computer simulator*, which can be further divided into *instruction set simulator or full system simulator*.

The term "*synthetic environment*" is also being used these days for simulations of many kinds. This term has been adapted to broaden the definition of simulations to encompass virtuality and computer-based representation.

1.7 Physical and Interactive Simulation

The physical simulation has already been discussed in sections 1.4.1, where some examples of physical models, were also given. It can be said that ***physical simulation*** refers to simulations in which physical objects are substituted for real things. These physical objects are often chosen because of their size, cost and convenience of use. These physical simulation models are always cheap to construct and safe to use. The model of an aeroplane wing placed in wind tunnel is a physical model.

Interactive simulations: As the name suggests interactive simulation is a special kind of physical simulation in which the human operator interacts with the physical model. The flight simulator and an automobile driving simulator are examples of interactive simulations. The interactive physical simulation is also referred to as a *human in the loop simulation*. Human is an essential part of the interactive simulation.

1.8 Real-Time Simulation

In many situations, the simulation of the complete system is either highly complicated or is not desirable form the application point of view. An actual physical part of the system is integrated with a computer, which simulates the parts of the system that do not exist or that cannot conveniently be used in the experiment. Such systems usually involve interaction with a human being thereby avoiding the need to design and validate a model of human behavior. The aircraft cockpit simulator for the training of pilots and the simulated zero gravity chambers for the training of astronauts are examples of real-time simulators. These systems are called real time because the time taken by the experiment will be real that is the same as in case of a completely physical system. The real time simulation requires computers that can operate in real time; this means that they must be able to respond immediately to signals sent from physical devices, and send out signals at specific points in time.

1.9 Comparison of Simulation and Analytical Methods

In contrast to analytical models, the simulations are "run" rather than solved. Running a simulation is like conducting an experiment and an experiment always gives specific solutions depending upon the input conditions and characteristics of the system, while the analytical methods give general solutions. For example in an optimization problem, the mathematical model will give optimum value in a single solution and in case of simulation model a number of simulations will have to be executed, each resulting in a different value, one of which will approximate the optimum value.

The results obtained by simulation are approximate while the results obtained by analytical methods are exact. To increase the reliability of simulation results, simulation has to be run for longer periods. However in case of complex situation, mathematical modeling becomes difficult and many assumptions and simplifications have to be made for constructing the model. In such cases, the analytical model may give highly approximate or unrealistic results. It is a question of judgement up to what level the system has to be approximated or abstracted to fit a mathematical model.

The accuracy of simulation results depends upon the level of details at which the mode has been developed. More detailed is the model, more complex is its construction. More detailed simulation mode requires greater time and effort to construct the model and its execution takes longer run time. Thus a compromise has to be made for the level of detail, to obtain reasonably realistic results.

The mathematical models can handle only a limited range of problems, while simulation can handle all sorts of problems. It is said that when every method fails, simulation can be employed to solve the problem.

1.10 When to Use Simulation?

Over the years, tremendous developments have taken place in computing capabilities and in special purpose simulation languages, and in simulation methodologies. The use of simulation technique has also become widespread. The situation under which simulation should be used, have been discussed by many authors, and it is essential to understand the purposes and situation, where this technique should be employed. Following are some of the purposes for which simulation may be used.

1. Simulation is very useful for experiments with the internal interactions of a complex system, or of a subsystem within a complex system.
2. Simulation can be employed to experiment with new designs and policies, before implementing them.
3. Simulation can be used to verify the results obtained by analytical methods and to reinforce the analytical techniques.
4. Simulation is very useful in determining the influence of changes in input variables on the output of the system.
5. Simulation helps in suggesting modifications in the system under investigation for its optimal performance.

1.11 Steps in Simulation Study

Like any other problem solving approach, simulation is also carried efficiently, if it is done in a predetermined orderly manner. The total procedure has been divided into different number of steps by different authors. In general a simulation study can be divided into following prominent steps:

- Problem formulation
- Model construction
- Data collection
- Model programming
- Validation
- Design of experiment
- Simulation run and analysis
- Documentation
- Implementation

1.11.1 *Problem Formulation*

The clear and unambiguous description of the problem, definition of the objectives of the study, identification of alternatives to be considered and methodology for evaluating the effectiveness of these alternatives needs to be stated at the beginning of any study. If the statement of the problem is provided by the policy makers, the analyst must ensure that the problem being described is clearly understood. Alternatively, if the problem statement is being formulated by the analyst, the policy makers should be able to understand it and accept it. At this stage, it should also be ascertained, whether the simulation technique is the appropriate tool for solving the problem. The overall plan should include a statement of the alternative systems to be considered, the measures of performance to be used, the methodologies of analysis to be used, and the anticipated result of the study.

1.11.2 *Model Construction*

The model building is much of an art than science. There are no standard rules for building a successful and appropriate model for all types of situations. There are only certain guidelines, which can be followed. The art of modeling is enhanced by the ability to abstract the essential features of the system, to select and modify the basic assumptions and simplifications that characterize the system, and then improve and elaborate the model. To start with a simple model is constructed, which is modified step-by-step, every time enriching and elaborating its characteristics, to achieve an appropriate model, which meets the desired objectives. In some situations, building block method is employed, where the blocks of components of system are built and validated. These blocks are then combined to obtain model for the complete system.

1.11.3 *Data Collection*

The availability of input data about the system is essential for the construction of its model. The kind of data to be collected depends upon the objectives of the study, the required data may be available as past history, or may have to be collected. The construction of the simulation model and the collection of data have a constant interplay, and the type and amount of data required may change as the model develops. The data is required not only as an input to the model, but also some data is used to validate the simulation model. Since data collection generally takes longer time, it should be started as early as possible.

1.11.4 *Model Programming*

Any simulation model worth the name requires enormous amount of computations and information storage, which is possible only with the use of high-speed computers. The translation of the model into a computer recognizable format is termed as programming. Many general and special purpose simulation languages are available to write simulation programs. Many special purpose and problem specific simulation softwares have been developed which can be used for simulation modeling. It is for the modeler to decide, whether a simulation language is to be used or special purpose software is to be used. If the situation under study is amenable to an available special purpose software, the model development time and effort is considerably reduced. On the other hand, simulation languages are usually more powerful and more flexible than the special purpose software packages. The general programming languages like BASIC, FORTRAN, C, C++ have also been extensively used for writing the simulation programs.

1.11.5 *Validation*

It is essential to ensure that the model is an accurate representation of the system, which has been modeled. That the computer program performs properly and the results obtained are identical to the ones from the real system. Validation involves both the validation of the logic and accuracy of programming. This requires step-by-step modification of the model. It is rarely possible to develop a reasonably large simulation model in its entirety in first step. Good deal of debugging is required. The validation is thus an iterative process of comparing the model to actual system behavior, identifying the discrepancies, applying corrections and again comparing the performance. This process continues till a model of desired accuracy is obtained. The data collected from the actual system is of great help in validation of the model.

1.11.6 *Design of Experiment*

The simulation is basically experimentation on the model of the system under investigation. Simulation experiment in most of the situations involves stochastic variables, which result into stochastic results. The average values of result obtained may not be of desired reliability. To make the results meaningful, it is essential that simulation experiment be designed in such a way that the results obtained are within some specified tolerance limits and at a reasonable level of confidence. Decisions regarding the length of simulation run, initial conditions, removal of initial bias, number of replications of each run; use of variance reduction techniques etc. has to be made.

1.11.7 *Simulation Run and Analysis*

The simulation program is run as per the simulation design; the results are obtained and analyzed, to estimate the measures of performance of the system. Based on the results, a decision is made, whether or not any modification in the design of simulation experiment is needed. This step is a sort of validation of the simulation design. It may reveal that more runs or more replications are required.

1.11.8 *Documentation*

Documentation of a simulation program is necessary as the program can be used by the same or different analyst in future. The program can be used with modifications for some other identical situation, which can be facilitated if the program is adequately documented. Documentation of the simulation model, allows the user to change parameters of the model at will to investigate the influence on outputs, to find optimal combinations. The program should be so documented, that a new user can easily understand it.

1.11.9 *Implementation*

There will not be any problems in the implementation of the simulation program, if the user is fully conversant with the model, and understands the nature of its inputs and outputs and underlying assumptions. Thus, it is important that the model user is involved in the development of the simulation model from the very first step.

1.12 Phases of a Simulation Study

The process of simulation model development has been detailed under nine steps in the previous section. Some authors divide this process into following four phases:

Phase 1: Problem formulation: This includes problem formulation step.

Phase 2: Model building: This includes model construction, data collection, programming, and validation of the model.

Phase 3: Running the model: This includes experimental design, simulation runs and analysis of results.

Phase 4: Implementation: This includes documentation and implementation.

1.13 Advantages of Simulation

The use of the simulation technique is widespread, and it is gaining popularity day-by-day. There are many advantages of this technique over the other techniques. Some of these are given below.

1. Simulation helps to learn about a real system, without having the system at all. For example, the wind tunnel testing of the model of an aeroplane does not require a full sized plane.
2. Many managerial decision making problems are too complex to be solved by mathematical programming.
3. In many situations, experimenting with an actual system may not be possible at all. For example, it is not possible to conduct experiment, to study the behavior of a man on the surface of moon. In some other situations, even if experimentation is possible, it may be too costly and risky.
4. In the real system, the changes we want to study may take place too slowly or too fast to be observed conveniently. Computer simulation can compress the performance of a system over years into a few minutes of computer running time. Conversely, in systems like nuclear reactors where millions of events take place per second, simulation can expand the time to required level.
5. Through simulation, management can foresee the difficulties and bottlenecks, which may come up due to the introduction of new machines, equipments and processes. It thus eliminates the need of costly trial and error method of trying out the new concepts.
6. Simulation being relatively free from mathematics can easily be understood by the operating personnel and non-technical managers. This helps in getting the proposed plans accepted and implemented.
7. Simulation models are comparatively flexible and can be modified to accommodate the changing environment to the real situation.
8. Simulation technique is easier to use than the mathematical models, and can be used for a wide range of situations.
9. Extensive computer software packages are available, making it very convenient to use fairly sophisticated simulation models.
10. Simulation is a very good tool of training and has advantageously been used for training the operating and managerial staff in the operation of complex systems. Space engineers simulate space flights in laboratories to train the future astronauts for working in weightless environments. Airline pilots are given extensive training on flight simulators, before they are allowed to handle real aeroplanes.

1.14 Limitations of the Simulation Technique

In spite of all the advantages claimed by the simulation technique, many operations research analysts consider it a method of last resort, and use it only when all other techniques fail. If a particular type of problem can be shown to be well represented by a mathematical model, the analytical approach is considered to be more economical, accurate and reliable. On the other hand, in very large and complex problems, simulation may suffer from the same deficiencies as other mathematical techniques. In brief, simulation technique suffers from following limitations.

1. Simulation does not produce optimum results. When the model deals with uncertainties, the results of simulation are only reliable estimates subject to statistical errors.
2. Quantification of the variables is another difficulty. In a number of situations, it is not possible to quantify all the variables that affect the behavior of the system.
3. In very large and complex problems, the large number of variables, and the inter-relationships between them make the problem very unwieldy.
4. Simulation is by no means a cheap method of analysis. Even small simulations take considerable computer time. In a number of situations, simulation is comparatively costlier and time consuming.
5. Other important limitation stem from too much tendency to rely on the simulation models. This results in applications of the technique to some simple situations, which can more appropriately be handled by other techniques of mathematical programming.

1.15 Areas of Applications

System simulation is a technique, which finds applications in almost each and every field. Some of the areas in which it can be successfully employed are listed below:

Manufacturing: Design analysis and optimization of production system, materials management, capacity planning, layout planning and performance evaluation, evaluation of process quality.

Business: Market analysis, prediction of consumer behavior, optimization of marketing strategy and logistics, comparative evaluation of marketing campaigns.

Military: Testing of alternative combat strategies, air operations, sea operations, simulated war exercises, practicing ordinance effectiveness, inventory management.

Healthcare applications: Such as planning of health services, expected patient density, facilities requirement, hospital staffing, estimating the effectiveness of a health care program.

Communication applications: Such as network design and optimization, evaluating network reliability, manpower planning, sizing of message buffers.

Computer applications: Such as designing hardware configurations and operating system protocols, sharing and networking.

Economic applications: Such as portfolio management, forecasting impact of Govt. Policies and international market fluctuations on the economy, budgeting and forecasting market fluctuations.

Transportation applications: Design and testing of alternative transportation policies, transportation networks — roads, railways, airways etc., evaluation of timetables, traffic planning.

Environmental applications: Solid waste management, performance evaluation of environmental programs, evaluation of pollution control systems.

Biological applications: Such as population genetics and spread of epidemics.

There is no end to the list of applications. There is no area, where the technique of system simulation cannot be applied. However, the analyst must look into the possible mathematical techniques, before deciding to use simulation. In many situations, the use of simulation is uneconomical. Also simulation produces only estimates of system performance, while mathematical analysis provides accurate answers. As the complexities of the problem increase, the scope of application of simulation increases.

1.16 Simulation — A Management Laboratory

The technique of system simulation is a very important tool of decision making. The managerial problems are generally too complex to be solved by the analytical techniques. Various techniques of operations research are applicable to only specific types of situations, and require many assumptions and simplifications to be made for fitting the problem into the model. Many of the events occurring in real systems are random with intricate interrelationships, with their solution beyond the scope of standard probability analysis. Under the circumstances, simulation is the only tool, which allows the management to test the various alternative strategies. Since, simulation is a sort of experimentation, and when used for analyzing managerial problems, it is rightly called the management laboratory. For training the business executives, simulations called management games are used in many universities and management institutes.

1.17 Simulation in Design

Computer simulation has been very effectively used by the managers, administrators, computer system users and designers, for achieving high performance at comparatively low costs. In addition to using simulation for better understanding the systems and for optimizing their performance and reliability, simulation is a very good tool for verifying the correctness of designs. Most of the digital integrated circuits manufactured today are first simulated and intensively tested and verified before they are manufactured. The design of most of the complicated systems like robots, transfer lines, flexible manufacturing systems and automated guided vehicles, are first tested on simulation models. Simulation along with animation helps to test the interactions and interferences of various components of a system. The manufacturing systems of various types varying from the flow line production systems to flexible manufacturing system are tested and validated on their simulation models. The analysis, design and balancing of assembly lines are carried out by simulation of the line. Complex civil engineering structures are first modeled and tested before their actual erection. Simulation helps to identify the errors in design and to do the necessary corrections and carry out the desired modifications. It is thus important that simulation is employed early in the design cycle because the cost of repairing mistakes increases dramatically when these are detected late in the design and manufacturing cycle. Simulation is also very helpful in evaluating the alternative designs, production schedules and processing plans.

1.18 Simulation in Computer Science

Simulation has played a very important role in the design, analysis and optimization of computing systems. In computer science, simulation has a very specialized meaning, where the term simulation refers to what happens when a digital computer executes a program. The whole operation of the digital computer is simulated and all information about the inputs, outputs, transaction of states taking place during execution becomes available to the programmer. This helps in designing the computer architecture and optimizing its operation. The programmer can easily test the alternatives in design at different speed with different input data. In theoretical computer science, 'simulation' represents relationships between state transitions in systems. Simulation helps in the study of operational semantics.

In computer architecture, a simulation is used to test a program that has to run on some inconvenient machine. Computer architecture simulators are available which are used to build the test computer architecture. The simulation is used to debug the computer program, which may be micro-program or commercial application.

Simulation is also used to analyze the *fault trees*. Simulation helps in better designing of circuits and optimizing their performance. All VLSI logic circuits are first simulated and tested then constructed.

1.19 Simulation in Training

Simulation has long been used in military training. Earlier it used to be physical simulation of war games on boards and now it has changed to computer war games. Simulation is a very useful technique of training in situations where it is either too dangerous or prohibitively expensive to impart training on real systems. Like the training of a pilot is never carried on a real aeroplane, it is first on a flight simulator where the pilot is thoroughly trained in using the various controls and instruments. There are situations where training on real equipments in real situations is not possible at all, like the training of astronauts to walk in space or on other planets and to work in zero gravity environment. Such trainings are given in simulated virtual environments, which are created in laboratories. These simulated equipments and environments are safe, economical and can be manipulated to suit the training requirements.

In army, any training that is not a real combat is defined to be a simulation. The same is true in other fields also. The purpose of the simulated training is to place people in situations, which replicate those they will experience in real situations, to test their reactive and decision making capabilities.

1.20 Classification of Training Simulations

The training simulations are classified into three broad categories; *Live*, *Virtual* and *Constructive*.

(*a*) *Live simulation*: In live simulation, real people use simulated equipment in real world. Real-time simulations are live simulations. Soldiers operate their real equipment in mock engagements. This simulated combat trains the troops to experience the rigors of living and working in the field.

(*b*) *Virtual simulation*: In virtual simulation, real people use simulated equipment in a simulated environment, also called virtual environment. The training of space astronauts is done on simulated equipment in virtual environment. In the military games, this is a modification of live simulation in the sense that real equipment is replaced by mock-ups and the field of battle is generated by a computer. In these simulators, soldiers practice at much lower operational cost and with greater freedom in taking risks. Since both the equipment and the battlefield are virtual, troops can practice actions which are too dangerous to attempt in live simulations. Live simulations on the other hand are limited to the terrain that is available at training sites only.

(*c*) *Constructive simulation*: In this type of simulation, simulated people use simulated equipment in a simulated environment. Constructive simulation is also called "war-gaming" It has extensively been used in military training. It is similar to table top war games in which simulated players command simulated armies of soldiers and equipments, that move around on a board.

While the 'Live' and 'Virtual' simulations are used to train individual to operate equipments, constructive simulation trains the commanders to face situations and make decisions under the stress of time and limited resources just as they will during the actual combat. Constructive simulation helps the commanders to test their strategies in situations where the enemy is highly trained, fully equipped, totally unpredictable and fully determined to win.

Simulation training has been employed in almost all the fields, where training on real system is not feasible like in medical science, space science, navy, air force and army and in managerial decision making, etc.

1.21 Simulation in Education

Tell me, I forget
Show me, I remember
Involve me, I understand

Simulation is an interactive teaching-learning technique which helps in better understanding of the learning materials. Educational simulations are creative units of instructions which incorporate traditionally taught material into a simulated environment. In addition to being a rich and flexible tool for teaching and learning, it provides instructors the evaluation procedures to assess how well they are educating their students.

The simulation has long been used in training, but now it is finding many important applications in education. Since training itself is a part of education, the education simulations are similar to training simulations. The educational technology researchers have shown that the video games are an efficient way of learning, and that these can be used very effectively in the teaching learning process, by preparing video games based on the school or college curriculum. The Animated Narrative Vignettes (ANV) are cartoon-like video narratives of hypothetical and realty-based stories. The simulated programmed learning helps the individual to learn at his own pace and convenience, and test his power of understanding and problem solving skills etc.

Simulation is finding extensive application in the science and engineering laboratories, where virtual experimental set up are being used for performing experiments. These are like training simulators. Animated learning materials are available for learning engineering drawing and other subjects.

Simulation has proved to be a tool of extreme importance in medical educations. A simulated patient or a model patient, saves putting the real human being into the hands of students and inexperienced doctors. Students can work without any hesitation and repeat the medical procedures any number of times on a simulated patient.

Since, simulation-based education cannot completely replace the traditional classroom education, a combination or blend of various education techniques, called "Blended Education" is being actually implemented. The idea behind blended learning is that the education designers, prepare a learning program, divide it into modules and determine the best medium to deliver those modules to learners. Thus it involves mixing various form of education like the classroom teaching, internet-based learning and simulated instructional modules delivered both on and off line.

The simulated education has some unique advantages.

- Simulation works very well over the internet, where instructional material can be delivered to a large population of learners spread over the globe.
- To an individual learner, simulated education allows to learn at his own pace and to repeat the process as many times as one requires for complete understanding.
- Simulated laboratories/equipments allow the learners to practice without any fear of damage to equipment. Especially in medical education, simulated patient provides an excellent apprehension free learning environment.

There are a good number of reasons why other forms of education must complement a simulation.

- Some concepts can be taught better by a teacher and understood better by discussion among fellow learners. This is especially true in case of many soft skills.
- Meeting face to face with the teacher can be highly motivating.
- In many situations, working on real equipments and in real environments gives better understanding of the system compared to simulated environment.

1.22 Medical Simulators

Medical simulators are increasingly being developed and used to teach therapeutic and diagnostic procedures as well as medical concepts to the medical students. The simulators have been deployed for educating and training in the procedures ranging from the basic as blood drawing to laparoscopic surgery and trauma cases. The medical simulators are extremely useful for development of new medical tools and equipments, new therapies and treatments and for making decision regarding medication and treatment. Biomedical engineering makes good use of simulators.

Simulators replace the real human subjects and take them out of the hands of the inexperienced medical students and professionals. Using simulated patient models, medical students can practice a procedure or diagnosis a number of times. Replacing human for safety is though the biggest advantage of simulation, the other important advantage is in better training of the students. When working on simulators, students concentrate only on the essential element which has been modeled and are able to ignore the rest. They learn without any hesitation and gain better confidence in handling real patients.

Computer simulations have the advantage of allowing the student to make judgments as well as errors. The process of iterative learning through assessment, evaluation, decision making and error correction creates a much stronger learning environment then the passive instructions.

Many medical simulators comprise of a plastic simulated model of the relevant anatomy connected to a computer. These are generally life size models that respond to injected drugs and are programmed to create simulations of life-threatening emergencies. In other simulators, computer graphic techniques are employed to visualize the components and procedure. Simulators are also being used in the development of tools for diagnosis and treatment of cancer like diseases.

Earlier, physical models made of clay or stone were used to demonstrate the clinical features of disease states and their effects on human. Then came the active models that attempted to reproduce the living anatomy or physiology. More recently interactive models have been developed that respond to actions taken by a student or physician.

1.23 Exercises

1.1 Name two or three of the main entities, attributes, activities, events and state variables which are to be considered for simulating the operation of,

(*a*) Post office

(*b*) Cafeteria

(*c*) A hospital OPD

(*d*) A garment store

(*e*) An automobile assembly line

(*f*) A traffic crossing

(*g*) A bus stand.

1.2 What are the events and activities associated with parking your car in a paid parking ?

1.3 Identify minimum five endogenous and five exogenous activities associated with a production shop.

1.4 Give five examples of each of the following :

(*a*) Continuous system

(*b*) Discrete system

(*c*) Stochastic system

(*d*) Physical model

(*e*) Mathematical model.

1.5 List the entities, attributes, activities and state variables in the working of your college workshop.

Chapter 2

MONTE CARLO METHOD

2.1 Monte Carlo Method

The technique of making calculations by random sampling was though already used for solving statistical problems, the term 'Monte Carlo Method' seems to have come into existence in 1949,* when a paper, "The Monte Carlo Method" was published in the *Journal of American Statistical Association*. John Von Neumann and Stanislav Ulam are thought to have developed this method. In USSR, the first papers on the Monte Carlo Method were published in 1955-56.**

The technique 'Monte Carlo' derives its name from the city of Monte Carlo in the principality of Monaco, France, famous for gambling and its casino. Gambling is the game of probability and random sampling and so is the Monte Carlo Method. The roulette is a simple mechanical device of generating random numbers and is commonly used in gambling. Though the birth of the term Monte Carlo Method is connected with gambling games, yet it cannot help to win in roulette. It is a technique of experimental sampling with random numbers or the method of trials, which can be used to solve many problems, which are otherwise difficult or impossible to solve. J. Von Neumann and S. Ulam, who coined this term 'Monte Carlo' were involved in the design of nuclear shields at the Los Alamos Scientific Laboratory. They needed to know how far neutrons would travel through various materials. The problem was too difficult to solve analytically, and it was highly time consuming and hazardous to conduct experiments. They, simulated the experiment on a computer, using random numbers, and gave it the name Monte Carlo Method.

The term "Monte Carlo Method" is a very general term and the Monte Carlo Methods are used widely varying from economics to nuclear physics to waiting lines and to regulating the flow of traffic etc. Monte Carlo Methods are stochastic techniques and make use of random numbers and probability statistics to solve the problems. The way this technique is applied varies from field to field and problem to problem. Monte Carlo Method is applied to solve both deterministic as well as stochastic problems. There are many deterministic problems also which are solved by using random numbers and interactive procedure of calculations. In such a case we convert the deterministic model into a stochastic model, and the results obtained are not exact values, but only estimates. The application of Monte Carlo Method for evaluation of Pi (π) is converting a deterministic model into a stochastic model. Thus to call something a "Monte Carlo" experiment, all you need to do is to use random numbers to analyze the problem. Some simple examples will help to illustrate the use of random sampling in problem solving.

Example 2.1. Area of an Irregular Figure

Let us consider a very simple case of determining the area of a plane Fig. 2.1. This problem can easily be solved by simple calculations and has been taken only to demonstrate the application of Monte Carlo Method.

* Metropolis N., Ulam S., 'The Monte Carlo Method', J.Amer. Statistical Assoc., 1949, 44, No. 247.

** Papers by V.V.Chaveha nidze, Yu. A.Shreider and V.S.Vladiminov.

Enclose this figure in a regular figure of known area, say square of area A. Inside the square mark, at random, N number of points. This may be done by asking a man with closed eyes, to mark the points on the paper. Count the points inside the irregular figure, let these be M, while the points inside the square are N. Now, it is geometrically obvious that the ratio of area F of the irregular figure to the known area A, would approximately be the same as the ratio M to N.

i.e., $$\frac{F}{A} = \frac{M}{N} \quad \text{or} \quad F = \frac{M}{N}A$$

In Fig. 2.1 (*a*), $N = 50$ and $M = 17$ which gives the ratio $\frac{M}{N}$ as 0.34, while the true ratio in this case is 0.3126. If the number of random points N is increased, the accuracy of the result will also increase. In the second experiment, Fig. 2.1 (*b*), $N = 84$, and count of M comes to 26, giving the ratio $\frac{M}{N} = 0.3095$ which is more nearer to the actual ratio of 0.3126. Thus if $A = 9 \times 9 = 81$ cm², area $F = .3095 \times 81 = 25.07$ cm².

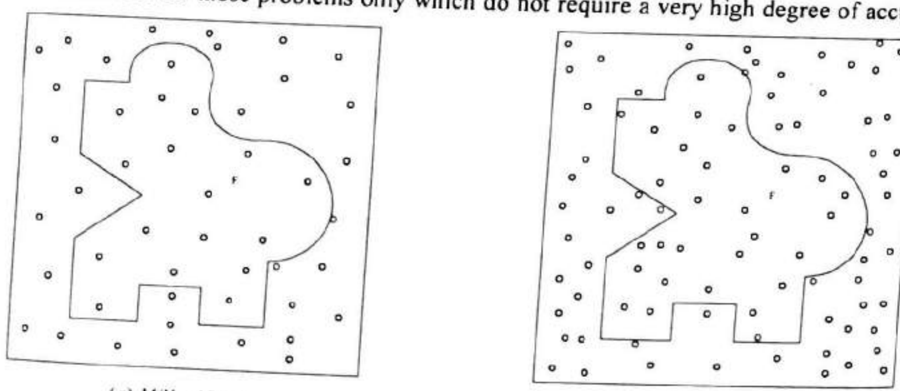
In this experiment, the randomness of the points is very important. Various methods can be employed for marking the points, and one such method is the use of random number tables. For each random point, two random numbers will be used to generate the two coordinates of the point.

In the present case our square is 9×9 cm². That is each side is 9 cm long. If we take the random numbers between 0 and 1, then the co-ordinates of a random point are obtained by multiplying the random number by 9.

Let .96, .42, .71, .10, .03, and .32, ..., be the random numbers, then we have

$$x1 = .96 \times 9 = 8.64, \quad y1 = .42 \times 9 = 3.78$$
$$x2 = .71 \times 9 = 6.39, \quad y2 = .10 \times 9 = 0.9$$
$$x3 = .03 \times 9 = 0.27, \quad y3 = .30 \times 9 = 2.7$$

Larger the number of points, more accurate will be the result. The error of calculation is proportional to $\frac{1}{N^2}$. Thus to reduce the error to half, value of N will have to be increased four times, and to have one more accurate decimal digit in the result a 100 fold increase in N will be required. It is thus clear that the Monte Carlo Method cannot be used to attain high accuracy and hence the Method is suitable for those problems only which do not require a very high degree of accuracy.



(*a*) $M/N = 17/50$ (*b*) $M/N = 26/84$

Fig. 2.1

Example 2.2. A Gambling Game

To illustrate the case of a random process, let us consider a game in which an unbiased coin is repeatedly flipped. For each flip you have to pay Re. 1. and when the difference between heads tossed

and tails tossed becomes 3, you get Rs. 8. Thus if the required difference is obtained in less than 8 flips, you win some money and if it is in more than 8 flips you lose. How to decide, whether to play this game or not ?

The easiest way is to play the game at home and check, that is what the Monte Carlo Method is, to simulate the process by carrying trials. A coin can be used for this purpose or a random number table can be made use of. Larger the number of trials we carry, more accurate will be our estimate.

When a coin is flipped, the probability of head or tail is same that is 50%, but there are 10 possible one digit random numbers. Thus half of these (0, 1, 2, 3, 4) can be taken for head and the remaining half (5, 6, 7, 8, 9) for the tail.

Following is a string of random numbers noted down from a random number table.

5, 9, 3, 6, 4, 8, 6, 8, 1 ———

If H denotes head and T denotes tail, the above random numbers simulate the flips as,

T T H T H T T , T H ———.

We notice that after 7 trails the difference between heads and tails becomes 3 (2 heads 5 tails). Thus in the first game you win Re. 1. Some more games are simulated in Table 2.1. For the first seven games, we find that total money paid is Rs. 7 + 9 + 5 + 11 + 11 + 9 + 7 = 59, while the return is Rs. 56. Thus each game on the average requires $\frac{59}{7} = 8.43$ flips.

This small simulation shows that you will lose some money in the long run. But can this estimate be relied upon ? No, since the number of trials conducted is very small, the accuracy of result would be very poor. If this simulation is continued over a sufficiently long period that is for a large number of trials the estimate close to the true mean can be obtained, which in this case is 9 flips for each game. Thus in the long run, you will lose Re. 1 per game.

Table 2.1. Simulation of a Gambling Game

Game No.	*Sl. No.*	*Random Number*	*Head or Tail*	*Cumulative Heads*	*Cumulative Tails*	*Difference*
1	1	5	T	0	1	1
	2	9	T	0	2	2
	3	3	H	1	2	1
	4	6	T	1	3	2
	5	4	H	2	3	1
	6	8	T	2	4	2
	7	6	T	2	5	3
						Win Re. 1
2	1	8	T	0	1	1
	2	1	H	1	1	0
	3	5	T	1	2	1
	4	2	H	2	2	0
	5	4	H	3	2	1
	6	0	H	4	2	2
	7	6	T	4	3	1
	8	3	H	5	3	2
	9	1	H	6	3	3
						Lose Re. 1

3	1	2	H	1	0	1	
	2	9	T	1	1	0	
	3	4	H	2	1	1	
	4	2	H	3	1	2	
	5	3	H	4	1	3	
							Win Rs. 3
4	1	3	H	1	0	1	
	2	3	H	2	0	2	
	3	7	T	2	1	1	
	4	1	H	3	1	2	
	5	8	T	3	2	1	
	6	0	H	4	2	2	
	7	6	T	4	3	1	
	8	2	H	5	3	2	
	9	9	T	5	4	1	
	10	3	H	6	4	2	
	11	4	H	7	4	3	
							Lose Rs. 3
5	1	3	H	1	0	1	
	2	2	H	2	0	2	
	3	9	T	2	1	1	
	4	6	T	2	2	0	
	5	1	H	3	2	1	
	6	8	T	3	3	0	
	7	7	T	3	4	1	
	8	0	H	4	4	0	
	9	8	T	4	5	1	
	10	6	T	4	6	2	
	11	7	T	4	7	3	
							Lose Rs. 3
6	1	4	H	1	0	1	
	2	1	H	2	0	2	
	3	8	T	2	1	1	
	4	2	H	3	1	2	
	5	6	T	3	2	1	
	6	0	H	4	2	2	
	7	9	T	4	3	1	
	8	2	H	5	3	2	
	9	4	H	6	3	3	
							Lose Re. 1
7	1	9	T	0	1	1	
	2	8	T	0	2	2	
	3	3	H	1	2	1	
	4	9	T	1	3	2	
	5	3	H	2	3	2	
	6	7	T	2	4	2	
	7	5	T	2	5	3	
							Win Re. 1

Example 2.3. Numerical Integration by Monte Carlo Method

The application of Monte Carlo Method for the integration of a function, can best be illustrated with the help of an example. Let us consider a simple case of a single variable function.

$$I = \int_2^5 x^3 dx = \int_a^b f(x)dx$$

The exact value of the integral, $I = \left[\frac{x^4}{4}\right]_2^5 = 152.25$

This function $f(x) = x^3$ is plotted in Fig. 2.2.

The area under the curve between the limits $x = 2$ and $x = 5$, shown shaded is the value of the integral. The value of the area can be obtained in the same way as of irregular figure in Example 2.1. Let this area be enclosed in a rectangle $ABCD$, whose area is known ($140 \times 3 = 420$). The darts are thrown at random on to the figure. Let N darts fall in the rectangle $ABCD$ and out of these let M darts fall in the shaded area. In other words, out of N random marks made on $ABCD$, M marks lie in the shaded area.

Then, the ratio of the shaded area I to the area A of rectangle $ABCD$ is M to N.

or
$$I = \frac{M}{N} A$$

The correctness of the result will depend upon the number of trials that is darts thrown. Larger the value of N, more accurate will be the result.

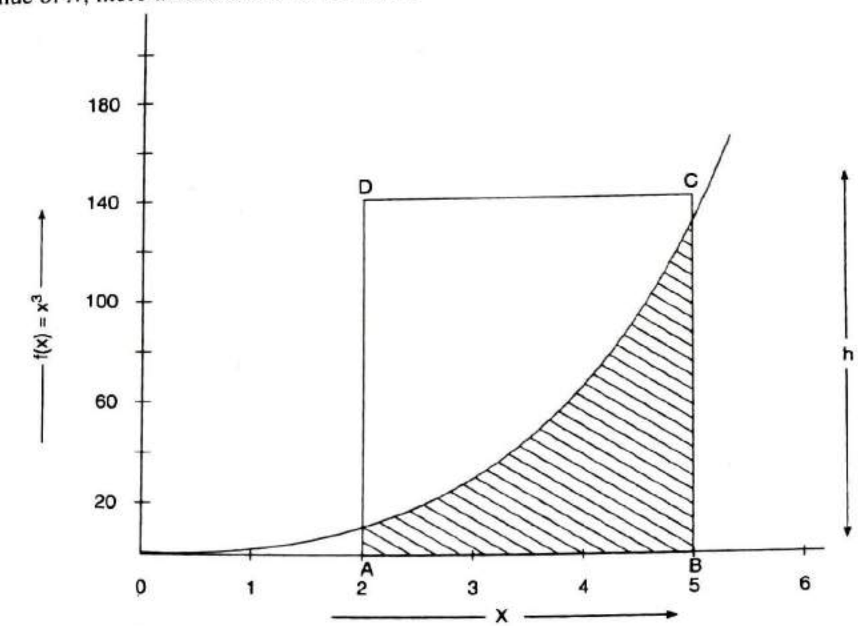


Fig. 2.2

Instead of throwing darts on the area, we can make use of random numbers. To mark a random point we need two random numbers to generate the two co-ordinates x and y. We can make use of a random number generator or of a random number table.

Let us take two digit random numbers between 20 and 50 to represent the x co-ordinate. If r is a random number between 20 and 50, then $x = 0.1\ (r)$. To generate y co-ordinate, take random number between 0 and 1, and multiply it by the height of the rectangle $ABCD$. Now check if $y \le x^3$. If so, the point lies below the curve, otherwise it is above the curve. Simulation procedure is demonstrated in Table 2.2. The point which lies below the curve *i.e.*, have $y \le x^3$ is added to M, while N is the number of trials (points generated). For twenty points simulated in the table, 8 points lie in the shaded area ($M = 8$; $N = 20$). Thus, the value of integral is $I = \frac{8}{20} \times A = 0.4 \times 420 = 168.00$. Actual value of the integral is 152.25. By increasing the number of points, value of I quite close to the actual value can be obtained, and this large number of points can be generated by a computer. Statistical techniques can be employed to determine the sample size, so as to obtain the result with reasonable accuracy. A flow diagram of the computer program, for evaluating an integral by Monte Carlo Method is given in Fig. 2.3.

Table 2.2. Simulation of $I = \int_2^5 x^3 dx$

Random Number	*x* *0.1 (r)*	*Random Number*	*y* *140 (r)*	x^3	*M*	*N*
22	2.2	.57	79.8	10.65	0	1
25	2.5	.18	25.2	15.63	0	2
18	1.8	.00	0.0	5.83	1	2
45	4.5	.90	126.0	91.13	1	4
25	2.5	.05	7.00	15.63	2	5
27	2.7	.77	107.8	19.68	2	6
48	4.8	.66	92.4	110.60	3	7
43	4.3	.10	14.0	79.51	4	8
40	4.0	.76	106.4	64.00	4	9
47	4.7	.42	58.8	103.82	5	10
38	3.8	.78	109.2	54.87	5	11
33	3.3	.88	123.2	35.94	5	12
24	2.4	.03	4.2	13.82	6	13
47	4.7	.09	12.6	103.80	7	14
42	4.2	.77	107.8	74.09	7	15
25	2.5	.61	85.4	15.63	7	16
33	3.3	.27	37.8	35.94	7	17
50	5.0	.60	84.0	125.00	8	18
34	3.4	.29	40.6	39.30	8	19
21	2.1	.40	56.0	9.26	8	20

$$I = \left(\frac{8}{20}\right) \times (140 \times 3) = 168.00$$

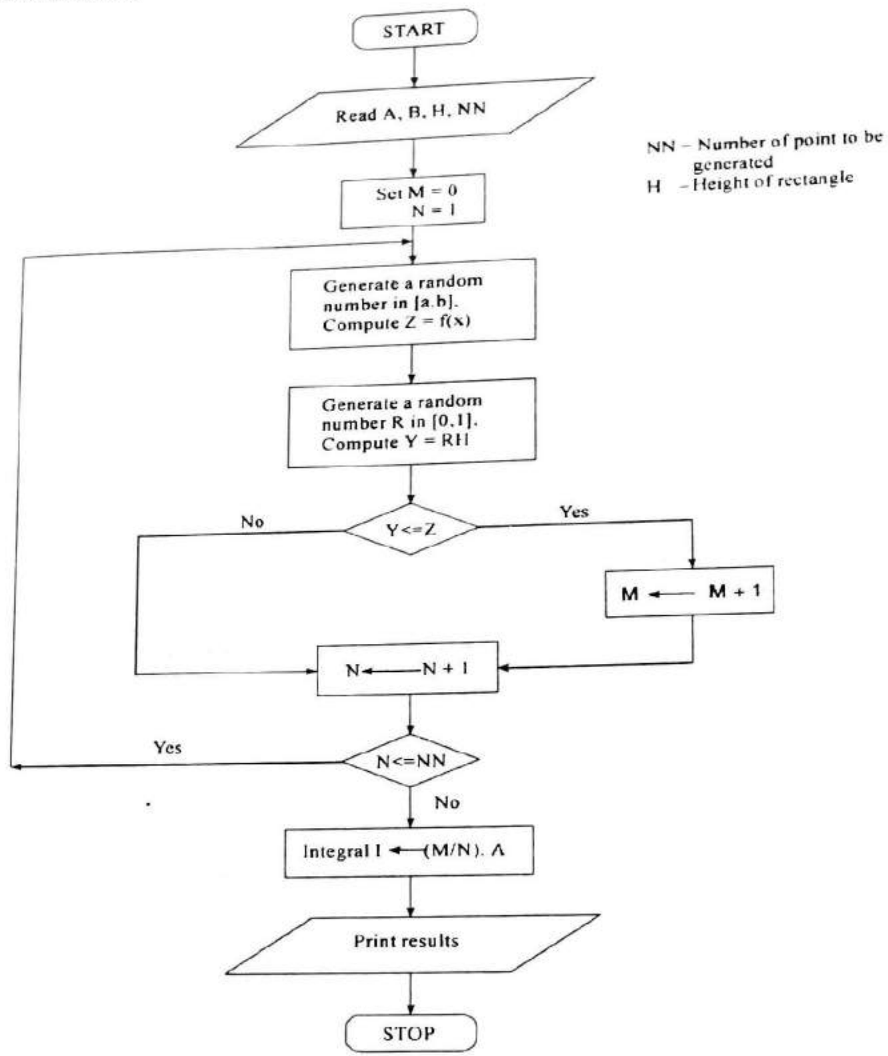


Fig. 2.3

A Note: It is not computationally economical to use the Monte Carlo Method for evaluating an integral of a single variable, especially when more efficient numerical methods are available. The method however is often used for solving multivariable integrals.

The problems of numerical integration and the area of an irregular figure, have been discussed to illustrate the application of Monte Carlo technique to problems of deterministic type, while Example 2.2, a gambling game, demonstrated the application of the method to probabilistic problems. Some, more problems are discussed in the following sections to illustrate the use of Monte Carlo Method, in handling various types of situation.

Example 2.4. Determination of value of π by Monte Carlo Method

The value of π can be determined by using the relation for area of a circle.

$$\text{Area of circle} = \pi r^2$$

Let us consider a quadrant of a unit circle as shown in Fig. 2.4.

Area of the quadrant is $\dfrac{\pi r^2}{4} = \dfrac{\pi}{4}$.

All points satisfying the equations $x^2 + y^2 \le 1;\ x, y \ge 0$ lie in this quadrant. Now if we have a pair of random numbers R_1 and R_2 in the range (0,1), then the point R_1 and R_2 may lie within the quadrant or outside the quadrant but within the square enclosing the quadrant. If we generate a large number of such points (say N) by taking pairs of random numbers, and out of them M lie within the quadrant, then the ration M/N will approach the area under the curve, which is $\dfrac{\pi}{4}$.

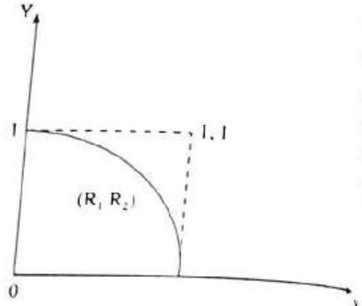


Fig. 2.4

The computations for 40 points are given in Table 2.3. For each pair of random numbers, the point is within the quadrant, when $R_1^2 + R_2^2 \le 1$

or $$R_2 \le \sqrt{1 - R_1^2}$$

Table 2.3. Value of π by Monte Carlo Method

R_1	R_2	$\sqrt{1-R_1^2}$	In/Out	R_1	R_2	$\sqrt{1-R_1^2}$	In/Out
.82	.95	.5724	Out	.48	.37	.8773	In
.34	.14	.9404	In	.51	.72	.8602	In
.20	.84	.9797	In	.06	.33	.9982	In
.58	.14	.8146	In	.22	.90	.9755	In
.52	.03	.8542	In	.80	.76	.6000	Out
.51	.60	.8602	In	.56	.29	.8285	In
.34	.38	.9404	In	.06	.66	.9982	In
.72	.11	.6940	In	.92	.40	.3919	Out
.50	.55	.8760	In	.51	.11	.8602	In
.08	.21	.9967	In	.13	.44	.9915	In
.76	.18	.6499	In	.65	.74	.7599	In
.33	.81	.9440	In	.60	.27	.8000	In
.35	.96	.9368	Out	.51	.16	.8602	In
.29	.57	.9570	In	.50	.41	.8660	In
.95	.47	.3123	Out	.13	.20	.9915	In
.84	.51	.5426	In	.94	.68	.3412	Out
.34	.68	.9404	In	.51	.95	.8216	Out
.85	.11	.5268	In	.26	.28	.9656	In
.58	.92	.8146	Out	.78	.75	.6258	Out
.69	.59	.7238	In	.33	.16	.9440	In

Out of 40 points, 31 lie within the quadrant, giving the value of π, as

$$\pi = \frac{31}{40} \times 4 = 3.10$$

The accuracy of the result can be increased by increasing the number of observations.

Example 2.5. Random Walk Problem

A drunkard moves from a point to a destination and takes steps in the forward direction, to his left and to his right, at random. The length of the step is almost constant. The probability of taking a step in the forward direction is 50%, while the probability of taking a step to the left 30% and to the right is 20%. If the co-ordinates of the starting point are taken as (0, 0), and the forward is due Y-direction, find the position of the drunkard after he takes 50 steps.

This type of problem, where a person walks at random in different direction is called a random walk. In this simple case, the position of the drunkard can easily be determined by using the probability theory. But we will simulate the walk using random numbers.

Using single digit random numbers, the random numbers can be allocated to steps in different directions, as under:

Direction		*Probability*	*Random Numbers*
Forward (F)	:	0.5	0, 1, 2, 3, 4
Left (L)	:	0.3	5, 6, 7
Right(R)	:	0.2	8, 9

Table 2.4. Random Walk Simulation

Step	*Random Number*	*Direction*	*Position*	
			x	*y*
1	6	L	–1	0
2	2	F	–1	1
3	0	F	–1	2
4	6	L	–2	2
5	8	R	1	2
6	5	L	–2	2
7	7	L	–3	2
8	7	L	–4	2
9	9	R	–3	2
10	8	R	–2	2
11	4	F	–2	3
12	8	R	–1	3
13	2	F	–1	4
14	6	L	–2	4
15	2	F	–2	5
16	1	F	–2	6
17	3	F	–2	7
18	0	F	–2	8
19	8	R	–1	8
20	4	F	–1	9

When the drunkard takes steps in forward direction, the value of y is incremented by one. If the drunkard moves to the right value of x is incremented by one, and if he moves towards his left, value of x is decremented by one. The simulation for 20 steps is given in Table 2.4. The trace of the movement of the drunkard for the 20 steps is given in Fig. 2.5. In the first 20 steps, the drunkard moves to a position (– 1, 9).

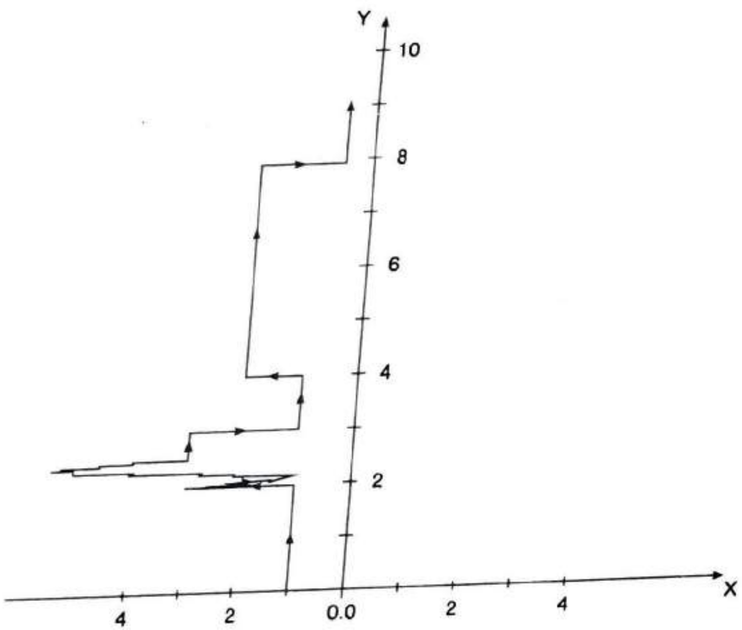


Fig. 2.5

Example 2.6. Reliability Problem

A machining center has three identical bearings, which fail according to the following probability distribution.

Bearing Life (Hours)	*Probability*
1000	0.10
1100	0.14
1200	0.24
1300	0.14
1400	0.12
1500	0.10
1600	0.06
1700	0.05
1800	0.03
1900	0.02

The present maintenance policy is to change a bearing as and when it fails. When a bearing fails, machining center stops, a repairman is called to replace the failed bearing with a new one. The time between the failure of the bearing and reporting of the repairman (delay time) is random and is distributed as under.

Delay Time (Minutes)	*Probability*
4	0.3
6	0.6
8	0.1

It takes 20 minutes to change one bearing, 30 minutes to change two bearings and 40 minutes to change all the three bearings. Downtime of the machining center costs Rs. 5 per minute, direct on job cost of the repairman is Rs. 25 per hour and the cost of a bearing is Rs. 20.

The maintenance department is interested in evaluating an alternative policy of replacing all the three bearings, whenever a bearing fails.

Solution: The reliability problem having random bearing lives and random repairman delay times can best be handled by employing simulation technique.

Table 2.5 shows the bearing life probability distribution, cumulative probability distribution and the two digit random numbers assigned to estimate the bearing lives. Table 2.6 shows the probability distribution and single digit random numbers assigned to determine the delay times of the repairman.

Table 2.5

Bearing Life (Hours)	*Probability*	*Cumulative Probability*	*Random Digits Assignment*
1000	0.10	.10	01 – 10
1100	0.14	.24	11 – 24
1200	0.24	.48	25 – 48
1300	0.14	.62	49 – 62
1400	0.12	.74	63 – 74
1500	0.10	.84	75 – 84
1600	0.06	.90	85 – 90
1700	0.05	.95	91 – 95
1800	0.03	.98	96 – 98
1900	0.02	1.00	99 – 00

Table 2.6

Delay Time (Minutes)	*Probability*	*Commutative Probability*	*Random Digits Assignment*
4	0.3	0.3	1 – 3
6	0.6	0.9	4 – 9
8	0.1	1.0	0

Table 2.7. Simulation of Reliability Problem – Present Policy

Bearing 1					Bearing 2					Bearing 3				
RD	Life	Clock	RD	Delay	RD	Life	Clock	RD	Delay	RD	Life	Clock	RD	Delay
25	1200	1200	9	6	65	1400	1400	5	6	94	1700	1700	7	6
82	1500	2700	8	6	91	1700	3100	0	8	87	1600	3300	0	6
16	1100	3800	0	8	30	1200	4300	5	6	63	1400	4700	6	6
14	1100	4900	5	6	66	1400	5700	6	6	66	1400	6100	2	4
82	1500	6400	2	4	32	1200	6900	5	6	30	1200	7300	2	4
45	1200	7600	3	4	29	1200	8100	3	4	69	1400	8700	7	6
81	1500	9100	7	6	11	1100	9200	2	4	37	1200	9900	2	4
80	1500	10600	3	4	43	1200	10400	8	6	01	1000	10900	8	6
97	1800	12400	7	6	40	1200	11600	0	8	66	1400	12300	8	6
62	1300	13700	4	6	65	1400	13000	8	6	51	1300	13600	3	4
30	1200	14900	9	6	82	1500	14500	2	4	92	1700	15300	8	6
31	1200	16100	5	6	73	1400	15900	4	6	36	1200	16500	3	4
80	1500	17600	6	6	15	1100	17000	4	6	47	1200	17700	7	6
72	1400	19000	7	6	70	1400	18400	6	6	80	1500	19200	0	8
63	1400	20400	6	6	65	1400	19800	9	6	94	1700	20900	4	6
59	1300	21700	0	8	33	1200	21000	9	6	31	1200	22100	7	6
09	1000	22700	2	4	54	1300	22300	5	6	07	1000	23100	1	4
25	1200	23900	3	4	87	1600	23900	7	6	09	1000	24100	6	6
70	1400	25300	9	6	27	1200	25100	7	6	19	1100	25200	2	4
64	1400	26700	8	6	37	1200	26300	4	6	29	1200	26400	3	4
61	1300	28000	3	4	99	1900	28200	8	6	29	1200	27600	2	4
92	1700	29700	3	4	94	1700	29900	4	6	94	1700	29300	5	6
12	1100	30800	2	4	12	1100	31000	4	6	87	1600	30900	6	6
		Total Delay		126			Total Delay		136			Total Delay		122

According to the existing repair policy, each bearing is replaced as and when it fails. It is highly unlikely that more than one bearing will fail at the same time, hence, it can be safely assumed that only one bearing is changed at any breakdown. Simulation of the system for 30,000 hours of operation, carried out manually, is given in Table 2.7. In case of each of the bearings, 23 changes have occurred during the simulation run. Thus in all 69 bearings have been replaced. The delay times in case of bearings numbered 1, 2 and 3 are 126, 136 and 122 minutes respectively, making a total of 384 minutes of repairman delay time. Since, it takes 20 minutes to change one bearing, the time spent in bearings replacement is 1380 minutes. Thus total downtime of the machine is 384 + 1380 = 1764 minutes. On job time spent by repairman is $69 \times 20 = 1380$ minutes. The estimate cost of the existing replacement policy is as below.

Cost of bearings	=	69×20	= 1380.00
Cost of downtime	=	1764×5	= 8820.00
Cost of repairman	=	$1380 \times \frac{25}{60}$	= 575.00
			10,775.00

The simulation of the system for the proposed repair policy is given in Table 2.8. Simulation has been carried for the same simulation run of 30,000 hours, and using the same life times as in the simulation for the existing policy. Since, the number of replacements in this case is more, additional bearing lives have been generated by using the next random numbers in the related series. The advantages of common random numbers are discussed in Chapter 9 of the book.

Table 2.8. Simulation of Reliability Problem — Proposed Policy

Bearing 1 Life (Hrs.)	*Bearing 2 Life (Hrs.)*	*Bearing 3 Life (Hrs.)*	*First Failure (Hrs.)*	*Cumulated Life (Hrs.)*	*RD*	*Delay (Minutes)*
1200	1400	1700	1200	1200	4	6
1500	1700	1600	1500	2700	6	6
1100	1200	1400	1100	3800	7	6
1100	1400	1400	1100	4900	3	4
1500	1200	1200	1200	6100	4	6
1200	2400	4700	1200	7300	6	6
1500	1100	2200	1100	8400	7	6
1500	1200	1000	1000	9400	7	6
1800	1200	1400	1200	10600	4	6
1300	1400	1300	1300	11900	7	6
1200	1500	1700	1200	13100	1	4
1200	1400	1200	1200	14300	0	8
1500	1100	1200	1100	15400	0	8
1400	1400	1500	1400	16800	1	4
1400	1400	1700	1400	18200	9	6
1300	1200	1200	1200	19400	5	6
1000	1300	1000	1000	20400	0	8
1200	1600	1000	1000	21400	6	6
1400	1200	1100	1100	22500	4	6
1400	1200	1200	1200	23700	1	4
1300	1900	1200	1200	24900	4	6
1700	1700	1700	1700	26600	9	6
1100	1100	1600	1100	27700	4	6
80/1500	56/1300	85/1600	1300	29000	4	6
21/1100	51/1300	66/1400	1100	30100	6	6
					Total Delay	**148**

The simulation of proposed policy reveals that for the same operation time 25 replacements of bearings will occur, requiring a total of 75 bearings. The delay time is 148 minutes, while the time spent in changing the bearings will be $25 \times 40 = 1000$ minutes, giving total downtime of 1148 minutes.

Cost of bearings $= 75 \times 20 = 1500.00$

Cost of downtime $= 1148 \times 5 = 5740.00$

Cost of repairman $= 1000 \times \frac{25}{60} = 416.67$

7656.67

The proposed repair policy is thus much more economical and will result in a saving of Rs. 3118.33 over a period of 30,000 hours of machine operation and hence, can be recommended for implementation.

2.2 Normally Distributed Random Numbers

There are a large number of real life situations, where the behavior of the system is described by normal probability distribution. The marks obtained by students in a class, number of defective parts produced, dimensions of parts made on a machine, number of bombs hitting a target area etc are generally described by the normal distribution. One method of generating the variates of such a distribution, by using the random numbers is described in Chapter 5. The value of variate y is calculated as,

$$y = \mu + \sigma\left(\sum_{i=1}^{12} r_i - 6\right), \qquad i = 1, 2, \ldots\ldots, 12.$$

In this method, 12 random numbers are used to generate one value.

The alternative method is the use of random numbers, which themselves are normally distributed about a mean of zero.

$$y = \mu + \sigma z$$

where z is a random normal number. In small simulations, where computations involving normal distribution are to be carried out, the tables of random normal numbers can be used. A sample of such numbers is given in Appendix Table A-3.

Example 2.7. Application of Random Normal Numbers

An ammunition depot, of rectangular shape measuring 1000 m along X-direction and 600 m along Y-direction is under attack from a squadron of bombers. In each sortie 10 bombers drop bombs, one each, on the ammunition depot. If the bomb lands anywhere in the marked area, it is a hit, otherwise it is a miss. All the bombers aim at the center of the area. The point of strike is assumed to be normally distributed around the aiming point with a standard deviation of 500 meters in the X-direction and 300 meters in the Y-direction. The operation of bombing is to be simulated to determine the percentage number of strikes, which are on the target.

Since, the strike points are randomly distributed about the point of aim, and the distribution is normal, the random normal numbers (normally distributed random numbers) can be used to generate the strikes.

A normal random variable X is given by:

$$X = \mu + \sigma_z$$

where μ is the true mean of the distribution, σ is the standard deviation of x and z is the normal random number.

If x and y are the co-ordinates of a point, then

$$x = \mu_x + \sigma_x z_x$$
$$y = \mu_y + \sigma_y z_y$$

z_x and z_y are normal random numbers. Taking the co-ordinates of the center point, of the area as 0, 0,

$$\mu_x = \mu_y = 0$$

The given values of standard deviations in X- and Y-directions are 500 and 300 meters respectively. Therefore, the co-ordinates of a point are,

$$x_1 = 500\ Z_x$$
$$y_1 = 300\ Z_y$$

The subscripts x and y added to the normal random number Z are only to indicate that different random numbers are to be taken. In this example, since manual simulation is to be done, the normal random number table will be used. To find some reasonable answer to the problem, a well-designed simulation experiment of 40 to 50 bombing sorties may be required. However, a simulation of only two sorties that is 20 bombs has been done to demonstrate the simulation procedure. Table 2.9 shows the results for this simulation. In the table the mnemonic RNN stands for 'random normal number'. The co-ordinates of each point have been generated. Those who lie within the marked area are designated as 'hit', others as 'miss'. This type of simulation is called Monte Carlo or Static Simulation, as process is not time dependent.

Table 2.9. Simulation of Bombing Problem

Bomb strike	*RNN*	*x*	*RNN*	*y*	*Result*
1	.23	115	.24	72	Hit
2	–1.16	–580	–0.02	–6	Miss
3	0.39	195	0.64	192	Hit
4	–1.90	–950	–1.04	–312	Miss
5	–0.78	–390	0.68	204	Hit
6	–0.02	–10	–0.47	–141	Hit
7	–0.40	–200	–0.75	–225	Hit
8	–0.66	–330	–0.44	–132	Hit
9	1.41	705	1.21	363	Miss
10	0.07	35	–0.08	–24	Hit
11	0.53	265	–1.56	–468	Miss
12	0.03	15	1.49	447	Miss
13	–1.19	–595	–1.19	–357	Miss
14	0.11	55	–1.86	–558	Miss
15	0.16	80	1.21	363	Miss
16	–0.82	–410	0.75	225	Hit
17	0.42	210	–1.50	–450	Miss
18	–0.89	–445	0.19	57	Hit
19	0.49	249	–1.44	–432	Miss
20	–0.21	–105	0.65	195	Hit

Number of hits $= 10$

Number of misses $= 10$

% Number of strikes on target $= 50\ \%$

2.3 Monte Carlo Method Vs. Stochastic Simulation

The term 'Monte Carlo Method' is very general and as discussed earlier is based on gambling like principles. It makes use of random numbers for finding the solution to the problems. Some authors differentiate between Monte Carlo Method and Stochastic Simulation, but that difference has only academic value.

Monte Carlo Method is considered to be a method of solving deterministic problems by employing random numbers, like finding the area of irregular figure, numerical integration, evaluation of π, trajectory computations etc. The stochastic simulation is used for finding the solution to a problem by employing random numbers, when the problems are stochastic. The gambling problem, the bombing problem, the reliability problem are the examples of stochastic simulation.

In practice both the Monte Carlo Method and stochastic simulation are employed to analyze models having random variables and both make use of random numbers and probability distribution to investigate the problem. Monte Carlo Methods has been employed in a wide range of situations varying from economics to nuclear physics, from regulating traffic to balancing of assembly lines, most of which are stochastic processes. The term "stochastic" is a synonym for random while the stochastic process is a particular type of model that represents the uncertainty in a dynamic system using the language of probability. Monte Carlo method and stochastic simulation have become almost synonyms and are used alternatively to describe the same method of stochastic modeling.

Example 2.8. Example of Stochastic Model – A Hinge Assembly

Fig. 2.6 shows a hinge assembly which comprise of four parts A, B, C and D, with a pin or bolt though the centre of the parts. The dimensions of the parts along the axis are critical, since if $a+b+c$ is greater than d, it will not be possible to put the parts together. The dimensions a, b, c and d are 2 ± 0.05, 2 ± 0.05, 30 ± 0.5 and 34.5 ± 0.5 respectively. Each of a, b, c and d is uniformly distributed over the given range. There is a huge stock of each part and the parts are randomly selected for making the hinge assemblies. What is the probability that the selected parts will not assemble?

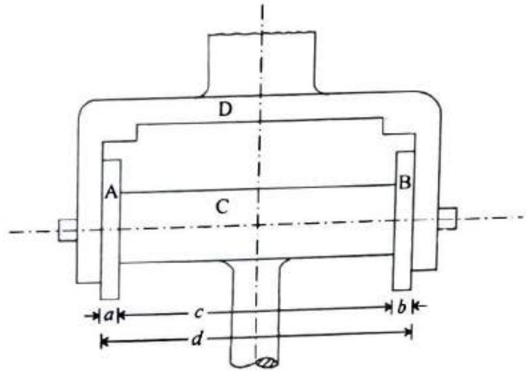


Fig. 2.6. Hinge assembly

Solution: In this example, we have to simulate the assembly process to obtain a good number of hinge assemblies. For each assembly the four parts are to be selected or we can say the dimensions a, b, c and d are to be generated. We will generate the critical dimensions by employing the two digit random numbers between 0 and 1. The range over which the dimensions are uniformly distributed is as given below:

Part	Minimum	Maximum
a	1.95	2.05
b	1.95	2.05
c	29.50	30.50
d	34.00	35.00

If R is a random number between 0 and 1, then $a = 1.95 + 0.1\,(R)$ where 1.95 is the minimum dimension.

Similarly,

$$b = 1.95 + 0.1\,(R)$$
$$c = 29.50 + 1\,(R)$$
$$d = 34.00 + 1\,(R)$$

After generating the dimension a, b, c and d we have to check the value of clearance, $x = d - (a + b + c)$. If the clearance is zero or positive parts will assemble with zero or positive clearance. If it is negative the parts will not assemble. Then we have to repeat the process of generating assemblies for different sets of parts. More the number of assemblies, better the result. Normally more than a 1000 iterations will be required to reach a reasonable answer to the problem.

Here we do it for a few iterations manually to demonstrate the simulation. The progress of simulation is shown in Table 2.10.

Table 2.10

Sr.	Random number	$a = 1.95 + 0.1(R)$	Random number	$b = 1.95 + .1R$	Random number	$c = 29.5 + R$	$a+b+c$	Random number	$d = 34 + R$	Clearance
1	.44	1.994	.56	2.006	.32	29.82	33.820	.12	34.12	+
2	.86	2.036	.69	2.019	.41	29.91	34.064	.83	34.83	+
3	.72	2.022	.52	2.002	.76	30.26	34.334	.34	34.34	+
4	.87	2.037	.84	2.034	.90	30.40	34.471	.42	34.42	–
5	.47	1.997	.43	2.043	.42	29.92	33.960	.97	34.37	+
6	.48	1.998	.51	2.001	.06	29.56	33.559	.22	34.22	+
7	.80	2.030	.56	2.006	.13	29.63	33.666	.92	34.92	+
8	.51	2.001	.17	1.967	.65	30.13	34.098	.60	34.60	+
9	.51	2.001	.52	2.002	.27	29.77	33.770	.94	34.94	+
10	.57	2.007	.26	1.976	.78	30.28	34.263	.33	34.33	+
11	.60	2.010	.31	1.981	.15	29.65	33.640	.64	34.64	+
12	.89	2.039	.74	2.024	.99	30.49	34.553	.63	34.63	+
13	.58	2.008	.83	2.033	.44	29.94	33.980	.64	34.64	+
14	.59	2.009	.03	1.953	.62	30.12	34.082	.30	34.30	+
15	.16	1.966	.57	2.007	.87	30.37	34.343	.21	34.21	–
16	.36	1.986	.60	2.010	.82	30.32	34.316	.83	34.83	+
17	.70	2.020	.06	1.956	.12	29.62	33.596	.59	34.59	+
18	.46	1.996	.54	2.004	.04	29.54	33.540	.51	34.51	+
19	.99	2.049	.84	2.034	.81	30.31	34.393	.15	34.15	–
20	.36	1.986	.12	1.962	.54	30.04	33.988	.97	34.97	+

Out of 20 iterations of generated samples in 3 cases the clearance is negative. Thus the probability of parts not assembling is $3/20 \times 100 = 15\%$ or 0.15.

Example 2.9

A piece of equipment contains four identical tubes and can function only if all the four are in working order. The lives of tubes has approximately uniform distribution from 1000 to 2000 hours. The current maintenance practice is to replace a tube when it fails. Equipment has to be shut down for 1 hour for replacing a tube, the cost of one tube is Rs. 100, while the shut down time costs Rs. 200 per hour. Simulate the system for about 6000 hours of run and find the maintenance cost.

P.U.M.E. (Mech.), 1991

Solution: The piece of equipment has four identical tubes and comes to halt when any one fails. It is then replaced which takes one hour time and incurs Rs. 300 as cost (cost of tube and cost of down time). The lives of tube are uniformly distributed form 1000 to 2000. To keep the computations simple, one digit random numbers can be used to generate the tube lives. If R is the random number, then the life of a tube L is given by

$$L = 1000 + (2000 - 1000)\ R/10 = 1000 + 100 \times R$$

The random numbers can be taken from a random number table. Start with any one digit in the table and move in row or column continuously. One string of random numbers is,

8, 2, 6, 3, 1, 0, 8, 9, 2, 8, 3, 7, 4, 8, 5, 6, 0, 4, 9

The corresponding tube lives are

1800, 1200, 1600, 1300, 1100, 1000, 1800, 1900, 1200, 1800, 1300, 1700, 1400, 1800, 1500, 1600, 1000, 1400, 1900.

The simulation of the system is given in Table 2.11. The variable '*clock*' indicates the time elapsed, while *T*1, *T*2, *T*3 and *T*4 represent the tube lives, that is the time after which the tube will fail. The simulation starts at zero time, with all the four tubes new having lives 1800, 1200, 1600 and 1300. The shortest life is of Tube *T*2, which will fail after 1200 hours. The clock jumps to 1200 and time to failure of tubes *T*1, *T*3 and *T*4 is computed. *T*2 is replaced with a new tube with life of 1100 hours. Now out of *T*1, *T*2, *T*3, and *T*4 the next tube to foil is *T*4 with life of 100 hours. Advance the clock by 100 hours, update the time to failure of all the tubes, and replace *T*4 by a new tube, corresponding to next random number. If two tubes fail at the same time, both are replaced but the downtime remains same. In the stipulated simulation period of 6000 hours the system failed 14 times, and 15 tubes were replaced.

Maintenance cost = 15 × 100 + 14 × 200 = Rs. 3300/- only.

Table 2.11

Clock	*T1*	*T2*	*T3*	*T4*	*No. of tubes replaced*
0	1800	**1200**	1600	1300	1
1200	600	1100	400	**100**	1
1300	500	1000	**300**	1300	1
1600	**200**	700	1000	1000	1
1800	1800	**500**	800	800	1
2300	1300	1900	**300**	**300**	2
2600	**1000**	1600	1200	1800	1
3600	1300	600	**200**	800	1
3800	1100	**400**	1700	600	1
4200	700	1400	1300'	**200**	1
4400	**500**	1200	1100	1800	1
4900	1500	700	**600**	1300	1
5500	900	**100**	1600	700	1
5600	800	1000	1500	**600**	1
6200	200	400	900	–	–

2.4 Exercises

1. Make an irregular figure of known area, then determine its area by the Monte Carlo Method. Use the random number table for generating the points. Compare the accuracy of your result, when the total number of points is,

   (*i*) 50 (*ii*) 100 (*iii*) 200.

2. Solve the following integrals by Monte Carlo Method.

   (*i*) $I = \int_0^1 x\,dx$ (*ii*) $I = \int_1^5 \frac{x^4}{3}\,dx$

3. Determine the value of π by employing the Monte Carlo Method, and using the relation,

   Area of Circle = $\frac{\pi D^2}{4}$. Determine the values for 20, 40 and 100 pairs of random numbers.

4. Develop a computer program for determining the value of π and run the simulation for 1000 observations.

5. In the random walk problem of Example 2.5, the drunkard can take steps in four directions, forward, backward, to left and to right. The probabilities associated with these are 40%, 10%, 25% and 25%. The distances covered in the forward, backward, to left and to right steps are 75 cm , 45 cm, 60 cm and 60 cm respectively. Simulate the walk for 50 steps, and find the location at the end of 50 steps, while the starting point is (0, 0) on the *x-y* scales.
6. In the bombing problem, the ammunition depot area is a circle of 500 m radius. The point of impact is normally distributed around the aiming point with standard deviation of 500 m in *X*-direction and 300 m in the *Y*-direction. Simulate the bombing operation for 40 strikes and find the percentage of strikes on target.
7. The random variables *x*, *y* and *z* are distributed as under,

   $$x \sim N\ (\mu = 50,\ \sigma^2 = 65\ )$$
   $$y \sim N\ (\mu = 100,\ \sigma^2 = 100\ )$$
   $$z \sim N\ (\mu = 150,\ \sigma^2 = 150\ )$$

   Simulate 50 values of the random variate,

   $$W = (\ x + y\ )/z.$$

8. Solve the following integral by Monte Carlo Method.

   $$\int_2^5 x^3\,dx$$ **[PTU B. Tech (Prod.) May 2006]**

9. The random variables *X* and *Y* are distributed as follows.

   $Y \sim 10 \pm 10$ (uniform)
   $Y \sim 10 \pm 8$ (uniform)

   Simulate 20 values of the random variables

   (*i*) $z = xy$ (*ii*) $k = \frac{x}{y}$

10. In the reliability problem of Example 2.6, the bearing life is distributed as follows:

Hours	*Probability*
1200	.10
1400	.25
1600	.30
1800	.25
2000	.10

The remaining data is same. Simulate the system for the present and proposed policies of maintenance, and find the best policy.

11. Write a computer program in any language for simulating the reliability problem of Example 2.6 and run the program for 10,000,00 hours of operation.
12. Write a computer program in C language for simulating the Random walk problem of Example 2.5. Simulate the walk 10 times, each time for 500 steps. Use different stream of random numbers each time. Find the mean value of distance traveled.
13. Write a computer program for the simulation of hinge assembly problem of Example 2.8. Run the simulation for 1000 iterations and determine the probability of parts not assembling.

Chapter 3

SIMULATION OF CONTINUOUS SYSTEMS

3.1 Introduction

In Chapter 1, the classification of systems into continuous and discrete was discussed. In a continuous system, the state of the system changes continuously generally with time. But the time is not an essential parameter of a continuous system. In many situations, some parameter, other than time, is continuously changed to determine the behavior of the system. The recursive procedures for solving problems do not involve time, but are examples of continuous systems. Similarly, determining the value of π by simulation and solving an integral by simulation are examples of continuous systems. But many of the continuous systems, as the progress of a chemical reaction, flight of a missile or a servo system are time dependent, and may be called dynamic systems. Such systems are generally described by means of a set of differential equations. If the differential equations are ordinary, linear and time-invariant, their solution can easily be obtained analytically. However, when the equations are complex and more involved, having variable coefficients, their solutions can be obtained only by employing numerical methods, which involve a very large amount of computations, which can be carried out only by using computers. Simulation of continuous systems, is the process of carrying out the computations with a detailed insight into the system. There is hardly any difference between the numerical solution and a continuous simulation of a problem.

In this chapter, we will discuss some examples of continuous systems.

3.2 A Pure Pursuit Problem

In a pure pursuit problem, there is a target, which moves along a predetermined path, and there is a pursuer, who follows the target, redirecting itself towards the target at fixed intervals of time. The target moves along its predetermined path and does not make any effort to evade the pursuer. A fighter aircraft following an enemy bomber is an example of pure pursuit problem. The aircraft flies towards the bomber to catch up with it and to destroy it. The bomber continues to fly along a predetermined path and the fighter has to change its direction to keep pointing towards the bomber. Now, the typical question is whether the fighter will be able to destroy the bomber or not, if so in how much time. The answer depends upon a number of factors, out of which one is the path of the bomber. If it is a straight line, the problem can easily be solved by employing analytical techniques. But if the target follows a curved path, the solution of the problem becomes difficult, and simulation is the technique, which can be employed.

Let us consider the situation, where the target and pursuer fly in the same horizontal plane, and stay in that plane during the period of pursuit. The fighter's speed S is constant at 20 km/min, while the target's path is specified as a function of time as below:

Time (t)	0	1	2	3	4	5	6	7	8	9
$xb(t)$	100	100	120	129	140	149	158	168	179	188
$yb(t)$	0	3	6	10	15	20	26	32	37	34

Time (t) :	10	11	12	13	14	15
$xb\ (t)$:	198	209	219	226	234	240
$yb\ (t)$:	30	27	23	19	16	14

The fighter aircraft is at position xf, yf (0, 50) when it sights the bomber that is at time $t = 0$, the time at which pursuit begins. The fighter corrects its direction after a fixed interval of one minute, so as to point towards the bomber, and shoots the bomber by firing a missile as soon as it is within a distance of 10 km. The pursuit ends. In case the bomber is not shot within 12 minutes of the pursuit, the pursuit is abandoned.

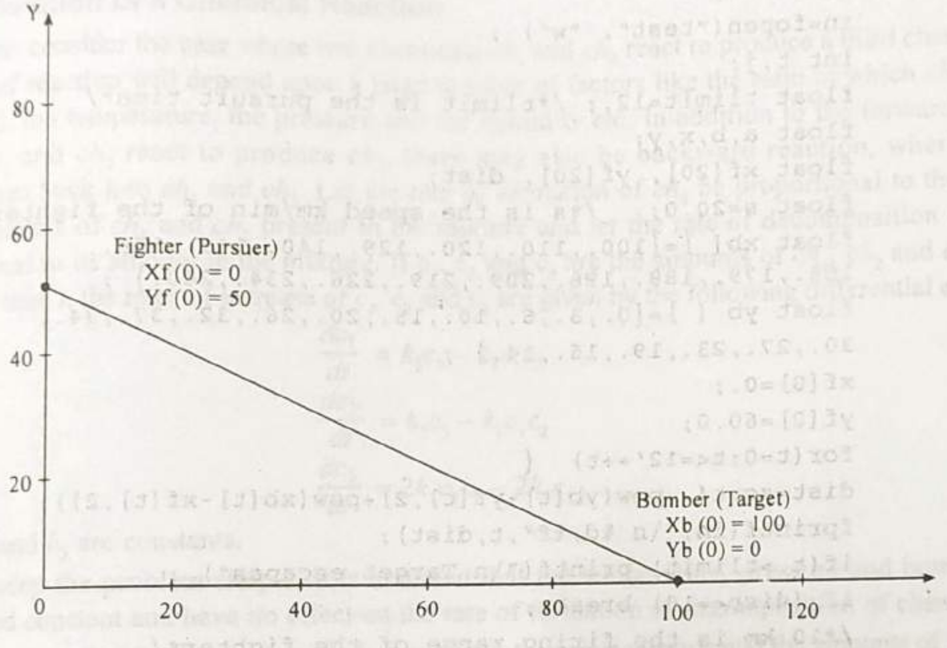


Fig. 3.1

The pursuit begins at time $t = 0$, when the bomber is at $(xb, yb) = (100, 0)$ and the fighter is at $(xf, yf) = (0, 50)$. The fighter aligns its velocity vector with the point of sight towards the bomber. It continues to fly in that direction for one minute and at time $(t + 1)$, it looks at the target again and realigns itself.

At a time t, the distance Dist (t) between fighter and the bomber is given by.

$$\text{Dist}\ (t) = \sqrt{\left(yb\ (t) - yf\ (t)\right)^2 + \left(xb\ (t) - xf\ (t)\right)^2}$$

If θ is the angle which the line connecting bomber and fighter makes with X-direction then,

$$\sin\theta = \frac{yb\ (t) - yf\ (t)}{Dist\ (t)};\quad \cos\theta = \frac{xb\ (t) - xf\ (t)}{Dist\ (t)}$$

The co-ordinates of the position of the fighter at time $(t + 1)$ can be determined as,

$$xf\ (t+1) = xf\ (t) + \text{S} \cos\theta$$

$$yf\ (t+1) = yf\ (t) + \text{S} \sin\theta$$

With these new coordinates of the fighter, its distance from the target is again computed. If the distance is 10 km or less, the pursuit ends, otherwise the values of $\sin\theta$, $\cos\theta$ and the new co-ordinates of the fighter are recomputed. If the pursuit does not end within the specified time, the bomber is considered as escaped and pursuit is abandoned.

A computer program for this simulation written in C language is given below :

```
#include<stdio.h>
#include <math.h>
void main ( )
   {
            /*Pursuit-Simulation of a pursuit problem.*/
            /* define and inialise variables*/
            FILE *in;
            in=fopen("test", "w") ;
            int t,j;
            float tlimit=12.; /*tlimit is the pursuit time*/
            float a,b,x,y;
            float xf[20], yf[20], dist;
            float s=20.0;   /*s is the speed km/min of the fighter*/
            float xb[ ]={100.,110.,120.,129.,140.,149.,158.,
            168.,179.,188.,198.,209.,219.,226.,234.,240.} ;
            float yb [ ]={0.,3.,6.,10.,15.,20.,26.,32.,37.,34.,
            30.,27.,23.,19.,16.,14.} ;
            xf[0]=0.;
            yf[0]=60.0;
            for(t=0;t<=12'++t)   {
            dist=sqrt(  pow(yb[t]-yf[t],2)+pow(xb[t]-xf[t],2));
            fprintf(in,"\n %d,%f",t,dist);
            if(t >tlimit) printf("\n Target escapes") ;
            if (dist<=10) break ;
           /*10 km is the firing range of the fighter*/
           else {
                  xf[t+1]=xf[t]+s*(xb[t]-xf[t])/dist;
                  yf[t+1]=xf[t]+s*(yb[t]-yf[t])/dist;
           }}
         fprintf(in,"\n Persuit ends,shot at %d minutes and at
         %5.2f kms.",t,dist);

         fprintf(in,"\n vlaue of s");
         scanf("%f",s);
         fclose(in);
```

The output of this program for the given data is,

0, 116.619041

1, 103.937393

2, 91.803185

3, 78.942284

4, 68.461441

5, 56.794758

6, 45.858341
7, 36.668800
8, 28.603636
9, 17.205299
10, 8.210612

Pursuit ends, shot at 10 minutes and at 8.21 km.

3.3 Simulation of a Chemical Reaction

Let us consider the case where two chemicals ch_1 and ch_2 react to produce a third chemical ch_3. The rate of reaction will depend upon a large number of factors like the ratio in which ch_1 and ch_2 are mixed, the temperature, the pressure and the humidity etc. In addition to the forward reaction where ch_1 and ch_2 react to produce ch_3, there may also be backward reaction, where by, ch_3 decomposes back into ch_1 and ch_2. Let the rate of formation of ch_3 be proportional to the product of the amounts of ch_1 and ch_2 present in the mixture and let the rate of decomposition of ch_3 be proportional to its amount in the mixture. If c_1, c_2 and c_3 are the amounts of ch_1, ch_2 and ch_3 at any instant of time t, the rates of increase of c_1, c_2 and c_3 are given by the following differential equations:

$$\frac{dc_1}{dt} = k_2c_3 - k_1c_1c_2$$

$$\frac{dc_2}{dt} = k_2c_3 - k_1c_1c_2$$

$$\frac{dc_3}{dt} = 2k_1c_1c_2 - 2k_2c_3$$

where k_1 and k_2 are constants.

To keep the problem simple, it is assumed that the temperature, pressure and humidity are maintained constant and have no effect on the rate of formation or decomposition of chemicals.

As soon as the chemicals ch_1 and ch_2 are mixed, the reaction starts and the amounts of c_1, c_2 and c_3 in the mixture go on changing as the time progresses. The simulation of the reaction will determine the state of the system, that is the value of quantities c_1, c_2 and c_3, at different points in time. Starting at zero time, a very small increment of time is taken in each step. It is assumed to be so small, that all changes in the mixture can be taken to occur at the end of each increment. If $c_1(t)$, $c_2(t)$ and $c_3(t)$ are the quantities of the three chemicals at time t, then at time $t + \Delta t$, the quantities are;

$$c_1(t + \Delta t) = c_1(t) + \frac{dc_1(t)}{dt}\,\Delta t$$

Identical equations can be written for

$c_2(t + \Delta t)$ and $c_3(t + \Delta t)$.

Taking $c_1(0)$, $c_2(0)$ and $c_3(0)$ as the quantities of ch_1, ch_2 and ch_3 at time zero, that is when the reaction starts the state of the system at time Δt will be,

$$c_1(\Delta t) = c_1(0) + [k_2c_3(0) - k_1c_1(0)\cdot c_2(0)]\,\Delta t$$

$$c_2(\Delta t) = c_2(0) + [k_2c_3(0) - k_1c_1(0)\cdot c_2(0)]\,\Delta t$$

$$c_3(\Delta t) = c_3(0) + [2k_1c_1(0)\cdot c_2(0) - 2k_2c_3(0)]\,\Delta t$$

Using the state of the system at time $2\Delta t$, the state of the system at time $3\Delta t$ and so on will be computed. This will continue till the specified time of reaction T is reached. At each increment of time, we can either count the number of steps taken or check the attained time with the prescribed simulation run time.

A computer program in C language for the simulation of above said chemical reaction is given below.

```
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
main()
    {       /*   Simulation of a chemical reaction*/
            int i,n,k;
             float c1[200],c2[200],c3[200];
             float k1=0.025,k2=0.01,delta=0.1;
            c1[0]=50.0,c2[0]=25.0,c3[0]=0.0;
            float t=0.0, time=8.0;

            i=0
            printf("\n Time    C1      C2      C3  ");
            while(t<=time)  {
            printf("\n%5.2f %6.2f %6.2f %6.2f  ",t,c1[i],c2[i],c3[i]);
            c1[i+1]=c1[i]+(k2*c3[i]-k1*c1[i]*c2[i])*delta;
            c2[i+1]=c2[i]+(k2*c3[i]-k1*c1[i]*c2[i])*delta;
            c3[i+1]=c3[i]+2.0*(k1*c1[i]*c2[i]-k2*c3[i])*delta;
            i=i+1;
            t=t+delta;
            if(t >= 2.0) delta=0.2;
            if(t >= 6.0) delta=0.4;
            }
            printf("\n Any digit");
            scanf("%d",k);
    }
```

The output of this program for the given values of constants and initial values of variables, for a reaction of 15 minutes, is given below. It can be noticed that as the time progresses, the changes in the mixture become slow. It may not be of much use to continue the reaction beyond a certain time.

Chemical Reactor

$k_1 = 0.025$ $\quad k_2 = 0.01$ $\quad$ time = 15.00

$c_1[0] = 50.0$ $\quad c_2[0] = 25.0$ $\quad c_3[0] = 0.0$

Time	c_1	c_2	c_3
0.00	50.00	25.00	0.00
0.1	46.88	21.88	6.25
0.2	44.32	19.32	11.36
0.3	42.19	17.19	15.62
0.4	40.39	15.39	19.22
0.5	38.86	13.86	22.29
0.6	37.53	12.53	24.93
0.7	36.38	11.38	27.24

0.8	35.37	10.37	29.25
0.9	34.49	9.49	31.03
1.0	33.70	8.70	32.60
1.1	33.80	8.00	34.00
1.2	32.37	7.37	35.25
1.3	31.81	6.81	36.38
1.4	31.31	6.31	37.39
1.5	30.85	5.85	38.30
1.6	30.44	5.44	39.13
1.7	30.06	5.06	39.88
1.8	29.72	4.72	40.56
1.9	29.41	4.41	41.18
2.0	29.13	4.13	41.74
2.2	28.61	3.61	42.78
2.4	28.18	3.18	43.64
2.6	27.82	2.82	44.36
2.8	27.52	2.52	44.97
3.0	27.26	2.26	45.48
3.2	27.04	2.04	45.92
3.4	26.86	1.86	46.28
3.6	26.70	1.70	46.60
3.8	26.57	1.57	46.87
4.0	26.45	1.45	47.09
4.2	26.35	1.35	47.29
4.4	26.27	1.27	47.46
4.6	26.20	1.20	47.60
4.8	26.14	1.14	47.73
5.0	26.08	1.08	47.83
5.2	26.04	1.04	47.92
5.4	26.00	1.00	48.00
5.6	25.97	0.97	48.07
5.8	25.94	0.94	48.13
6.0	25.91	0.91	48.18
6.2	25.89	0.89	48.22
6.6	25.85	0.85	48.30
7.0	25.82	0.82	48.35
7.4	25.81	0.81	48.39
7.8	25.79	0.79	48.42
8.2	25.78	0.78	48.44
9.0	25.77	0.77	48.46
10.20	25.76	0.76	48.48
12.20	25.75	0.75	48.49
15.00	25.75	0.75	48.49

3.4 The Runge-Kutta Method

This method of solving the differential equations was devised by C. Runge about the year 1894 and was extended by W. Kutta in 1901. It is a method designated to approximate the Taylor series without having to evaluate the higher order derivatives. In a simpler Euler's formula, the value of variable y at $(i+1)$th instant is estimated in the form of its value and shape at ith instant. That is,

$$y_{i+1} = y_i + \frac{dy}{dt} \cdot \Delta t.$$

But in Runge-Kutta method, the first derivative $f(t, y)$ of y is computed at several carefully chosen points in the interval $(t, t + \Delta t)$, and these values are combined in such a way as to obtain good accuracy in the computed increment $y_{i+1} - y_i$. Following this technique, a family of formulas have been derived, which are known as Runge-Kutta formulas. The most commonly used is the fourth order Runge-Kutta formula. If $\frac{dy}{dx} = f(x, y)$ represents any first order equation and h denotes the interval between equally distant values of x, then with the initial values x_0, y_0, the first increment in y is computed from the formula, which is a set of equations given below.

$$k_1 = f(x_0, y_0)\, h,$$

$$k_2 = f\left(x_0 + \frac{h}{2}, y_0 + \frac{k_1}{2}\right) h,$$

$$k_3 = f\left(x_0 + \frac{h}{2}, y_0 + \frac{k_2}{2}\right) h,$$

$$k_4 = f(x_0 + h, y_0 + k_3)\, h,$$

$$\Delta y = \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

Then

$$x_1 = x_0 + h$$
$$y_1 = y_0 + \Delta y$$

The increment in y for the second interval is computed in a similar manner by means of the formula,

$$k_1 = f(x_1, y_1)\, h,$$

$$k_2 = f\left(x_1 + \frac{h}{2}, y_1 + \frac{k_1}{2}\right) h,$$

$$k_3 = f\left(x_1 + \frac{h}{2}, y_1 + \frac{k_2}{2}\right) h,$$

$$k_4 = f(x_1 + h, y_1 + k_3)\, h,$$

$$\Delta y = \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4),$$

and so on for the succeeding intervals.

The derivation and any discussion of the Runge-Kutta formulas is beyond the scope of this book. Any book on Numerical Analysis may be referred for further details of this formula. To illustrate the use of this formula in continuous simulation, let us consider the simulation of an amplifier circuit.

3.5 Simulation of an Amplifier Circuit

Fig. 3.2 shows an electronic circuit, where 'Amp' is a voltage amplifier with amplification factor A. The input impedance to the amplifier is infinite, while output impedance is zero. The input capacitor C_1 is a coupling capacitor of large value, $C_1\,\mu f$. Output capacitor C_2 is of small size, $c_2\mu f$; R_1, R_2, R_3 are resistances.

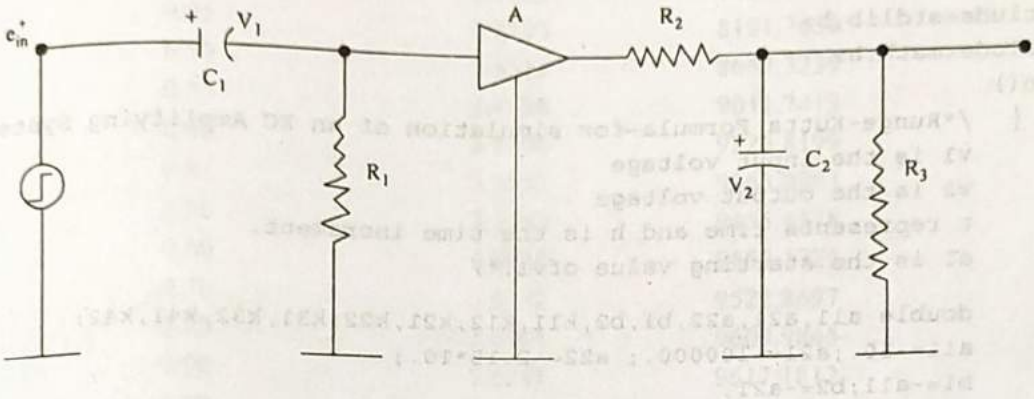


Fig. 3.2

The currents entering the capacitors C_1 and C_2 are given by the equations.

$$C_1 \frac{dv_1}{dt} = (e_{in} - v_1)\frac{1}{R_1}$$

$$C_2 \frac{dv_2}{dt} = (e_{in} - v_1)\frac{A}{R_2} - \frac{v_2\,(R_2 + R_3)}{R_2 R_3}$$

where e_{in} is the input voltage.

These equations can be written as:

$$\frac{dv_1}{dt} = \frac{1}{R_1 C}(e_{in} - v_1)$$

$$\frac{dv_2}{dt} = \frac{A}{R_2 C_2}(e_{in} - v_1) - \frac{(R_2 + R_3)}{R_2 R_3 C_2} \cdot V_2 \quad \text{or} \quad \frac{dv_1}{dt} = A_{11} v_1 + B_1 e_{in}$$

$$\frac{dv_2}{dt} = A_{21} v_1 + A_{22} v_2 + B_2 e_{in}$$

where constants $A_{11} = \dfrac{1}{R_1 C_1} = -B_1$

$$A_{21} = \frac{A}{R_2 C_2} = -B_2$$

$$A_{22} = \frac{R_1 + R_2}{R_1 R_2 C_3}$$

Let us take the values of constants as under:

$$A_{11} = -50 \text{ sec}^{-1}$$
$$A_{21} = -10{,}000 \text{ sec}^{-1}$$
$$A_{22} = 21.5 \text{ sec}^{-1}$$

For simulating the system, let us assume the starting conditions, as time $t = 0$, $v_1 = 0$, $v_2 = 0$. The voltage applied at the input terminal $e_{in} = 1.5$ for $t \geq 0$ and $e_{in} = 0$ for $t < 0$.

The simulation program written in C language is given below.

```
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
main()
  {  /*Runge-Kutta Formula—for simulation of an RC Amplifying System
     v1 is the input voltage
     v2 is the output voltage
     t represents time and h is the time increment.
     e1 is the starting value ofv1.*/

     double a11,a21,a22,b1,b2,k11,k12,k21,k22,k31,k32,k41,k42;
     a11=-50.;a21=-100000.; a22=-2.15*10.;
     b1=-a11;b2=-a21;
     double t=0.,v1=0., v2=0.;
     double h=.05,e1=1.5;
     int i,k;
     int n=10;
     printf("\n a11=%8.2f a21=%8.2f  a22=%8.2f  h=%4.2f e1=%4.2f
     ",a11,a21,a22,h,e1);
     printf("\n i   Time(u sec.)   Voltage,v1   Voltage v2");

     for(i=1;i<=n;++i) {
     k11=h*((a11*v1)+e1);
     /*printf("\n %6.2f",k11);*/
     k12=h*((a21*v1)+(a22*v2)+b2);
     k21=h*(a11*(v1+.5*k11)+b1);
     k22=h*(a21*(v1+.5*k11)+a22*(v2+.5*k12)+b2);
     k31=h*(a11*(v1+.5*k21)+b1);
     k32=h*(a21*(v1+.5*k21)+a22*(v2+.5*k22)+b2);
     k41=h*(a11*(v1+k31)+b1);
     k42=h*(a21*(v1+k31)+a22*(v2+k32)+b2);
     /*printf("\n %6.2f %6.2f %6.2f %6.2f %6.2f
     %6.2f",k11,k12,k21,k22,k31,k32);*/
     v1=v1+(k11+2.*k21+2.*k31+k41)/6.;
     v2=v2+(k12+2.*k22+2.*k32+k42)/6.;
     t=t+h;
     printf("\n %d %8.2f     %12.4f   %12.4f",i,t,v1,v2);
     }
     printf("\n any digit");
     scanf("%d",k);
     }
```

The output of this program is as under.

$a_{11} = -50.00$ $\quad a_{21} = -10,0000.00$ $\quad a_{22} = -21.50$

$h = 0.05$ $\quad e_1 = 1.50$

Time (μ/sec)	*Voltage* (v_1)	*Voltage* (v_2)
0.050	1.2736	2465.5143
0.10	2.1034	4659.2401
0.15	2.6349	6290.8198
0.20	2.9821	7422.2709
0.25	3.2173	8181.7658
0.30	3.3533	8683.3239
0.35	3.4580	9011.7413
0.40	3.5194	9225.8199
0.45	3.5592	9365.0303
0.50	3.5750	9455.4378
0.60	3.6026	9552.1725
0.70	3.6142	9522.8697
0.80	3.6191	9609.9845
0.90	3.6211	9617.1812
1.00	3.6220	9620.2072
1.20	3.6225	9622.0146
1.40	3.6226	9622.3582
1.50	3.6226	9622.3739

3.6 A Serial Chase Problem

In this chase problem, there are a number of moving objects, which chase each other in a serial order like A chases B; B chases C and C chases D etc. Each object moves towards its target unmindful of the fact that it itself is being targeted by some other object. Depending upon the original location, and speed of motion of the objects, each object will take its own time to hit its target. As soon as a hit occurs, the chase ends.

Let us consider the objects A, B, C and D located at the four corners of a square field. A is to kill B, B is to kill C and C is to kill D. The velocities of A, B, C and D are *va*, *vb*, *vc* and *vd* respectively. Let θ*a* be the angle which the velocity vector of A makes with the direction AB. Similarly let θ*b* be the angle which velocity vector of B makes with the direction BC. Though C and D move in straight lines in the present case, to generalize the problem, let θ*c* and θ*d* be the angles, which velocity vectors of C and D make with the direction CD. As A runs towards B, it continuously changes its direction so that its velocity vector is in the direction of B. Similarly B continuously changes its direction, so that its velocity vector is always in the direction of its target C. Since D is not chasing any object, it will move in a straight line, so as to keep maximum distance from its hitter C. Object C has to run towards D. This again will move in a straight line, as its target is running straight. Which of the object will be hit first ? is to be determined. This can be determined by simulating the chase. The location of all the four objects is determined after every small increment in time. The direction of motion at new points is determined. The distances between objects are computed. As soon as the distance between any set of objects that is distance AB, BC or CD becomes zero, hit occurs and chase ends. It can be assumed that when the distance becomes very small, say less than 0.005, hit is taken to occur.

The simulation program, written in C language is given below.

Following values for velocities and initial locations of the objects have been taken.

$Va = 35$ km/hr

$Vb = 25$ km/hr

$Vc = 15$ km/hr

$Vd = 10$ km/hr

Location of A is (10, 10), of B is (30, 10), of C is (30, 30) and of D is (10, 30).

```
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
main()
{ /*It is a serial chase problem. Four animals A,B,C,D are there
  in a field. A chases B, B chases C and C chases D.as soon as
  the target is within say 0.005km, it is killed and the chase
  ends.
  Speeds of A,B,C,D are given say 35,25,15 and10 km /hour.
  The initial positions are also fixed.It is assumed that A will
  always run in the direction of B and B straight towards C and C
  straight towards D. D runs straight away from C in first case
  and towards A in the second case.
  xa and xb are the coordinates of A.
  thab is the angle with direction ab.
  dab is the distance between A and B.
  va is the velocity of A.
  */
  float xa=10.,ya=10.,xb=30.,yb=10.,xc=30.,yc=30.,xd=10.,yd=30.;
  float thab=0.,thbc=0.,thcd=0.,thd=0.;
  float dab,dbc,dcd,t,delt,k;
  float va=35.,vb=25.,vc=15.,vd=10.;
  t=0.;delt=.001;
  dab=sqrt((xa-xb)*(xa-xb)+(ya-yb)*(ya-yb));
  dbc=sqrt((xb-xc)*(xb-xc)+(yb-yc)*(yb-yc));
  dcd=sqrt((xc-xd)*(xc-xd)+(yc-yd)*(yc-yd));
  while(t<= 1.70) {
  xa=xa+va*delt*(xb-xa)/dab;
  ya=ya+va*delt*(yb-ya)/dab;
  xb=xb-vb*delt*(xb-xc)/dbc;
  yb=yb+vb*delt*(yc-yb)/dbc;
  xc=xc-vc*delt*(xc-xd)/dcd;
  yc=yc-vc*delt*(yc-yd)/dcd;
  xd=xd;
  yd=yd-vd*delt;
  dab=sqrt((xa-xb)*(xa-xb)+(ya-yb)*(ya-yb));
  dbc=sqrt((xb-xc)*(xb-xc)+(yb-yc)*(yb-yc));
  dcd=sqrt((xc-xd)*(xc-xd)+(yc-yd)*(yc-yd));
```

```
printf("\nt=%6.4f   A=%5.2f,%5.2f   B=%5.2f,%5.2f  C=%5.2f,%5.2f
D=%5.2f,%5.2f",
t,xa,ya,xb,yb,xc,yc,xd,yd);
/* printf("\nt=%6.3f dab=%6.2f  dbc=%6.2f
dcd=%6.2f",t,dab,dbc,dcd);
*/
t=t+delt;
if(dab<=0.005) {
t=4.0; printf("\n B killed, chase ends");}
if(dbc<=0.005) {
t=4.0; printf("\n C killed,chase ends");}
if(dcd<=0.005) {t=4.0; printf("\n D killed,chase ends");}
}
printf("\n any digit");
scanf("%d",k);
}
```

For the given data, the path traced by the four objects will be as in Fig. 3.3. The chase ends at 0.897 hrs, when object A hits object B, at location X. (19.87, 28.57). At that instant object C is at C′ and D is at D′. The location of the four objects at different points in time is given in the following table.

A		*B*		*C*		*D*		*T*
x	*y*	*x*	*y*	*x*	*y*	*x*	*y*	
10.00	10.00	30.00	10.00	30.00	30.00	10.00	30.00	0.000
12.69	10.13	29.94	11.92	28.85	30.00	9.23	30.00	0.076
13.18	10.19	29.92	12.27	28.64	30.00	9.09	30.00	0.090
13.49	10.23	29.90	12.50	28.50	30.00	9.00	30.00	0.099
13.84	10.28	29.88	12.75	28.35		8.90		0.109
14.18	10.34	29.86	13.00	28.20		8.80		0.119
14.53	10.40	29.83	13.24	28.05		8.70		0.129
15.21	10.54	29.78	13.74	27.75		8.50		0.149
16.23	10.79	29.67	14.48	27.30		8.20		0.179
18.55	11.59	29.55	16.20	26.25		7.50		0.249
21.61	13.33	28.67	18.63	24.74		6.49		0.350
24.06	15.81	27.71	20.94	23.25		5.50		0.450
25.51	18.97	26.45	23.10	21.74		4.49		0.550
25.51	22.44	24.89	29.05	20.24		3.49		0.650
24.92	24.08	24.00	25.92	19.49		2.99		0.700
23.98	25.55	23.04	26.72	18.74		2.49		0.750
22.76	26.81	22.01	27.44	17.99		1.99		0.800
21.34	27.82	20.95	28.06	17.24		1.49		0.850
19.87	28.57	19.87	28.57	16.50		1.02		0.897

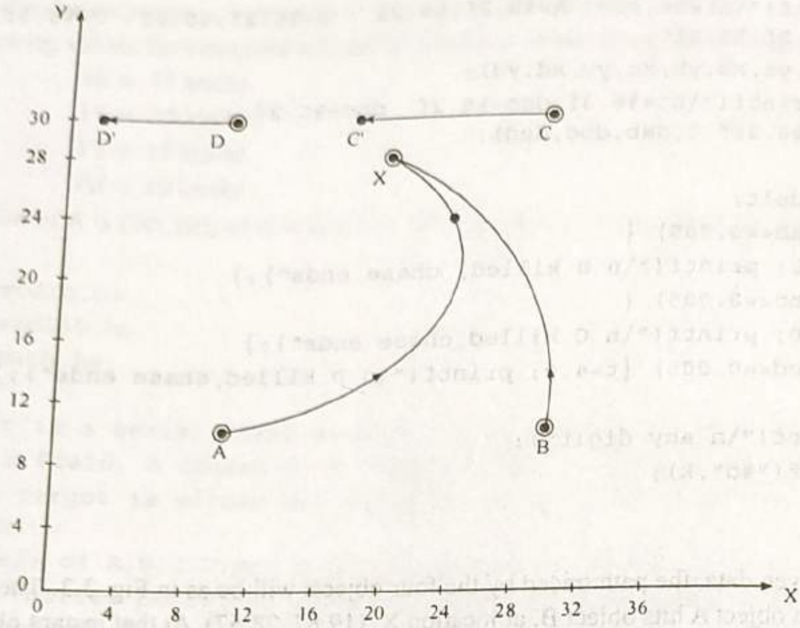


Fig. 3.3

D moves towards A

If the object D decides to run towards point A, its hitter object C will move in a curved path and may be it is hit by the object B. When the simulation is run with this modification, keeping all other data same, the paths traced by the four objects will be as shown in Fig. 3.4. The chase ends, when object A hits its target B; at location X (19.52, 26.64) and at time 0.87 hours. Thus, as compared to the earlier case, the chase ends in a lesser time.

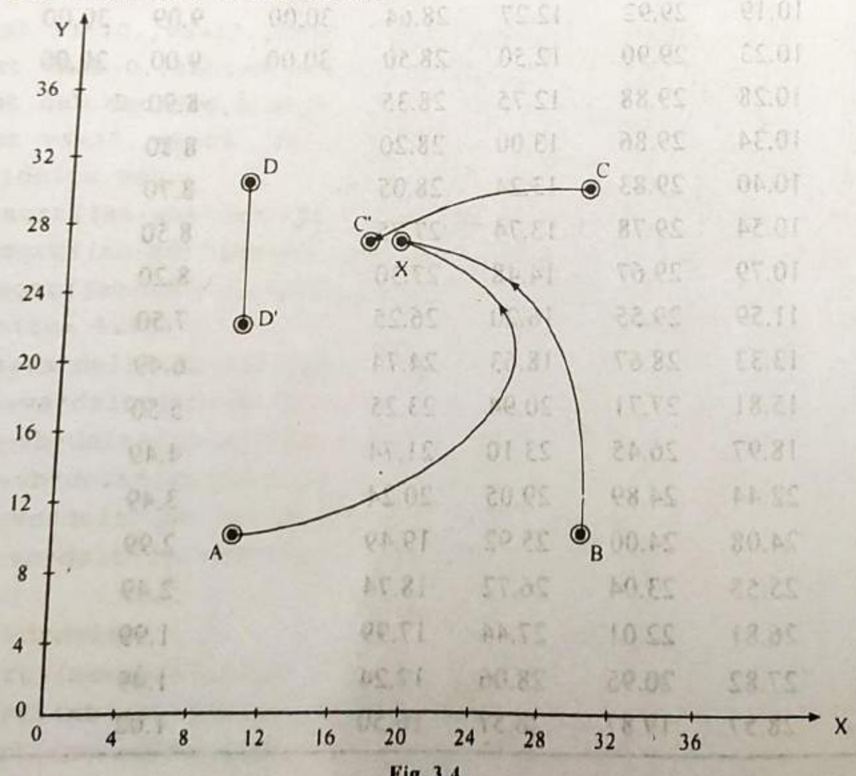


Fig. 3.4

3.7 Simulation of Exterior Ballistics

The exterior ballistics is the science, which deals with the motion of a projectile after it leaves the gun. The projectile is the material particle, which is acted upon by the force of gravity, by a tangential drag and the retarding force due to the resistance of air. In the simplest case, the acceleration due to gravity is assumed to be constant in magnitude and direction, which means that we are assuming a flat earth and that the projectile does not reach a great height. The air resistance is taken proportional to some power of velocity, like kv^n, where k is a constant. The value of n, which further depends upon velocity, can be taken as,

$0 < v < 790$ m/sec;	$n = 2$
320	3
410	5
450	3
600	2
860	1.7
1200	1.55

In simple cases, the problem can be analyzed to form differential equations, which can further be solved by some numerical method. In case of complicated problems, the solution of differential equations becomes difficult and simulation can be employed to solve the problems.

Let a projectile of mass m be fired with an initial velocity and an angle of elevation ϕ. Let v denote the velocity of projectile at any point in its path and let θ denote the inclination of the velocity vector at that point. Let kv^n be the drag due to air resistance, and ρ the radius of curvature of the trajectory at that point. (Fig. 3.5)

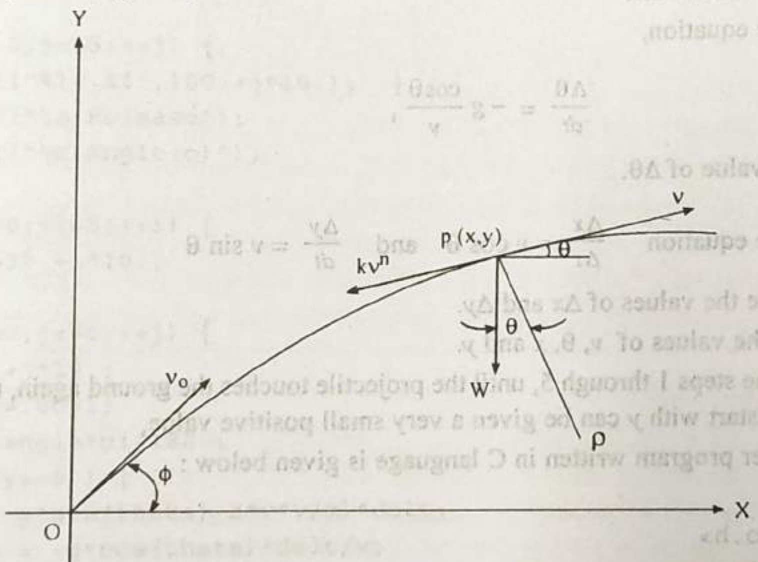


Fig. 3.5

Resolving the forces along the tangent and the normal at point p. We obtain:

$$m\frac{dv}{dt} + mg \sin\theta + kv^n = 0 \qquad ...(i)$$

$$m\frac{v^2}{\rho} + mg \cos\theta = 0 \qquad ...(ii)$$

If at that point $p\ (x, y)$, ds is the distance moved in a small time interval dt and $d\theta$ is the change in angle θ,

$$\rho = \frac{ds}{d\theta} \quad \text{and} \quad v = \frac{ds}{dt}.$$

Equation (*ii*) can be written as

$$mv\frac{d\theta}{dt} + mg\cos\theta = 0$$

Also we can write $$\frac{dx}{dt} = v\cos\theta$$

and $$\frac{dy}{dt} = v\sin\theta$$

The simulation of the projectile trajectory, will be the determination of a large number of points on its path. A very small interval of time Δt is taken, and the velocity and θ are assumed to remain constant over this small duration. Smaller the value of Δt, more accurate will be the simulation results. The simulation will proceed through the following steps:

1. Start with initial values $v = v_0$, $\theta = \phi$, $x = 0$, $y = 0$ and initialize k, m and n etc.
2. From the equation,
$$\frac{\Delta v}{\Delta t} = -g\sin\theta - \frac{kv^n}{m}$$
find the value of Δv.
3. From the equation,
$$\frac{\Delta\theta}{dt} = -g\frac{\cos\theta}{v},$$
find the value of $\Delta\theta$.
4. From the equation $\dfrac{\Delta x}{\Delta t} = v\cos\theta$ and $\dfrac{\Delta y}{dt} = v\sin\theta$
determine the values of Δx and Δy.
5. Update the values of v, θ, x and y.
6. Repeat the steps 1 through 5, until the projectile touches the ground again, that is y becomes zero. To start with y can be given a very small positive value.

The computer program written in C language is given below :

```
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
main()
      {  /* Projectile Trajectory simulation
      Release velocity (v) changes from 100 to 150m/sec in steps of 10
m/sec.
      Angle of release (angle) changes from 30 to 80 degree in steps of
10 degree.
```

```
    Mass of the projectile(m) =30 kg.,
    Coefficient of air resistance(c)=.001
    theta: angle in radians,
    delv: change in velocity in one iteration,
    delt: change in angle in one iteration,
    delx, dely and dtheta: changse in values of x, y and theta
                when time changes through delt,

    */
    float v=100.,angle,theta,delv,pi=22./
    7.,m=30.,c=.001,delt=.01,g=9.81;
    float x,y;
    float vel[10] ,range[10];
    float delx,dely,dtheta,xx,yy;
    int i,j,k;
    printf("\n Enter the value of mass m");
    scanf("%f",&m);

    printf("\n                    RANGE TABLE");
    printf("\n              Mass of projectile=%5.2fKg",m);
    printf("\n              Release Velocity(m/sec.) \n");
    printf("\n              ----------------------------------------------------
-\n ");

    for(j=0;j<=5;++j) {
    printf("%10.2f",100.+j*10.); }
    printf("\n Release");
    printf("\n Angle(o)");

    for(i=0;i<=5;++i) {
    angle=30.+i*10.;

    for(j=0;j<=5;++j) {
    v=100.+j*10.;
    x=0.;y=.0001;
    theta=angle*pi/180.;
    while(y>=0.) {
    delv=(-g*sin(theta)-c*v*v/m)*delt;
    dtheta = -g*cos(theta)*delt/v;
    delx=v*cos(theta)*delt;
    dely=v*sin(theta)*delt;

    v+=delv;
    theta+=dtheta;
    x+=delx;
    y+=dely;
    /*printf("\n %5.2f %5.2f %5.2f %5.2f",v,theta,x,y); */
```

```
    }
    range[j]=x;
    }
    printf("\n\n %4.1f",angle);
    for(j=0;j<=5;++j) {
    printf("%10.2f",range[j]);  }

    }
    }
```

The program can be further modified to find different parameter of the projectile, as the maximum height reached, the distance traveled and the time of flight. A range table for the above projectile, with θ varying from 30° to 80° and velocity varying from 100 to 150 m/sec and range in meters is given below:

Range Table

Mass of Projectile = 30.00 kg

Release Velocity (m/sec)

Release Angle	*100*	*110*	*120*	*130*	*140*	*150*
30	866.18	1043.64	1235.99	1442.74	1665.32	1899.93
35	937.00	1198.65	1336.01	1559.50	1797.63	2050.94
40	980.20	1179.72	1396.40	1628.88	1877.52	2140.80
45	993.84	1195.35	1411.88	1649.55	1900.39	2165.94
50	976.95	1175.91	1390.44	1621.30	1867.19	2128.35
55	931.29	1120.59	1325.38	1545.08	1779.07	2027.56
60	857.94	1032.20	1220.72	1423.01	1638.49	1866.62
65	758.38	912.52	1079.35	1258.39	1448.63	1650.64
70	635.84	765.00	905.20	1055.31	1215.76	1385.29
75	494.04	594.48	703.54	820.69	945.27	1077.31
80	336.97	405.68	480.11	560.28	645.61	735.83

3.8 Analog Simulation

The analog simulation in general, is the simulation performed by using the analog models of the system. There are many examples of physical analog models, where the properties of the system are represented by a set of another properties, and an analogy between the two is established. However, when we talk of the system simulation, the simulation carried on by employing the analog computers is called the analog system simulation. Simulation with an analog computer is based on a mathematical model, than being on a physical model. The electronic analog computers came into existence much before the digital computers, and were extensively used for simulating the engineering systems, which were too complex to be handled by the analytical techniques.

In analog simulation, various mathematical functions and operations are represented by some hardware devices whose behavior is equivalent to the mathematical functions like addition, multiplication or integration. The most widely used form of analog computer is the electronic analog computer, based on the use of high gain direct current amplifiers, called operational amplifiers.

Voltages in the computer are equated to mathematical variables, and the operational amplifiers can add or integrate the voltages. In each simulation, depending upon the system being simulated, some amount of hardware devices is required. These hardware elements are interconnected to imitate the system. The elements performing mathematical operations like integration, multiplication, addition, subtraction, square root etc. can be expressed as blocks, which are like subroutines. Any continuous system, can be simulated by suitably combining together a number of blocks. Different systems will have different number of blocks and their interconnections. Since, the same blocks can be organized in different ways, it is advantageous to have packages of standard subroutines to perform the operation of blocks. Hence, it is more convenient to have a special purpose block oriented programming language. A large number of such continuous system simulation languages have been designed and implemented. However, the discussion of these languages, is beyond the scope of this book.

3.9 Disadvantages of Analog Simulation

Though analog computers came into existence much before the digital computers, and have played a major role in the simulation of continuous dynamic systems, but they are giving way to digital computers at a fast pace. Following are some of the important disadvantages of analog computers.

1. *Limited accuracy*: The results obtained from analog simulation has limited accuracy, while much more accurate results can be obtained by employing digital simulation. Thus, where, high accuracy in results is required, as in space vehicles, guided missiles and fusion, the analog simulation cannot be used.
2. *Magnitude scaling*: In analog simulation, values of various variables are represented by voltages, which have a fixed and limited range. The variables must remain within the limited range, otherwise the results become inaccurate. Thus, all program variables have to be scaled, so that none exceeds the voltage range. This is a very difficult task, especially when the number of variables is large, and the range of their variation is not known. On the other hand, in digital computers, the problem of magnitude scaling does not arise, as they have a very large range. With floating point arithmetic, they have a very large precision. Hence, no magnitude scaling is normally required in digital simulation.
3. *Hardware set up required*: In analog simulation, hardware elements have to be combined to simulate the system, they have to be tested and calibrated while no such set up is required in digital simulation. Switching from one simulation to other takes time in analog simulation, while it requires no time in digital simulation. A simulation program on a digital computer can be easily stored, for reuse.

Analog simulation is considered to have two advantages over digital simulation. One is higher speed of solution, and second is immediate display of simulation results. However, these two advantages are also now becoming historical with the advent of super speed digital computers having on-line CRT displays.

3.10 Exercises

1. Solve the following employing continuous simulation

$$I = \int_0^{\pi} \sin x dx.$$

2. A bullet is fired at an angle of 40° with horizontal and with an initial velocity of 350 meter/sec. Assuming that the air resistance varies as the square of the velocity of the bullet and that the resistance coefficient is –0.00005, find the range, time of flight, and the angle of fall of the bullet.

3. Simulate the Pure Pursuit Problem (Fighter Bomber Problem). The initial position of the fighter is (0, 50) and the bomber in following the path :

Time (t) :	0	1	2	3	4	5	6	7	8	9	10	11	12
$xb(t)$:	80	90	99	108	116	125	133	141	151	160	169	179	180
$yb(t)$:	0	−2	−5	−9	−15	−18	−23	−29	−28	−25	−21	−20	−17

Assume the speed of the fighter as constant at 20 km/min. Calculate whether the fighter will hit the bomber (the time for chase is 10 min). The range of the fighter to hit the bomber is 10 km. Specify any other assumptions you made. **[PTU, B.Tech (Elect.) 3rd Sem. 2002]**

4. A cannon fires spherical balls of mass m, a range table that gives ranges for various values of muzzle velocity and gun elevation (firing angle) is to be produced. The sensitivity of range to small changes in muzzle velocity and gun elevation is also to be investigated. Assume that the drag is proportional to the square of the instantaneous velocity of the cannon ball, and it is purely along the direction of flight. Making use of the following four equations, write a computer program to produce the range table:

$$m\frac{dy}{dt} + mg \sin \theta + cv^2 = 0$$

$$mv\frac{d\theta}{dt} + mg \cos \theta = 0$$

$$\frac{dx}{dt} = v \cos \theta$$

$$\frac{dy}{dt} = v \sin \theta$$

Constant c is the drag coefficient for the cannon ball. Variables v and θ are the instantaneous velocity and angle of elevation of the cannon ball, and x, y are the coordinates of its instantaneous position. For various values of the starting conditions, i.e., v_0, θ_0, and $x_0 = y_0 = 0$, the program should yield the range.

5. A missile is fired vertically up. It has a rocket motor that produces a certain constant upward thrust F for as long as the fuel-burning motor is on. The motor burns the fuel at a constant rate b. Assume that the missile does not go so high as to vary the gravity constant g nor does the drag coefficient vary during the flight. Set up the equation of the vertical motion. Write a program, which gives the vertical distance y at any time t.

6. Suppose a 50-kg heat-seeking missile is fired at a jet bomber flying at a constant speed of 1,500 km/hour. The missile has a rocked motor that produces a thrust of 6000 kg for a period of 5 minutes. Simulate the missile system to test its effectiveness and to estimate how close the target should be before a missile is fired. Assume the drag coefficient $c = .04$ kg/km² and the weight of liquid fuel is 20 kg which is burned in 5 minutes at a constant rate ?

7. In the pure pursuit problem of Section 3.2, plot the path traced by the target and the pursuer. What should be the speed of the fighter to ensure that the bomber is hit in 8 minutes? Determine this value by selecting different values of the time increment. How do this effect the results?

8. In the pure pursuit problem of Section 3.2, at time zero the (x, y) coordinates of the bomber (target) are (0, 0) and of the fighter are (0, d). The target moves in a straight line along the x-axis at constant speed of Vt. The fighter moves at a velocity of Vf and continuously corrects its direction to point in the direction of its target. In this simple case the path traced by the fighter, can be described by the equation,

$$x = \frac{1}{2}\left[\frac{y^{1+r}}{(1+r) dr} - \frac{y^{1-r}}{(1-r)\, d^{-r}}\right] + \frac{rd}{1-r^2}, \qquad \text{where } r = \frac{vt}{vf}$$

First trace the path of the fighter using the simulation technique and then the above analytic expression. Discuss if there are any differences between the two results.

9. Design a three-dimensional pure pursuit problem, and develop a simulation model for the same.
10. In the serial chase problem of Section 3.6, the initial locations of the objects A, B, C and D are (0, 0), (0, 10), (0, 20) and (0, 30) respectively and their velocities are 30 km/hr, 25 km/hr, 20 km/hr and 15 km/hr respectively. Object D moves towards a fixed target at (30, 50), while C moves always in a direction pointing towards D, The object B moves always pointing towards C and object A always pointing towards B. Which object will hit its target first and in what time?
11. An environment consists of two populations, the predators and prey, which interact with each other. The prey are passive, but the predators depend upon prey as their source of food, let $x(t)$ and $y(t)$ denote, respectively, the numbers of individuals in the prey and predator populations at time t. There is an ample supply of food for the prey and in the absence of predators, their growth rate is $rx(t)$ for some positive r. The death rate of the prey due to interaction with predator can be assumed to be proportioned to the product of two population sizes, $x(t)$, $y(t)$. Thus, the overall rate of change of the prey population is given by,

$$\frac{dx}{dt} = rx(t) - ax(t)\,y(t)$$

where a is a positive constant of proportionality.

The predators depend upon prey, and in the absence of prey, their rate of change is $-sy(t)$ for some positive value of s. Due to interaction between the populations, the predators increase at a rate, which is proportionate to $x(t)\,y(t)$. Thus, overall rate of change of the predator population is,

$$\frac{dy}{dt} = -sy(t) + bx(t)\,y(t)$$

where b is a positive constant.

Considering the following values for constants and the initial starting conditions, plot the growth of two populations with time, for a period of 5000 time units.

Constants:

$r = .001 \qquad a = 2 \times 10^{-6} \qquad s = 0.01 \qquad b = 10^{-6}$

Initial conditions;

$x(0) = 12{,}000 \qquad y(0) = 600.$

12. In the chemical reaction simulation of Section 3.3, take the value of delta as 0.05 for time up to 2 minutes and 0.15 for time up to 4 minutes. In the second run, the corresponding values of delta are 0.1 and 0.2. Run the simulations and discuss if there are any differences in the results.

RANDOM NUMBERS

4.1 Random Numbers

The earlier chapters have very clearly illustrated that random numbers are a necessary basic ingredient in the application of Monte Carlo method or simulation of situations involving randomness. There are a large number of systems, where chance plays a part. These systems are called *stochastic* systems. Even for the solution of problems, which are deterministic, random numbers are required for simulation.

What are random numbers? These numbers are samples drawn from a uniformly distributed random variable between some specified intervals, and they have equal probability of occurrence.

Properties of Random Numbers :

A sequence of random numbers has two important statistical properties.

- uniformity, and
- independence.

Each random number is an independent sample drawn from a continuous uniform distribution between an interval 0 to 1. The probability density function (pdf) is shown in Fig. 4.1 and is given by,

$$f(x) = \begin{cases} 1, 0 \le x \le 1 \\ 0, \text{ otherwise} \end{cases}$$

The expected value of each random number R_i is given by

$$E(R) = \int_0^1 x\,dx = \frac{x^2}{2}\Big|_0^1 = \frac{1}{2}$$

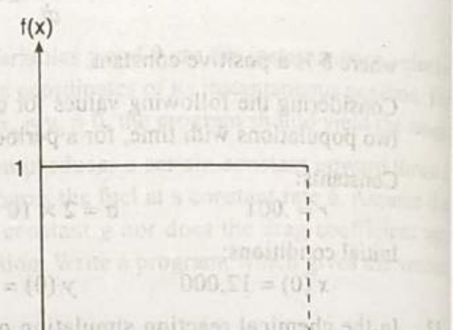


Fig. 4.1

and variance is given by $V(R) = \int_0^1 x^2 dx - [E(R)]^2 = \frac{x^3}{3}\Big|_0^1 - \left[\frac{1}{2}\right]^2 = \frac{1}{3} - \frac{1}{4} = \frac{1}{12}$

If the interval between 0 and 1 is divided into n equal parts or classes of equal length, then,

- the probability of observing a value in a specified interval is independent of the previous values drawn.
- if a total of m observations are taken, then the expected number of observations in each interval is m/n, for uniform distribution.

4.2 Random Number Table

Let us conduct a simple experiment to demonstrate the generation of random numbers. Take ten identical chips of paper and write down the digits 0, 1, 2, 3, ..., 9 on them. Put them in a box, mix them well, and take out one chip. It is a random number between 0 and 9 both inclusive. Repeat the experiment, each time returning the chip to the box and mixing them well. Instead of 10 pieces of paper we can have say 50 with digits 0, 1, 2, 3, ..., 9 repeated 5 times. Each time, draw 5 pieces and note down their numbers. Thus each time 5 random digits are obtained. These can be listed in the form of a table similar to Appendix Table A-1. Such a table is called a random number table. The most comprehensive of all published tables of random numbers is due to RAND Corporation, which contains one million random digits. These numbers were generated by using a special roulette, which incorporated electric devices. A simple roulette wheel, shown in Fig. 4.2, comprises of a disc divided into 10 equal sectors numbered from 0 to 9. The rotating disc is abruptly stopped and the number against the pointer is noted down as a random digit.

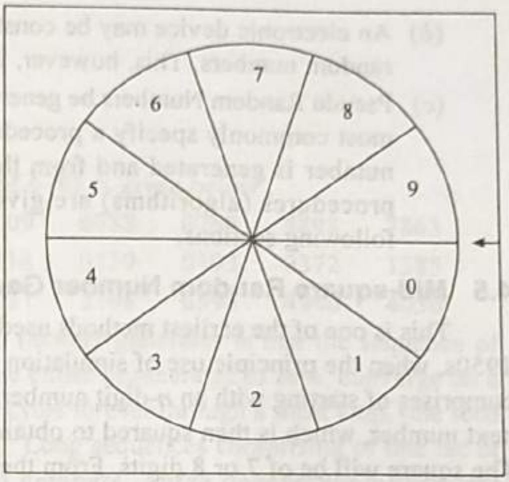


Fig. 4.2

4.3 Pseudo Random Numbers

The 'pseudo' means false. But here word 'pseudo' implies that the random numbers are generated by using some known arithmetic operation. Since, the arithmetic operation is known and the sequence of random numbers can be repeatedly obtained, the numbers cannot be called truly random. However, the pseudo random numbers generated by many computer routines, very closely fulfill the requirement of desired randomness.

If the method of random number generation that is the random number generator is defective, the generated pseudo random numbers may have following departures from ideal randomness.

- The generated numbers may not be uniformly distributed.
- The generated numbers may not be continuous.
- The mean of the generated numbers may be too high or too low.
- The variance may be too high or too low.
- There may be cyclic patterns in the generated numbers, like;
  (*a*) Auto correction between numbers.
  (*b*) A group of numbers continuously above the mean, followed by a group continuously below the mean.

Thus, before employing a pseudo random number generator, it should be properly validated, by testing the generated random numbers for randomness. The random number tests commonly used are explained in Sections 4.10 to 4.14.

4.4 Generation of Random Numbers

In the examples discussed so far, random number table was used to obtain the random observations. In computer simulation, where a very large number of random numbers is generally required, the random numbers can be obtained by the following methods.

(*a*) Random numbers may be drawn from the random number tables stored in the memory of the computer. This process, however, is neither practicable nor economical. It is a very slow process, and the numbers occupy considerable space of computer memory. Above all, in real system simulation, a single run may require many times more random numbers than available in tables.

(b) An electronic device may be constructed as part of the digital computer to generate truly random numbers. This, however, is considered very expensive.

(c) Pseudo Random Numbers be generated by using some arithmetic operation. These methods most commonly specify a procedure, where starting with an initial number, the second number is generated and from that a third number and so on. A number of recursive procedures (algorithms) are given in literature, some of which are described in the following sections.

4.5 Mid-square Random Number Generator

This is one of the earliest methods used for generating pseudo random numbers. It was used in 1950s, when the principle use of simulation was in designing thermonuclear weapons. The method comprises of starting with an *n*-digit number, squaring it and taking the *n* digits in the middle as the next number, which is then squared to obtain the next number. Say, we start with a 4-digit number. The square will be of 7 or 8 digits. From the squared number chop off the two low order digits and one or two high order digits, to obtain a 4-digit number in the middle.

Let the *seed* number be 5673, when squared we get 32182929. After removing two low order digits and two high order digits, we get the next random number 1829. Its square is 3345241. After removing two low order and one high order digit, we get 3452 as random number. Some random numbers obtained from the seed 5673 are;

5673	1829	3452	9163	9605	2560	5536
6472	8867	6236	8876	7833	3558	6593
4676	8649	8052	8347	6724	2121	4986
8601	9772	4919	1965	8612	1665	7722
6292	5892	7156	2083	3388	4785	8962

A computer program in C language for generating random numbers by the mid-square algorithm is given below :

```
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
main()
  {
    /* Mid-Square Method of Generating 4-digit Random Numbers*/
    /* seed is the starting 4-digit random number,
       n is the number of random numbers to be generated
       x is the random number */

   long int i,s,x,y,z,nd,seed;
   int n;
   seed= 6785;
   printf("\n Number of random numbers to be generated n=");
   scanf("%d",&n);

   for(i=1;i<=n;++i) {
   y= seed*seed / 100.;
   z= y/10000.;
```

```
x=int( (y/10000.-z)*10000.);
seed=x;
printf(" %4ld ",x);
}
}
```

A sequence of 30 numbers generated with a seed of 6785 is given below :

0361	1303	6978	6924	9416	6609	6788	0768	5898	7863
8267	3431	7716	5366	7938	0118	0139	0193	0372	1385
9126	2837	484	2342	4848	5031	3108	6595	4940	4036

The problem with the mid-square pseudo random number generator is that the sequence of numbers is limited. With very few exceptions mid-square either degenerates to zero, converge on a constant (the seed 2500 never departs from that value) or cycle forever through a short loop, (the seed 7777 ends up in the cycle 2100, 4100, 8100, 6100, ...). Long sequences comprising of one lac or more numbers can be obtained by using longer seed numbers. Three degenerate mid-square sequences are given below.

Seed 2061 gives;

2061	2477	1355	8360	8896	1388	9265	8402
5936	2360	5696	4444	7491	1150	3225	4006
0480	2304	3084	5110	1121	2566	5843	1406
9768	4138	1230	5129	3066	4003	3240	3576
3317	0024	0005	0000	0000	0000	0000	0000

Seed 1357 gives;

1357	8414	7953	2502	2600	7600	7600	7600 ....

Seed 1379 gives;

1379	9016	2882	3059	3574	7734	8147
3736	9576	6997	9580	7764	2796	8176
8469	7236	4031	2489	1951	8064	0280
0784	6146	7733	7992	8720	0384	1474
1726	9790	8441	2504	2700	2900	4100
8100	6100	2100	4100	8100	6100	2100 ....

4.6 Congruence Method or Residue Method

The most commonly employed pseudo random number generators use the congruence method, also called the method of power residues. This algorithm is described by the expression,

$$r_{i+1} = (ar_i + b) \text{ modulo } m$$

Where a, b and m are constants, r_i and r_{i+1} are ith and $(i + 1)$th random numbers. The expression implies multiplication of a by r_i addition of b and then dividing by m. The r_{i+1} is the remainder or residue. To begin the process of random number generation, in addition to a, b and m, the value of r_0 is also required. It may be any random number and is called seed.

The congruential random number generator may be of the additive, multiplicative or mixed type. The expression given above with $a > 1$ and $b > 0$ is of the mixed type.

If $a = 1$, the expression reduces to the *additive* type.

$$r_{i+1} = (r_i + b) \text{ modulo } m$$

If $b = 0$, the expression reduces to the multiplicative congruential method.

$$r_{i+1} = ar_i \text{ modulo } m.$$

The multiplicative methods are considered better than the additive methods and are as good as the mixed methods.

The selection of values for the constants a, b and m is very important, because on them depends the length of the sequence of random numbers, after which the sequence repeats. It is not possible to generate a non-repeating sequence of numbers with these methods. However, a sufficiently long sequence can be obtained by making a suitable selection of the constants. Since the number can be predicted, rather computed from r_i, and the whole string is reproducible, the numbers obtained are not truly random. They are called pseudo random numbers and hence the method is termed as pseudo random number generator.

Most of the computer languages have a standard function for generating random numbers. In the modern scientific calculators, a random number key is provided. While pressed a random number between 0.000 and 0.999 is generated.

Example 4.1. The pseudo random number generation by the congruential methods can be illustrated by taking some values for a, b and m in the recursive equation.

$$r_{i+1} = (ar_i + b) \bmod m$$

It is better to start with a prime number as modulus m, and prime multiplier a; b can be taken any, say 1. The seed r_0 may be any.

(*a*) **Mixed Multiplicative Congruential (MMC) Generator :**

Taking $a = 13$, $b = 1$ and $m = 19$

And let $r_0 = 1$

$r_1 = (1 \times 13 + 1) \bmod 19 = 14 \bmod 19 = 0$ residue $14 = 14$

$r_2 = (14 \times 13 + 1) \bmod 19 = 183 \bmod 19 = 9$ residue $12 = 12$

$r_3 = (12 \times 13 + 1) \bmod 19 = 157 \bmod 19 = 8$ residue $5 = 5$

$r_4 = (5 \times 13 + 1) \bmod 19 = 66 \bmod 19 = 3$ residue $9 = 9$

$r_5 = (9 \times 13 + 1) \bmod 19 = 118 \bmod 19 = 6$ residue $4 = 4$

$r_6 = (4 \times 13 + 1) \bmod 19 = 53 \bmod 19 = 2$ residue $15 = 15$

$r_7 = (15 \times 13 + 1) \bmod 19 = 196 \bmod 19 = 10$ residue $6 = 6$

$r_8 = (6 \times 13 + 1) \bmod 19 = 79 \bmod 19 = 4$ residue $3 = 3$

$r_9 = (3 \times 13 + 1) \bmod 19 = 40 \bmod 19 = 2$ residue $2 = 2$

$r_{10} = (2 \times 13 + 1) \bmod 19 = 27 \bmod 19 = 1$ residue $8 = 8$

$r_{11} = (8 \times 13 + 1) \bmod 19 = 105 \bmod 19 = 5$ residue $10 = 10$

$r_{12} = (10 \times 13 + 1) \bmod 19 = 131 \bmod 19 = 6$ residue $17 = 17$

$r_{13} = (17 \times 13 + 1) \bmod 19 = 222 \bmod 19 = 11$ residue $13 = 13$

$r_{14} = (13 \times 13 + 1) \bmod 19 = 170 \bmod 19 = 8$ residue $18 = 18$

$r_{15} = (18 \times 13 + 1) \bmod 19 = 235 \bmod 19 = 12$ residue $7 = 7$

$r_{16} = (7 \times 13 + 1) \bmod 19 = 92 \bmod 19 = 4$ residue $16 = 16$

$r_{17} = (16 \times 13 + 1) \bmod 19 = 209 \bmod 19 = 11$ residue $0 = 00$

$r_{18} = (0 \times 13 + 1) \bmod 19 = 01 \bmod 19 = 0$ residue $1 = 01$

This sequence of random numbers between 0 and 18 both inclusive will continue repeating. The number 18, which is one less than modulus is called **Euler function.** This sequence is very short and is no better than the mid-square sequence. However, if the number chosen as the modulus is very large, the pseudo random number sequence will be large and acquires the properties of a true random number sequence.

Random numbers between 0 and 1 can be generated by

$$R_i = \frac{r_i}{m},\ i = 1, 2, 3, \ldots..$$

which gives the sequence as,

$$R_1 = \frac{14}{19} = 0.7368 \qquad R_2 = \frac{12}{19} = 0.6316 \qquad R_3 = \frac{5}{19} = 0.2632$$

$$R_4 = \frac{9}{19} = 0.4737 \qquad R_5 = \frac{4}{19} = 0.2105 \qquad R_6 = \frac{15}{19} = 0.7895$$

$$R_7 = \frac{6}{19} = 0.3158 \qquad \text{etc .....}$$

(*b*) Multiplicative Congruential (MC) Generator :

$$r_{i+1} = ar_i \bmod m$$

Again taking $a = 13$, $m = 19$, and seed $r_0 = 1$

$r_1 = 1 \times 13 \bmod 19 = 0$ residue $13 = 13$
$r_2 = 13 \times 13 \bmod 19 = 8$ residue $18 = 18$
$r_3 = 13 \times 18 \bmod 19 = 12$ residue $6 = 6$
$r_4 = 13 \times 6 \bmod 19 = 4$ residue $2 = 2$
$r_5 = 13 \times 2 \bmod 19 = 1$ residue $7 = 7$
$r_6 = 13 \times 7 \bmod 19 = 4$ residue $15 = 15$
$r_7 = 13 \times 15 \bmod 19 = 10$ residue $5 = 5$
$r_8 = 13 \times 5 \bmod 19 = 3$ residue $8 = 8$
$r_9 = 13 \times 8 \bmod 19 = 5$ residue $9 = 9$
$r_{10} = 13 \times 9 \bmod 19 = 6$ residue $3 = 3$
$r_{11} = 13 \times 3 \bmod 19 = 2$ residue $1 = 1$

The sequence of numbers obtained is 1, 13, 18, 6, 2, 7, 15, 5, 8, 9, 3, 1, which will repeat forever.

(*c*) Additive Congruential Generator :

$$r_{i+1} = (r_i + b) \bmod m$$

Again taking $m = 19$ and $b = 11$

Taking seed $r_0 = 1$

$r_1 = (1 + 11) \bmod 19 = 12$
$r_2 = (12 + 11) \bmod 19 = 4$
$r_3 = (4 + 11) \bmod 19 = 15$
$r_4 = (15 + 11) \bmod 19 = 7$
$r_5 = (7 + 11) \bmod 19 = 18$
$r_6 = (18 + 11) \bmod 19 = 10$
$r_7 = (10 + 11) \bmod 19 = 2$
$r_8 = (2 + 11) \bmod 19 = 13$
$r_9 = (13 + 11) \bmod 19 = 5$
$r_{10} = (5 + 11) \bmod 19 = 16$
$r_{11} = (16 + 11) \bmod 19 = 8$
$r_{12} = (8 + 11) \bmod 19 = 1$

A computer program in C language for generating the random numbers by the mixed congruential method is given below:

```
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
main()
  {    /*Mixed congruential method of generating random numbers
     a,band m are constants,the values of which are to be suitablly
     selected and entered.
     nn is the number of random numbers to be generated.
     seed is the starting random number, which also is to be
     entered. */

     int a,b,k,i,j,m,nn,seed,r[50];
     printf("\n Enter the INTEGER values of a,b,m");
     scanf("%d %d %d",&a,&b,&m);
     printf("\n Enter the INTEGER value of seed ");
     scanf("%d",&seed);
     printf("\n Enter the number of random numbers to be generated");
     scanf("%d",&nn);
     /*a=21; b=53; m=1000; nn=30; */
     r[0]= seed;
     for(i=1;i<=nn;++i) {
     r[i]=(a*r[i-1]+b) % m;
     printf("%4d",r[i]);
     }

  }
```

The output of the above program is given below :

145	98	111	384	117	510	763	76	649
682	375	928	541	414	747	740	593	506
679	312	605	758	971	444	377	970	423
936	709	942	835	588	407	474	007	200
253	366	739	572	065	418	831	504	637
430	83	796	769	202	295	53	281	282

4.7 Arithmetic Congruential Generator

Another kind of pseudo random number generator is the arithmetic congruential algorithm which is given as,

$$r_{i+1} = (r_{i-1} + r_i) \bmod m$$

The process starts with two random numbers, which are added and divided by m with the residue giving the third number. Then 2nd and 3rd numbers result into 4th number, and so on.

For example,

If $r_1 = 9$, $\quad r_2 = 13$ and $m = 17$

$$r_3 = (9 + 13) \bmod 17 = 5$$
$$r_4 = (13 + 5) \bmod 17 = 1$$
$$r_5 = (5 + 1) \bmod 17 = 6$$
$$r_6 = (1 + 6) \bmod 17 = 7$$
$$r_7 = (6 + 7) \bmod 17 = 13$$
$$r_8 = (7 + 13) \bmod 17 = 3$$
$$r_9 = (13 + 3) \bmod 17 = 16$$
$$r_{10} = (3 + 16) \bmod 17 = 2$$

and so on as 1, 3, 4, 7, 11, 13, 8, 4, 12, 16, 11, 10, 4, 14, 1, 15, 16, 14, 13, 10, 6, ...

This results into quite a long sequence.

Remarks: Some studies reported in literature indicated that the multiplicative congruence method is superior to the additive method, and that mixed congruence methods are not noticeably better than simple multiplicative methods. In each method, the quality and length of the random number sequence obtained depends upon the values of constants, the selection of which is a complex problem. To be considered random the sequence of numbers produced must meet various tests to ensure that they are uniformly distributed and that there is no significant correlation between the digits of individual numbers and between the sequential numbers. The length of the sequence before it starts repeating should be sufficiently long.

According to the available guidelines modulus m should be the largest prime number that can be filled to the computer word size, and the multiplier a should be a positive primitive root of m. A positive primitive root is defined as a prime factor or any positive integral power thereof.

$a = 5^5 = 3125$ and $m = 2^{35} - 31 = 3435973837$ is one combination suitable for a 36 word size computer.

$a = 7^5 = 16807$ and $m = 2^{31} - 1 = 2147483647$ is one combination suitable for a 32 bit word size computer.

These values ensure a sequence of over two billion random numbers. Further, specify a seed $r_0 = 123457$. The first few numbers generated using the above values for a, m and r_0 are as under.

$$r_1 = 7^5\,(123457) \bmod (2^{31}\quad 1) = 2{,}074{,}941{,}799$$

$$R_1 = \frac{r_1}{2^{31}} = 0.9662$$

$$r_2 = 7^5\,(2{,}074{,}941{,}799) \bmod (2^{31} - 1) = 559{,}872{,}160$$

$$R_2 = \frac{r_2}{2^{31}} = 0.2607$$

$$r_3 = 7^5\,(559{,}872{,}160) \bmod (2^{31} - 1) = 1{,}645{,}535{,}613$$

$$R_3 = \frac{r_3}{2^{31}} = 0.7662$$

In this case while determining random number between 0 and 1, r_1 has been divided not by m but by $m + 1$. This does not make much difference, as the value of m is very large.

4.8 Combined Congruential Generators

The random number generators discussed so far, may not be adequate for some complex applications, where hundreds and thousands of elementary events must be simulated before a significant event occur. The simulation of complex computer networks, in which thousands of users are executing hundreds of programs, require substantially longer periods. One method of meeting such a demand is to combine two or more multiplicative congruential generators in such a way that the combined generator has good statistical properties and a longer period.

If $r_{i,1}, r_{i,2}, \ldots, r_{i,k}$ are the ith output from k different multiplicative congruential generators, where the jth generator has prime modulus m_j and the multiplier a_i chosen so that the period is $m_j - 1$, then the combined generator will give,

$$r_i = \left(\sum_{j=1}^{k} (-1)^{j-1} r_{i,j}\right) \bmod m_1 - 1$$

with

$$R_i = \begin{cases} \dfrac{r_i}{m_i}, & r_i > 0 \\ \dfrac{m_i - 1}{m_i} & r_i = 0 \end{cases}$$

The maximum possible period for such a generator is

$$P = \frac{(m_1 - 1)(m_2 - 1) \ldots\ldots (m_k - 1)}{2^k - 1}$$

One such a generator suggested by Le Ecuyer is for $k = 2$. Values of m_1 and m_2 are 2147483563 and 2147483399, while a_1 and a_2 are 40014 and 40692. The value of seed $r_{1,0}$ is selected between 1 and 2147483562 and that of $r_{2,0}$ is selected between 1 and 2147483398.

4.9 Qualities of an Efficient Random Number Generator

- It should have a sufficiently long cycle, that is, it should generate a sufficiently long sequence of random numbers, before beginning to repeat the sequence.
- The random numbers generated should be replicable, that is by specifying the starting conditions, it should be possible to obtain the same set of random numbers, as and when desired. Many times common random numbers are required for the comparison of two systems.
- The generated random numbers should fulfill the requirements of uniformity and independence.
- The random number generator should be fast and cost-effective.
- It should be portable to different computers and ideally to different programming languages.

4.10 Testing Numbers for Randomness

A sequence of random numbers is considered to be random, if;

(*i*) The numbers are uniformly distributed, that is every number has an equal chance of occurrence.

(*ii*) The numbers are not serially autocorrelated. This means that there is no correlation between adjacent pairs or numbers, or that the appearance of one number does not influence the appearance of next number. The sequence 1, 3, 5, 7, 9, ... or 1, 4, 2, 6, 3, 1, 4, 7, 5, ... or 1, 3, 5, 2, 4, 6, 3, 5, 9, ... are serially correlated.

There are a number of tests, which are used to ensure that the random numbers are uniformity distributed and are not serially autocorrelated, some of which are discussed below.

4.11 Uniformity Test

The test of uniformity or frequency test is a basic test that should always be performed to validate a random number generator. Two frequency tests are available. They are, Kolmogorov-Smirnov test and the Chi-Squared test. Both of these tests compare the generated random numbers with the theoretical uniform distribution.

The Kolmogorov-Smirnov Test

This test compares the continuous cdf, $F(x)$ of the uniform distribution to the empirical cdf, $S_N(x)$, of the sample of N random numbers. The largest absolute deviation between $F(x)$ and $S_N(x)$ is determined and is compared with the critical value, which is available as function of N in Appendix Table A-6, for various levels of significance. The procedure of employing the Kolmogorov-Smirnov uniformity test is clearly illustrated in the next example.

Example 4.2. The Kolmogorov-Smirnov test is to be performed to test the uniformity of following random numbers with a level of significance of $\alpha = 0.05$.

.24, .89, .11, .61, .23, .86, .41, .64, .50, .65

The calculations of the test are given in Table 4.1. The top row of the table lists the given random numbers R_i ($i = 1, N$) in the ascending order. Here $N = 10$. In the second row, the numbers are computed from the empirical distribution, i.e., i/N values are listed. In the third row deviation, $\frac{i}{N} - R_i$ is computed maximum of which gives D^+, while in the fourth row, the deviation $R_i - \frac{(i-1)}{N}$ is computed, the maximum of which given D^-.

The largest deviation, $D = \max(D^+, D^-)$.

From the table, $D^+ = 0.15$, $D^- = 0.13$ giving the largest deviation $D = 0.15$.

The critical value of D obtained from Appendix Table A-6 for $\alpha = 0.05$ and $N = 10$ is 0.410. Since the computed value 0.15 is less than the critical value, the given random numbers are uniform at 95% level of significance. At $\alpha = 0.01$, critical values is 0.368, which again is more than 0.15, hence, the given random numbers are uniform even at 99% level of significance.

Table 4.1

R_i	.11	.23.	.24	.41	.50	.61	.64	.65	.86	.89
i/N	.10	.20	.30	.40	.50	.60	.70	.80	.90	1.0
$i/N - R_i$	—	—	.06	—	00	—	.06	.15	.04	.11
$R_i - (i-1)/N$	.11	.13	.04	.11	.10	.11	.04	—	.06	—

4.12 Chi-Squared Test

The Chi-Squared test uses the sample statistic

$$\chi_0^2 = \sum_{i=1}^{n} \frac{(O_i - E_i)^2}{E_i}$$

where O_i is the observed number in the ith class, E_i is the expected number in the ith class and n is the number of classes. For the uniform distribution, E_i, the expected number in each class is given by

$$E_i = \frac{N}{n}$$

for equally spaced classes, where N is the total number of observations. It can be shown that the sampling distribution of χ_0^2 is approximately the chi-square distribution with $n - 1$ degrees of freedom.

This test involves the classification of 500 random numbers between 0 and 1 into 10 equal intervals, that is numbers less than or equal to 0.1 less than or equal to 0.2, 0.3, ..., 1.0. A bar chart or histogram can than be plotted to illustrate the uniformity of distribution. If the 500 numbers were distributed among the 10 classes, with perfect uniformity, all bars will be of equal length of 50 numbers.

Chi-square is a characteristic of the distribution, which is a measure of its randomness. The statistic Chi-square is computed by subtracting the number of random numbers in each class from the expected number (that is 50), squaring the difference, adding the squares for the ten classes, and dividing the sum by the expectation (50).

For example, if 47, 49, 55, 54, 46, 53, 51, 43, 46 and 54 are the numbers in 10 classes.

Then their differences from 50 are, 3, 1, 5, 4, 4, 3, 1, 7, 4, and 4; and sum of the squares of differences is $9 + 1 + 25 + 16 + 16 + 9 + 1 + 49 + 16 + 16 = 158$

$$\text{Chi-square} = \frac{158}{50} = 3.16$$

There are tables of Chi-square that will tell us about the goodness of the results. The acceptable value of Chi-square will depend upon the degrees of freedom which are one less than the number of classes, and the level of confidence we wish to place in our results. In the present case, since there are 10 classes in which we have divided the random numbers, there are 9 degrees of freedom. At 95% confidence level the acceptable value of Chi-square for 9 degrees of freedom is 16.919 (Appendix Table A-2). This means that when the value of Chi-square is less than or equal to 16.9, there are only 5 chances out of 100 that our results are wrong.

Chi-Squared Test can be employed to compare different sets of random numbers or different random number generators. For each set of random numbers the statistics Chi-square is computed. The one with smaller value of Chi-square is more uniformly distributed.

Example 4.3. The two-digit random numbers generated by a multiplicative congruential method are given below. Determine Chi-square. Is it acceptable at 95% confidence level?

36, 91, 51, 02, 54, 06, 58, 06, 58, 02, 54, 01, 48, 97, 43,
22, 83, 25, 79, 95, 42, 87, 73, 17, 02, 42, 95, 38, 79, 29,
65, 09, 55, 97, 39, 83, 31, 77, 17, 62, 03, 49, 90, 37, 13,
17, 58, 11, 51, 92, 33, 78, 21, 66, 09, 54, 49, 90, 35, 84,
26, 74, 22, 62, 12, 90, 36, 83, 32, 75, 31, 94, 34, 87, 40,
07, 58, 05, 56, 22, 58, 77, 71, 10, 73, 23, 57, 13, 36, 89,
22, 68, 02, 44, 99, 27, 81, 26, 85.

Solution: The given 100 random numbers can be divided into 10 classes as given below:

Class	*Count*	*Frequency*	*Diff*	*(Diff)*^{2}
$0 < r \leq 10$	*************	13	3	9
$10 < r \leq 20$	********	8	2	4
$20 < r \leq 30$	*********	9	1	1
$30 < r \leq 40$	*************	13	3	9
$40 < r \leq 50$	*******	7	3	9
$50 < r \leq 60$	*************	13	3	9
$60 < r \leq 70$	*****	5	5	25
$70 < r \leq 80$	************	12	2	4
$80 < r \leq 90$	************	12	2	4
$90 < r \leq 100$	********	8	2	4
				78

$$\text{Chi-square} = \frac{78}{10} = 7.8$$

For 9 (10 – 1) degrees of freedom, at 95% confidence level, the acceptable value of Chi-square is 16.9, which is more than 7.8. Hence, the given set of random numbers is acceptable, so far as its uniformity in distribution is concerned.

4.13 Testing for Autocorrelation

The uniformity test of random numbers is only a necessary test for randomness, not a sufficient one. A sequence of numbers may be perfectly uniform, and still not random. For example, the sequence .1, .2, .3., .4., .5, .6, .7, .8, .9, .1, .2, .3, ..., would give a perfectly uniform distribution with the chi-square value as zero. But the sequence can by no means be regarded as random. The numbers are not independent, as the occurrence of one number say .3 decides the next, which is to be .4 etc. This defect is called 'serial autocorrelation' of adjacent pairs of numbers.

The Chi-Squared test for serial autocorrelation makes use of a 10 × 10 matrix (checker board). The 10 classes described in the uniformity test are represented both along the rows and columns. If the classes are to be represented on a bar chart, 100 bars, one for each cell of the matrix, will be required. To reduce the number of groups, instead of 10, random numbers are divided into smaller number of classes as 3, or 4. Three classes will be: less than or equal to 0.33, less than or equal to 0.67 and less than or equal to 1.0. With three classes in rows and three in columns, there will be 9 cells.

Let us consider the following random numbers :

49	95	82	19	41	31	12	53	62	40	87	83	26	01	91
55	38	75	90	35	71	57	27	85	52	08	35	57	88	38
77	86	29	18	09	96	58	22	08	93	85	45	79	68	20
11	78	93	21	13	06	32	63	79	54	67	35	18	81	40
62	13	76	74	76	45	29	36	80	78	95	25	52.		

These 73 random numbers giving 72 pairs, are grouped into 9 classes with expectation of 8 in each group.

Class	*Count*	*Frequency*	*Diff*	*(Diff)*^{2}
$R_1 \le .33$ & $R_2 \le 0.33$	*********	9	1	1
$R_1 \le .67$ & $R_2 \le 0.33$	*******	7	1	1
$R_1 \le 1.0$ & $R_2 \le 0.33$	******	6	2	4
$R_1 \le .33$ & $R_2 \le 0.67$	******	6	2	4
$R_1 \le .67$ & $R_2 \le 0.67$	********	8	0	0
$R_1 \le 1.0$ & $R_2 \le 0.67$	*********	9	1	1
$R_1 \le .33$ & $R_2 \le 1.0$	******	7	1	1
$R_1 \le .67$ & $R_2 \le 1.0$	*********	9	1	1
$R_1 \le 1.0$ & $R_2 \le 1.0$	***********	11	3	9
		72		24

$$\text{Chi-square} = \frac{24}{8} = 3.0$$

The counts in different classes have been determined by taking the pairs of random numbers. Pair .49 and .95 falls in class $R_1 \le .67$ and $R_2 \le 1.0$. Then the next pair is .95 and .82 which falls in class $R_1 \le 1.0$ and $R_2 \le 1.0$, Pair .82 and .19 falls in class $R_1 \le 1.0$ and $R_2 \le 0.33$ and so on. Since the total number of pairs in 72, one less than the number of random numbers, the expectation is 8, that is 8 pairs in each class. Then squares of differences (frequency – expectation) are determined and their sum is obtained, which on division by expectation gives the value of Chi-square as 3.0.

In this case there are two variables R_1 and R_2 and hence the degrees of freedom are nine minus two that is seven. The criterion value of χ^2 (chi-square) for seven degrees of freedom at 95% confidence level is 14.067. The value of Chi-square obtained for the given set of random numbers is well within the acceptable limit, and hence, they are not serially autocorrelated.

Example 4.4. Given below is a sequence of random numbers. Perform the Chi-Squared tests to check the numbers for uniform distribution and serial autocorrelation.

07	05	96	14	10	90	21	15	84	28	20	78	35	25
72	42	30	66	49	35	60	56	40	54	63	45	48	70
50	42	77	55	36	84	60	30	91	65	24	98	70	18
07	75	12	14	80	06	21	85	96	28	90	90	35	95
84	42	05	78	49	10	72	56	15	66	63	20	60	70
25	54	77	30	48	84	35	42	91	40	36	98	45	30
07	50	24	14	55	18	21							

4.13.1 Uniformity Test

For checking the random numbers for the uniformity, we will divide these in 10 class and take 90 random numbers to have the expected value in each class as 9.

There are ten classes and one variable, giving 9 degrees of freedom. For 9 degrees of freedom at 95% confidence level, the acceptable value of χ^2 is up to 16.9. Our value of χ^2 is well within the acceptable limit and hence, the random numbers in the given sequence are uniformly distributed.

Class	*Count*	*Frequency*	*(Diff)²*
$0 \le R \le 10$	* * * * * * * * *	9	0
$11 \le R \le 20$	* * * * * * * * *	9	0
$21 \le R \le 30$	* * * * * * * * * * * *	12	9
$31 \le R \le 40$	* * * * * * * *	8	1
$41 \le R \le 50$	* * * * * * * * * * * *	12	9
$51 \le R \le 60$	* * * * * * * * *	9	0
$61 \le R \le 70$	* * * * * * * *	8	1
$71 \le R \le 80$	* * * * * * * *	8	1
$81 \le R \le 90$	* * * * * * * *	8	1
$91 \le R \le 100$	* * * * * * *	7	4
		90	26

Chi-square, $\chi^2 = \dfrac{26}{9} = 2.9$

4.13.2 Chi-Squared Test for Autocorrelation

For this test, the overlapping pairs of random numbers are to be taken. Taking three classes 33; 34 to 67 and 68 to 100 for each random number in the pair, the number of pairs in each s is obtained as follows. Since there are 9 classes, the expectation value is 10.

R_1	R_2	Count	Frequency	$(Diff)^2$
≤ 33,	≤ 33	* * * * * * * * * *	10	0
≤ 67,	≤ 33	* * * * * * * * * *	10	0
≤ 100,	≤ 33	* * * * * * * * *	9	1
≤ 33,	≤ 67	* * * * * * *	7	9
≤ 67,	≤ 67	* * * * * * * * * * * * * * * * * * * *	20	100
≤ 100,	≤ 67	* * * * * * * * * * *	11	1
≤ 33,	≤ 100	* * * * * * * * * * * * *	13	9
≤ 67,	≤ 100	* * * * * *	6	16
≤ 100,	≤ 100	* * * *	4	36
			90	172

$$\text{Chi-square } (\chi^2) = \frac{172}{10} = 17.2$$

The criterion value of χ^2 for 7 degrees of freedom at 95% confidence level is 14.1, which is less than the value we have obtained for the given sequence. Thus, the given sequence of random numbers is serially autocorrelated at 95% confidence level.

4.14 Poker Test

This test gets its name from a game of cards called poker. This test not only tests the randomness of the sequence of numbers, but also the digits comprising of each number. Every random number of five digits or every sequence of five digits is treated as a poker hand.

71549 are five different digits
55137 would be a pair
33669 would be two pairs
55513 would be three of a kind
44477 would be a full house
77774 would be four of a kind
88888 would be five of a kind

The occurrence of five of a kind is rare. The order of the 'cards' within a 'hand' is unimportant, the straights, flushes and royals of the poker game are disregarded in Poker Test.

In 10,000 random and independent numbers of five digits each, you may expect the following distribution of various combinations.

Five different digits	3024	or	30.24%
Pairs	5040		50.40%
Two-pairs	1080		10.80%
Three of a kinds	720		7.20%
Full houses	90		0.90%
Four of a kinds	45		0.45%
Five of a kinds	1		0.01%

Example 4.5 : Poker Test

A sequence of 10,000 five-digit random numbers has been generated, and an analysis of numbers indicate that there are 3075 numbers having five different digits, 4935 having a pair, 1135 having two pairs, 695 having three of a kind, 105 having full house (three of a kind and a pair) 54 having four of a kind and one having all five of a kind. Use Poker Test to determine if these random numbers are independent, at $\alpha = 0.01$.

The calculations of χ^2 are given in Table 4.2.

Table 4.2

Combination Distribution i	Observed Distribution O_i	Expected E_i	$\frac{(O_i - E_i)^2}{E_i}$
Five different digits	3075	3024	0.8601
Pairs	4935	5040	2.1875
Two-pairs	1135	1080	1.8750
Three of a kind	695	720	0.868
Full houses	105	90	2.5
Four of a kind	54	45	1.8
Five of a kind	01	01	0.0
	10,000	10,000	10.0907

The appropriate degrees of freedom in this case are 6, one less than the number of combinations.

The critical value of χ^2 for six degrees of freedom at $\alpha = 0.01$ is 16.8. The value of χ^2 obtained from calculations is 10.097, which is less than the critical value. Hence, the hypothesis of independence cannot be rejected on the basis of Poker Test.

Example 4.6. Use the mixed congruential method to generate the following sequence of random numbers.

(*a*) A sequence of ten two-digit numbers, such that $r_{n+1} = (21r_n + 53)$ modulo 100. Take $r_0 = 52$.

(*b*) A sequence of ten random numbers between 0 and 31 such that,

$r_{n+1} = (13r_n + 15)$ modulo m. Table $r_0 = 11$.

Solution:

(*a*) $r_{n+1} = (21r_n + 53) \bmod 100$

Given $r_0 = 52$

$$
\begin{aligned}
r_1 &= (21 \times 52 + 53) \bmod 100 = 1145 \bmod 100 = 45\\
r_2 &= (21 \times 45 + 53) \bmod 100 = 998 \bmod 100 = 98\\
r_3 &= (21 \times 98 + 53) \bmod 100 = 2111 \bmod 100 = 11\\
r_4 &= (21 \times 11 + 53) \bmod 100 = 284 \bmod 100 = 84\\
r_5 &= (21 \times 84 + 53) \bmod 100 = 1817 \bmod 100 = 17\\
r_6 &= (21 \times 17 + 53) \bmod 100 = 410 \bmod 100 = 10\\
r_7 &= (21 \times 10 + 53) \bmod 100 = 263 \bmod 100 = 63\\
r_8 &= (21 \times 63 + 53) \bmod 100 = 1376 \bmod 100 = 76\\
r_9 &= (21 \times 76 + 53) \bmod 100 = 1649 \bmod 100 = 49\\
r_{10} &= (21 \times 49 + 53) \bmod 100 = 1082 \bmod 100 = 82
\end{aligned}
$$

The required sequence of random numbers is 52, 45, 98, 11, 84, 17, 10, 63, 76, 49, 82.

(*b*) $r_{n+1} = (13r_n + 15) \bmod m$

Since the random numbers required have to be between 0 and 31, the value of m will be taken as 32.

Given $r_0 = 11$

$$
\begin{aligned}
r_1 &= (13 \times 11 + 15) \bmod 32 = 158 \bmod 32 = 30\\
r_2 &= (13 \times 30 + 15) \bmod 32 = 405 \bmod 32 = 21\\
r_3 &= (13 \times 21 + 15) \bmod 32 = 288 \bmod 32 = 00\\
r_4 &= (13 \times 0 + 15) \bmod 32 = 15 \bmod 32 = 15\\
r_5 &= (13 \times 15 + 15) \bmod 32 = 210 \bmod 32 = 18\\
r_6 &= (13 \times 18 + 15) \bmod 32 = 249 \bmod 32 = 25\\
r_7 &= (13 \times 25 + 15) \bmod 32 = 340 \bmod 32 = 20\\
r_8 &= (13 \times 20 + 15) \bmod 32 = 275 \bmod 32 = 19\\
r_9 &= (13 \times 19 + 15) \bmod 32 = 262 \bmod 32 = 06
\end{aligned}
$$

The required sequence of 10 numbers is 11, 30, 21, 00, 15, 18, 25, 20, 19, 06.

Example 4.7. Generate a sequence of five random numbers such that

(*a*) $r_{i+1} = ar_1 \bmod m$, taking $a = 16$, $m = 23$ and $r_0 = 15$

(*b*) $r_{i+1} = (r_1 + b) \bmod m$, taking $b = 11$, $m = 17$ and $r_0 = 1$

Solution:

(*a*) $r_{i+1} = ar_1 \bmod m = 16r_i \bmod 23$

Given $r_0 = 15$

$$
\begin{aligned}
r_1 &= 16 \times 15 \bmod 23 = 240 \bmod 23 = 10\\
r_2 &= 16 \times 10 \bmod 23 = 160 \bmod 23 = 22\\
r_3 &= 16 \times 22 \bmod 23 = 352 \bmod 23 = 7\\
r_4 &= 16 \times 7 \bmod 23 = 112 \bmod 23 = 20\\
r_5 &= 16 \times 20 \bmod 23 = 320 \bmod 23 = 21\\
r_6 &= 16 \times 21 \bmod 23 = 336 \bmod 23 = 14\\
r_7 &= 16 \times 14 \bmod 23 = 224 \bmod 23 = 17\\
r_8 &= 16 \times 17 \bmod 23 = 272 \bmod 23 = 19\\
r_9 &= 16 \times 19 \bmod 23 = 304 \bmod 23 = 5\\
r_{10} &= 16 \times 5 \bmod 23 = 80 \bmod 23 = 11\\
r_{11} &= 16 \times 11 \bmod 23 = 176 \bmod 23 = 15
\end{aligned}
$$

The required sequence of random numbers is 15, 10, 22, 07, 20, 21, 14, 17, 19, 05.

(*b*) $r_{i+1} = (r_1 + b) \bmod m = (r_i + 11) \bmod 17$

Given $r_0 = 1$

$$
\begin{aligned}
r_1 &= (1 + 11) \bmod 17 = 12\\
r_2 &= (12 + 11) \bmod 17 = 06\\
r_3 &= (6 + 11) \bmod 17 = 00\\
r_4 &= (0 + 11) \bmod 17 = 11\\
r_5 &= (11 + 11) \bmod 17 = 05\\
r_6 &= (5 + 11) \bmod 17 = 16\\
r_7 &= (16 + 11) \bmod 17 = 10\\
r_8 &= (10 + 11) \bmod 17 = 04\\
r_9 &= (4 + 11) \bmod 17 = 15\\
r_{10} &= (15 + 11) \bmod 17 = 09
\end{aligned}
$$

The required sequence of random numbers is 01, 12, 06, 00, 11, 05, 16, 10, 04, 15, 09.

Example 4.8. Use the mid of square method to generate 10 four-digit numbers, taking the seed as 9876.

Solution: After squaring the given four-digit number, we will chop off the last two digits and then take the next four as the mid of the square. This will give the next four-digit random number.

r_i	r_i^2	r_{i+1}
9876	97,5353,76	5353
5353	28,6546,09	6546
6546	42,8501,16	8501
8501	72,2670,01	2670
2670	7,1289,00	1289
1289	1,6615,21	6615
6615	43,7582,25	7582
7582	57,4867,24	4867
4867	23,6876,89	6876
6876		

Thus, 9876, 5353, 6546, 8501, 2670, 1289, 6615, 7582, 4867, 6876 is the required sequence of random numbers.

Example 4.9. Employ the arithmetic congruential generator, to generate a sequence of 10 random numbers given $r_1 = 987$ $r_2 = 535$ and modulo $m = 1000$

Solution:

$$
\begin{aligned}
r_{i+2} &= (r_i + r_{i+1}) \text{ modulo } m \\
r_3 &= (r_1 + r_2) \text{ modulo } 1000 \\
&= (987 + 535) \bmod 1000 = 1522 \bmod 1000 = 522 \\
r_4 &= (535 + 522) \bmod 1000 = 1057 \bmod 1000 = 057 \\
r_5 &= (522 + 057) \bmod 1000 = 579 \bmod 1000 = 579 \\
r_6 &= (057 + 579) \bmod 1000 = 636 \bmod 1000 = 636 \\
r_7 &= (579 + 636) \bmod 1000 = 1215 \bmod 1000 = 215 \\
r_8 &= (636 + 215) \bmod 1000 = 851 \bmod 1000 = 851 \\
r_9 &= (215 + 851) \bmod 1000 = 1066 \bmod 1000 = 066 \\
r_{10} &= (851 + 66) \bmod 1000 = 917 \bmod 1000 = 917
\end{aligned}
$$

987, 535, 522, 057, 579, 636, 215, 851, 066, 917 is the required sequence of random numbers.

Example 4.10. A sequence of 10,000 random numbers, each of four digits has been generated. This sequence is to be tested for independence using Poker Test. The analysis of the numbers reveals that in 5120 numbers all four digits are different, in 4230, there is one pair in each number, in 560, there are two pairs, while in 75, there are three digits of a kind and in 15 cases all the four digits are same. Determine, if the random numbers are independence at $\alpha = 0.05$.

In this example, each random number is of four digits. The following combination are possible:

(*a*) Four different digits

(*b*) One pair in the digits

(*c*) Two pairs

(*d*) Three digits of one kind

(*e*) All four digits of one kind.

The probability of occurrence of combination (*a*), that is four different digits is,

$$P(a) = .9 \times .8 \times .7 = 0.504$$

The probability having one pair and the other two different digits is,

$$P(b) = \binom{4}{2} \times .1 \times .9 \times .8 = 0.432$$

The probability of having two pairs of like digits is,

$$P(c) = \binom{4}{2} \times .1 \times .1 \times .9 = 0.054$$

The probability of having three digits of a kind is,

$$P(d) = .1 \times .1 \times .9 = 0.009$$

The probability of having all four digits of one kind is

$$P(e) = .1 \times .1 \times .1 = 0.001$$

The test is summarized below in Table 4.3.

Table 4.3

Combination Distribution i	*Observed Distribution* O_i	*Expected* E_i	$\frac{(O_i - E_i)^2}{E_i}$
Four different digits	5120	5040	1.2698
Pairs	4230	4320	1.8750
Two pairs	560	540	0.7407
Three of a kind	75	90	2.5000
Four of a kind	15	10	2.5000
	10000	10000	8.8856

Since, there are five combinations, number of degrees of freedom is four. The critical value $\chi^2_{0.05,\,4} = 9.49$, which is more than the χ^2 obtained above. Hence, the hypothesis that the random numbers are independent cannot be rejected on the basis of Poker Test.

Example 4.11. A sequence of 1000 three-digit random numbers has been generated and their analysis indicates that 680 have three different digits, 289 contain exactly one pair of like digits and 31 contain three like digits. Using Poker Test, these numbers are to be tested for independence at $\alpha = 0.05$.

In case of three-digit number, the possible combinations are

(*a*) All three of different kind

(*b*) One pair

(*c*) All three of same kind.

The probabilities of occurrence of these three combinations are,

$$p(a) = .9 \times .8 = 0.72$$

$$p(b) = \binom{3}{2} \times .1 \times .9 = 0.27$$

$$p(c) = .1 \times .1 = 0.01$$

The calculations of χ^2 are given in Table 4.4.

Table 4.4

Combination Distribution i	Observed Distribution O_i	Expected E_i	$\frac{(O_i - E_i)^2}{E_i}$
Three different digits	680	720	2.2222
One Pair	289	270	1.3370
Three of a kind	31	10	44.1000
	1000	1000	47.6592

The number of degrees of freedom in this case is 2, one less than the number of combinations. Since $\chi^2_{0.05,\,2} = 5.99 < 47.66$, the hypothesis of independence of the numbers is rejected.

4.15 Exercises

1. Describe a procedure to physically generate random numbers on the interval [0, 1] with two-digit accuracy.
2. Why do the random numbers generated by computer are called pseudo random numbers. Demonstrate the Mid-square random number generation method, taking the following numbers as seeds. Generate 40 random numbers in each case.

   (*a*) (*i*) 2061 (*ii*) 1357 (*iii*) 1379 (*iv*) 3452

   What is wrong with each sequence of random numbers ?

   (*b*) (*i*) 4789 (*ii*) 7583 (*iii*) 3789
3. Use the mixed congruential method to generate the following sequences of random numbers.

   (*a*) A sequence of 10 two-digit random numbers such that
   $$r_{n+1} = (21r_n + 53) \text{ modulo } 100.$$
   Take $r_0 = 46$.

   (*b*) A sequence of 10 random numbers between 0 and 40, such that
   $$r_{n+1} = (9r_n + 15) \text{ modulo } m.$$
   Take $r_0 = 12$.
4. Repeat problem 3, when the mixed congruential method is reduced to

   (*a*) multiplicative congruential method

   (*b*) additive congruential method
5. Generate a sequence of 10 random numbers employing the Arithmetic congruential generator, when $r_1 = 89$ and $r_2 = 38$ modulus $m = 23$.
6. Use the multiplicative congruential method to generate a sequence of four three-digit random numbers. Let $r_0 = 117$, $a = 3$, $m = 1000$. **[PTU, B. Tech (Prod.) May 2006]**
7. The following sequence of random numbers have been generated

   0.37, 0.55, 0.71, 0.97, 0.65, 0.29, 0.84, 0.78, 0.23

   Use the Kolmogorov-Smirnov Test with $\alpha = 0.05$ to determine, if these numbers are uniformly distributed over the interval 0 to 1.
8. Take 100 two-digit random numbers from the random number table and test their uniformity employing the Chi-Squared Test. Are the numbers uniformly distributed at (*a*) 95% confidence level, (*b*) 99% confidence level?

9. Test the random numbers taken in Problem 8 for serial autocorrelation, by employing the Chi-Squared Test.

10. A sequence of 100 random numbers is given below. Use Chi-Squared Test with $\alpha = 0.05$ to test whether these numbers are uniformly distributed.

21	81	92	23	96	20	68	57	79	84
82	62	12	08	92	83	74	85	60	49
48	37	65	74	22	11	28	10	55	82
72	95	08	85	79	95	86	11	16	52
70	55	50	87	67	51	72	38	29	62
71	12	07	75	56	34	40	67	24	86
18	82	41	29	63	06	84	01	20	06
06	33	14	79	25	65	57	47	74	68
54	35	81	07	88	96	70	85	29	13
12	91	26	57	30	22	90	03	13	31

11. Test the numbers given in Exercise 10, for independence, by employing the Chi-Squared Test. Take $\alpha = 0.05$.

12. Take 100 five-digit random numbers from the random number table and employ Poker Test to check the uniformity of their distribution.

13. A sequence of 1000 four-digit numbers has been generated and an analysis indicates the following combinations and frequencies. Use Poker Test to determine, whether these numbers are independent. Use $\alpha = 0.05$.

Combination	*Observed frequency*
Four different digits	567
One pair	390
Two pairs	19
Three digits of a kind	23
Four digits of a kind	1
	1000

14. Write a computer program that will generate four-digit random numbers using the multiplicative congruential method. Allow the user to input values of r_0, a, b and m.

15. Test the following sequence of numbers for uniformity and independence using procedures you learned in this chapter:

.883, .748, .969, .302, .866, .053, .214, .111, .554, .611,
.964, .033, .444, .459, .000, .725, .724, .405, .160, .736

16. (*a*) What do you mean by system simulation?

(*b*) Why the random numbers generated by computer are called pseudo random numbers? Discuss the congruence method of generating random numbers.

Chapter 5

STATISTICAL CONSIDERATIONS

5.1 Stochastic Activities

The description of activities can be of two types, deterministic and stochastic. The process in which the outcome of an activity can be described completely in terms of its input is deterministic and the activity is said to be deterministic activity. On the other hand, when the outcome of an activity is random, that is there can be various possible outcomes, the activity is said to be stochastic. In case of an automatic machine, the output per hour is deterministic, while in a repair shop, the number of machines repaired will vary from hour to hour, in a random fashion. The terms random and stochastic are used interchangeably.

A stochastic process, can be defined as an ordered set of random variables, the ordering of the set usually being with respect to time. The variation in the ordered set may be discrete or continuous.

A random variable x is called **discrete** if the number of possible values of x (*i.e.*, range space) is finite or countably infinite, i.e., possible values of x may be $x_1, x_2, \ldots, x_n$. The list terminates in the finite case and continues indefinitely in the countable infinite case. The demand of an item can be say 0, 1, 2, 3 or 4 per day, each having its own probability of occurrence. This demand is a discrete activity. Colour of a traffic signal light encountered by a randomly arriving vehicle may be red, amber or green, is a discrete activity. A random variable is called **continuous** if its range space is an interval or a collection of intervals. A continuous variable can assume any value over a continuous range. For example, heights of school children, temperatures and barometric pressures of different cities and velocity of wind are examples of continuous variables.

A stochastic process is described by a probability law, called probability density function.

5.2 Discrete Probability Functions

If a random variable x can take x_i $(i = 1, n)$ countably infinite number of values, with the probability of value x_i being $p(x_i)$, the set of numbers $p(x_i)$ is said to be a probability distribution or probability mass function of the random variable x. The numbers $p(x_i)$ must satisfy the following two conditions:

(*i*) $p(x_i) \geq 0$ for all values of i.

(*ii*) $\sum_{i=1}^{n} p(x_i) = 1$

The probability distribution may be a known set of numbers. For example, in case of a dice, the probability of each of the six faces is 1/6.

x	1	2	3	4	5	6
$p(x)$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$

If two coins are flipped together, the probabilities of both heads, one head one tail and both tails are $\frac{1}{4}$, $\frac{1}{2}$ and $\frac{1}{4}$ respectively.

Example 5.1: A pair of fair dice is rolled once. The sum of the two numbers on the dice represents the outcome for a random variable x. Determine the probability distribution of x. What is the probability that x is odd?

[P.U. B.E. (ECE) April 2004]

Solution: The sum of the two numbers on the dice, x will range from $1 + 1 = 2$ to $6 + 6 = 12$, or the range of x is 2 to 12 both inclusive.

$x = 2$ can occur in **one** way $(1 + 1)$
$= 3$ can occur in **two** ways $(1 + 2, 2 + 1)$
$= 4$ can occur in **three** ways $(1 + 3, 2 + 2, 3 + 1)$
$= 5$ can occur in **four** ways $(1 + 4, 2 + 3, 3 + 2, 4 + 1)$
$= 6$ can occur in **five** ways $(1 + 5, 2 + 4, 3 + 3, 4 + 2, 5 + 1)$
$= 7$ can occur in **six** ways $(1 + 6, 2 + 5, 3 + 4, 4 + 3, 5 + 2, 6 + 1)$
$= 8$ can occur in **five** ways $(2 + 6, 3 + 5, 4 + 4, 5 + 3, 6 + 2)$
$= 9$ can occur in **four** ways $(3 + 6, 4 + 5, 5 + 4, 6 + 3)$
$= 10$ can occur in **three** ways $(4 + 6, 5 + 5, 6 + 4)$
$= 11$ can occur in **two** ways $(5 + 6, 6 + 5)$
$= 12$ can occur in **one** way $(6 + 6)$

Total number of ways $= 1 + 2 + 3 + 4 + 5 + 6 + 5 + 4 + 3 + 2 + 1 = 36$

Thus the probability of x being 2 is $\frac{1}{36}$, being 3 is $\frac{2}{36}$ etc. The distribution is given in Table 5.1 below:

Table 5.1

x :	2	3	4	5	6	7	8	9	10	11	12
$p(x)$:	$\frac{1}{36}$	$\frac{2}{36}$	$\frac{3}{36}$	$\frac{4}{36}$	$\frac{5}{36}$	$\frac{6}{36}$	$\frac{5}{36}$	$\frac{4}{36}$	$\frac{3}{36}$	$\frac{2}{36}$	$\frac{1}{36}$

Probability that x is an odd number is,

$$= p(x = 3) + p(x = 5) + p(x = 7) + p(x = 9) + p(x = 11)$$

$$= \frac{2}{36} + \frac{4}{36} + \frac{6}{36} + \frac{4}{36} + \frac{2}{36} = \frac{18}{36} = \frac{1}{2} \text{ or } 50\%$$

5.3 Cumulative Distribution Function

It is a function which gives the probability of a random variable being less than or equal to a given value. In the discrete case, the cumulative distribution function is denoted by $P(x_i)$. This function implies that x takes values less than or equal to x_i.

Let us consider the following data, which pertains to the demand of an item by the customers. Total 200 customers demand has been recorded.

Table 5.2

Number of items demanded (x)	*Number of customers*	*Probability Distribution*	*Cumulative Distribution*
0	7	.035	.035
1	25	.125	.160
2	50	.250	.410
3	72	.360	.770
4	33	.165	.935
5	13	.065	1.000

The probability of zero demand is $\frac{7}{200} = .035$, while that of one item is $\frac{25}{200} = .125$, and that of two items is $\frac{50}{200} = .250$, and so on. The cumulative probability, $P(2) = p(0) + p(1) + p(2) = .035 + .125 + .250 = .410$. The probability distribution and the cumulative probability distribution are given in third and fourth columns of Table 5.2.

The probability distribution (p.m.f.) and the cumulative probability distribution (CDF) are commonly represented graphically. The data of Table 5.2, is shown plotted in Fig. 5.1. Since, p.m.f. is defined at discrete points only, the CDF is a step function type of graph.

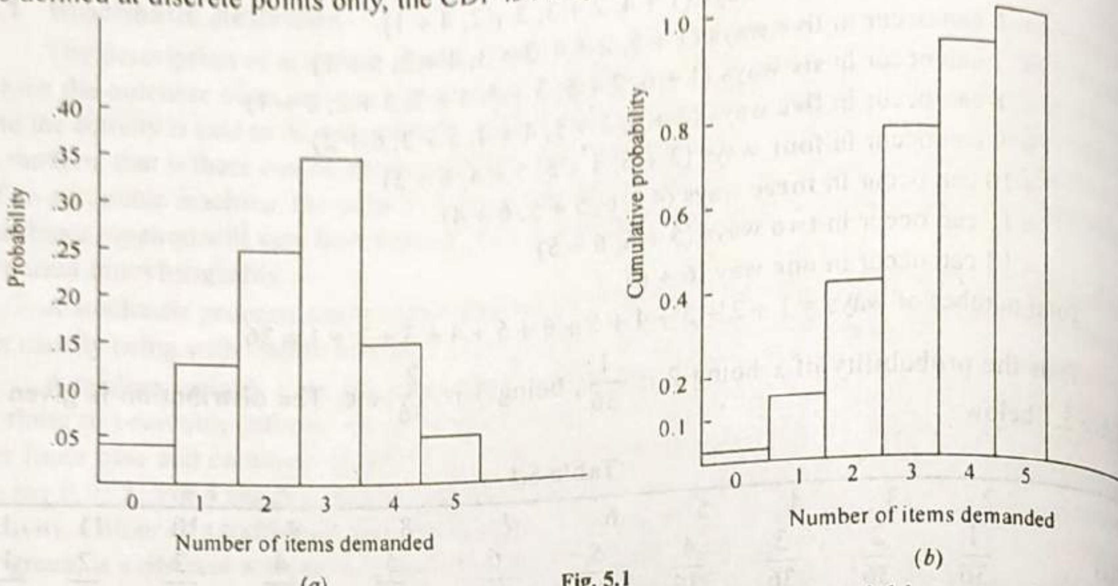


Fig. 5.1

Example 5.2. In Example 5.1, the probability distribution of a variable x, was determined. Represent the same as well as the cumulative distribution in the form of a graph.

Solution: The probability density function obtained in Example 5.1, and the CDF are as follows:

x	2	3	4	5	6	7	8	9	10	11	12
$p(x)$	$\frac{1}{36}$	$\frac{2}{36}$	$\frac{3}{36}$	$\frac{4}{36}$	$\frac{5}{36}$	$\frac{6}{36}$	$\frac{5}{36}$	$\frac{4}{36}$	$\frac{3}{36}$	$\frac{2}{36}$	$\frac{1}{36}$
$P(x)$	$\frac{1}{36}$	$\frac{3}{36}$	$\frac{6}{36}$	$\frac{10}{36}$	$\frac{15}{36}$	$\frac{21}{36}$	$\frac{26}{36}$	$\frac{30}{36}$	$\frac{33}{36}$	$\frac{35}{36}$	$\frac{36}{36}$

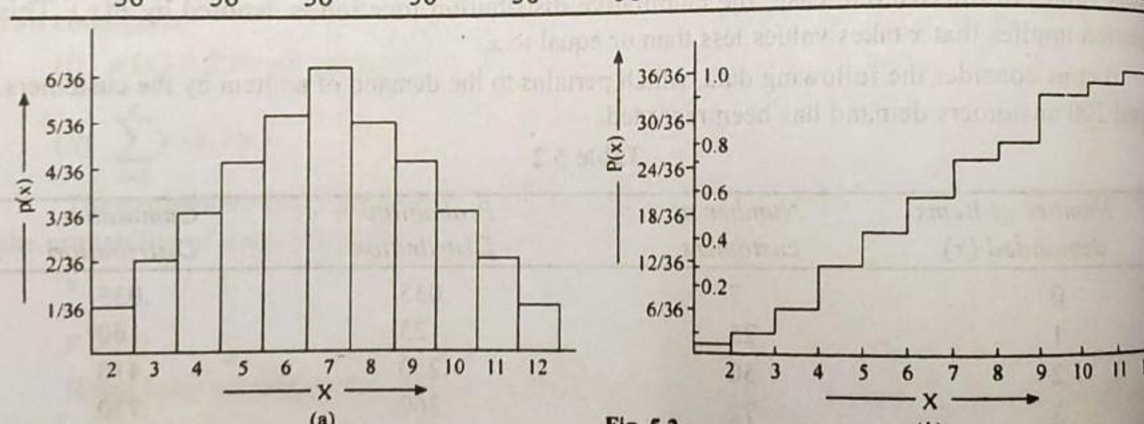


Fig. 5.2

The probability mass function p.m.f. and cummulative density function C.D.F. for the above data are given in Fig. 5.2.

5.4 Continuous Probability Functions

If the random variable is continuous and not limited to discrete values, it will have infinite number of values in any interval, howsoever small. Such a variable is defined by a function $f(x)$ called a probability density function (p.d.f.). The probability that a variable x, falls between x and $x + dx$ is expressed as $f(x)\, dx$, and the probability that x falls in the range x_1 to x_2 is given by

$$p(x) = \int_{x_1}^{x_2} f(x)\, dx$$

If x is a continuous variable on the range $-\infty$ to ∞ its p.d.f. $f(x)$ must satisfy the following two conditions :

(*i*) $f(x) \geq 0 \quad -\infty < x < \infty$ (*ii*) $\int_{-\infty}^{\infty} f(x)\, dx = 1$

Example 5.3. If x is a random variable with the following distribution,

$f(x) = xe^{-x}, x \geq 0$ show graphically its p.d.f. and C.D.F.

Solution: Values of the function $f(x)$, corresponding to various values of x are computed below.

Table 5.3

x	$f(x) = xe^{-x}$	*p.d.f.*	C.D.F.
0	0	0	0
.2	.164	.042	.042
.4	.268	.069	.111
.6	.329	.085	.196
.8	.359	.093	.289
1.0	.368	.095	.384
1.2	.361	.093	.477
1.4	.345	.089	.566
1.6	.323	.084	.650
1.8	.298	.077	.727
2.0	.271	.070	.797
2.4	.218	.056	.853
2.8	.170	.044	.897
3.2	.130	.034	.931
3.6	.098	.025	.956
4.0	.073	.619	.975
4.6	.046	.012	.987
5.2	.029	.008	.995
6.0	.015	.004	.999
	3.865		

The probability density function $f(x)$ and the cumulative distribution function $F(x)$ are given in Figs. 5.3 and 5.4 respectively.

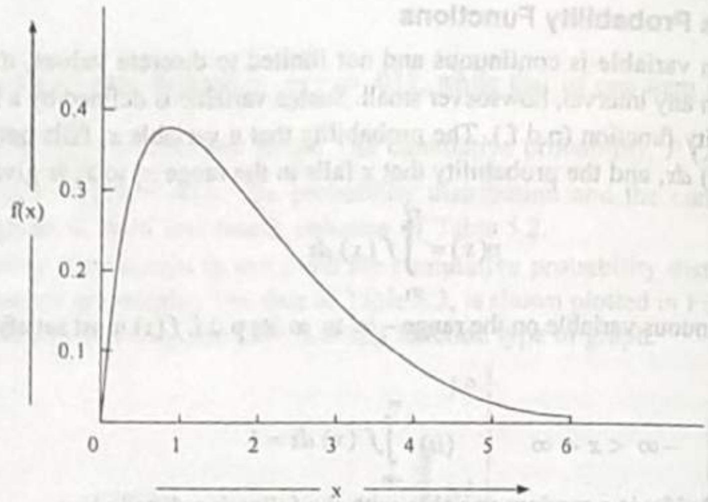


Fig. 5.3

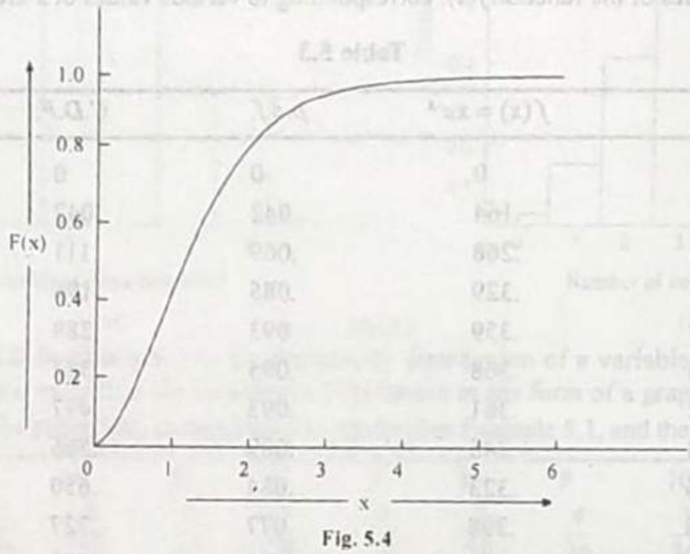


Fig. 5.4

5.5 Measures of Probability Function

The two important characteristics of a probability density function are the central tendency and dispersion.

5.5.1. Central Tendency

Three most important measures of central tendency are mean, mode and median.

Mean

If x is a random variable, then its expected value $E(x)$ itself is called the mean or average value of x, and is identified as $\bar{x}$. In the case of a discrete distribution, if there are N observations, taking the individual values x_i ($i = 1,2, \ldots, N$), the mean is given by

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{i=N} x_i$$

If the N observations are divided into I groups, where the ith group takes the value x_i and has n_i observations, then mean is given by

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{i=I} n_i x_i$$

In case the probability $p(x_i)$ of occurrence of each observation x_i is given, then the mean is given by,

$$\bar{x} = \sum p(x_i)\, x_i$$

For a continuous variable, the mean value is defined as,

$$\bar{x} = \int_{-\infty}^{\infty} x\, f(x)\, dx,$$

where $f(x)$ is the p.d.f. of x *i.e.* $\int_{-\infty}^{\infty} f(x)\, dx = 1$

Mode

Mode represents the most frequently occurring value of a random variable. In other words, when the probability density function has a peak the value of x at which the peak occurs is called mode. A distribution may have more than one peaks, that is may be multimodal (Fig. 5.5). Depending upon the number of peaks, it may be called unimodal, bimodal, trimodal etc. The highest peak, that is the most frequently occurring value is then called the mode of the distribution.

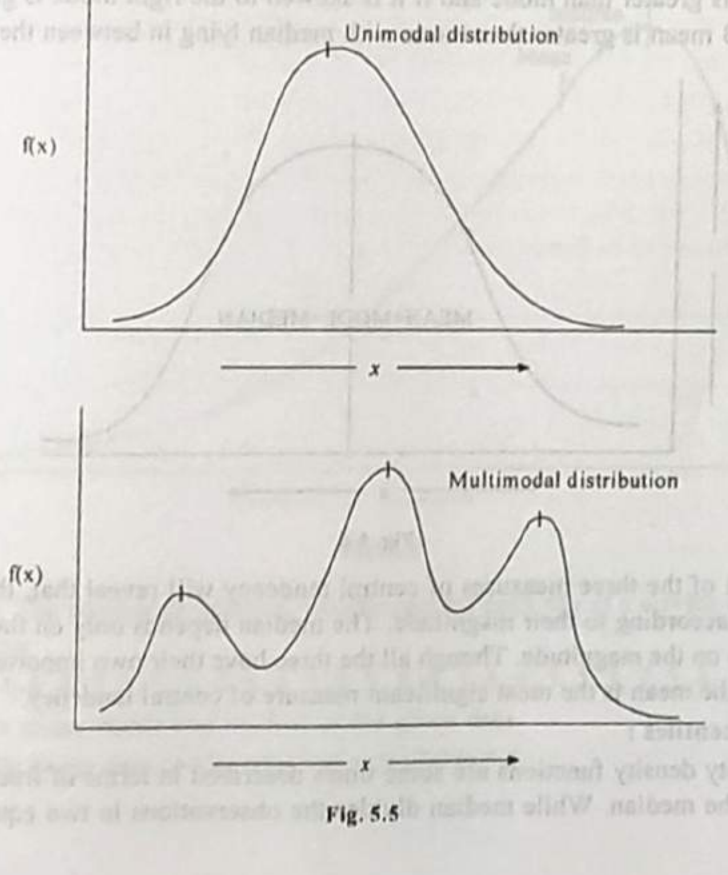


Fig. 5.5

In case of discrete distribution, mode is determined by the following inequations:

$$p\,(x = x_i) \le p\,(x = \hat{x}),\ x_i \le \hat{x}$$

and $$p\,(x = x_j) \le p\,(x = \hat{x}),\ x_j \le \hat{x}$$

For a continuous distribution, mode is determined as,

$$\frac{d}{dx}[f\,(x)] = 0$$

$$\frac{d^2}{dx^2}[f\,(x)] = 0$$

Median

Median divides the observations of the variable in two equal parts. Half the values of a random variable will fall below the median, and half above the median. For a discrete or continuous distribution of random variable x, if X denotes the median, then

$$p\,(x \le X) = p\,(x \ge X) = 0.5$$

The median is easily found from the cumulative distribution, since it is the point at which $P(x) = 0.5$. In Fig. 5.3, the median corresponds to $x = 1.28$.

If any two of the mean, mode and median are known, the third can be computed by the following relation.

$$\text{Mean} - \text{Mode} = 3\,(\text{Mean} - \text{Median})$$

The relative values of mean, mode and median depend upon the shape of the probability density function. If the probability distribution curve is symmetric about the center and is unimodal as in Fig. 5.6, then all the three mean, mode and median will coincide. If the curve is skewed to the left as in Fig. 5.7, mean is greater than mode and if it is skewed to the right mode is greater than mean (Fig. 5.8). In Fig. 5.3 mean is greater than mode with median lying in between the two.

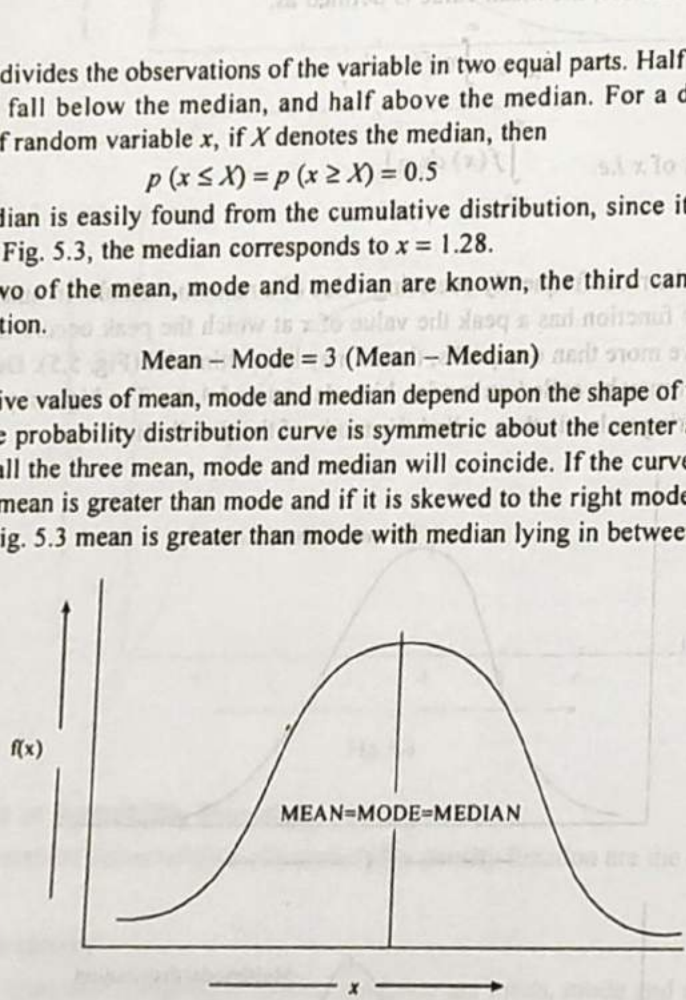


Fig. 5.6

A comparison of the three measures of central tendency will reveal that, the mean involves weighting the data according to their magnitude. The median depends only on the order, while the mode depends only on the magnitude. Though all the three have their own importance in describing a distribution, but the mean is the most significant measure of central tendency.

Fractiles and Percentiles :

The probability density functions are some times described in terms of fractiles, which are a generalization of the median. While median divides the observations in two equal parts, fractiles divide the data into a number of parts. For example, ten fractiles, divide the data into ten parts as the first tenth, second tenth and so on. The fractiles taken as percentage are called percentiles.

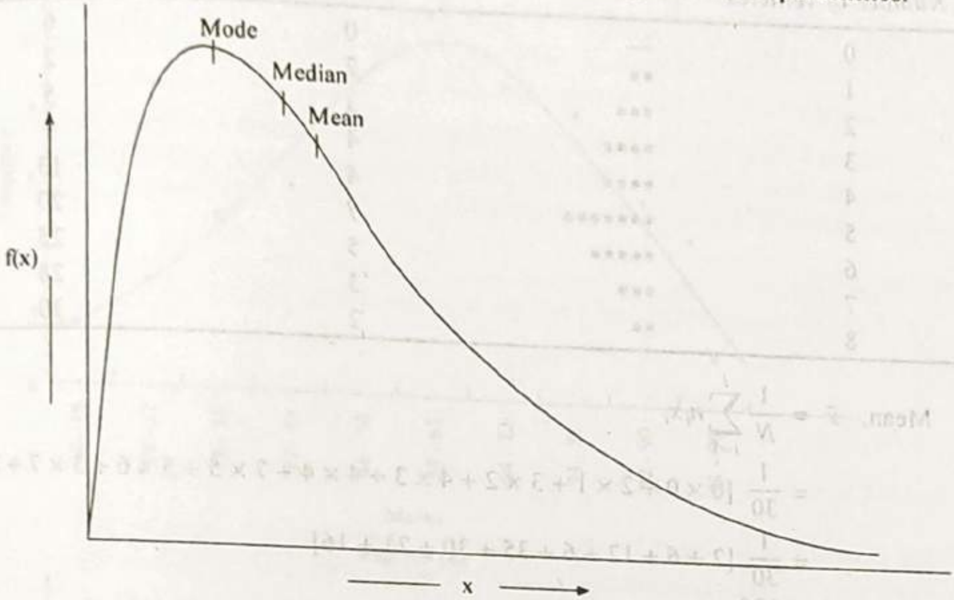


Fig. 5.7

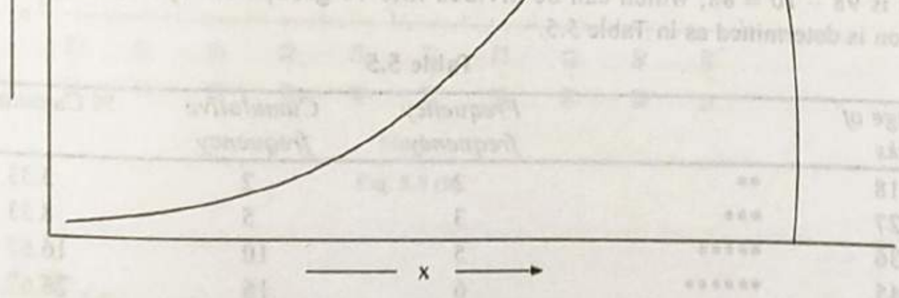


Fig. 5.8

Example 5.4. The number of vehicles, which reported each day at a service station during June 1999, is given below.

3, 2, 6, 1, 5, 4, 6, 8, 7, 5, 3, 2, 4, 6, 4, 5, 2, 5, 3, 6, 5, 4, 6, 5, 5, 1, 7, 3, 8, 7.

Determine the mean, mode and median of the given data.

Solution : The given data can be grouped as in Table 5.4.

Table 5.4

Number of vehicles		*Frequency*	*Cumulative frequency*
0	—	0	0
1	**	2	2
2	***	3	5
3	****	4	9
4	****	4	13
5	*******	7	20
6	*****	5	25
7	***	3	28
8	**	2	30

$$\text{Mean, } \bar{x} = \frac{1}{N}\sum_{i=1}^{I} n_i x_i$$

$$= \frac{1}{30}[0\times 0 + 2\times 1 + 3\times 2 + 4\times 3 + 4\times 4 + 7\times 5 + 5\times 6 + 3\times 7 + 2\times 8]$$

$$= \frac{1}{30}[2 + 6 + 12 + 6 + 35 + 30 + 21 + 16]$$

$$= \frac{138}{30} = 4.6$$

Mode, the most frequently occurring value is 5 or the mode $\hat{x} = 5$.

Median: It is the value of the number of vehicles corresponding to the cumulative frequency of 15, that is half the total number of vehicles, which in this case is slightly more than 4.

Example 5.5. Marks obtained by 60 students in the subject of Operations Research are given below. Plot the frequency distribution curve. Determine the mean, mode and median of the marks.

55, 25, 36, 28, 15, 48, 58, 66, 84, 32, 10, 46, 54, 65, 30, 59, 81, 87, 69
78, 49, 83, 50, 61, 40, 71, 60, 59, 73, 30, 80, 90, 76, 59, 62, 51, 68, 66, 38, 24,
38, 43, 63, 60, 40, 75, 71, 81, 78, 40, 20, 49, 98, 53, 61, 70, 72, 87, 85, 93.

Solution: The lowest marks obtained are 10 while the highest marks obtained are 98. The range of marks is 98 − 10 = 88; which can be divided into 10 groups of say 9 each. The frequency distribution is determined as in Table 5.5.

Table 5.5

Range of marks		*Frequency frequency*	*Cumulative frequency*	*% Cumulative*
10–18	**	2	2	3.33
19–27	***	3	5	8.33
28–36	*****	5	10	16.67
37–45	******	6	16	26.67
46–54	********	8	24	40.00
55–63	***********	11	35	58.33
64–72	*********	9	44	73.33
73–81	********	8	52	86.67
82–90	******	6	58	96.67
91–99	**	2	60	100.00

The probability density function is plotted in Fig. 5.9

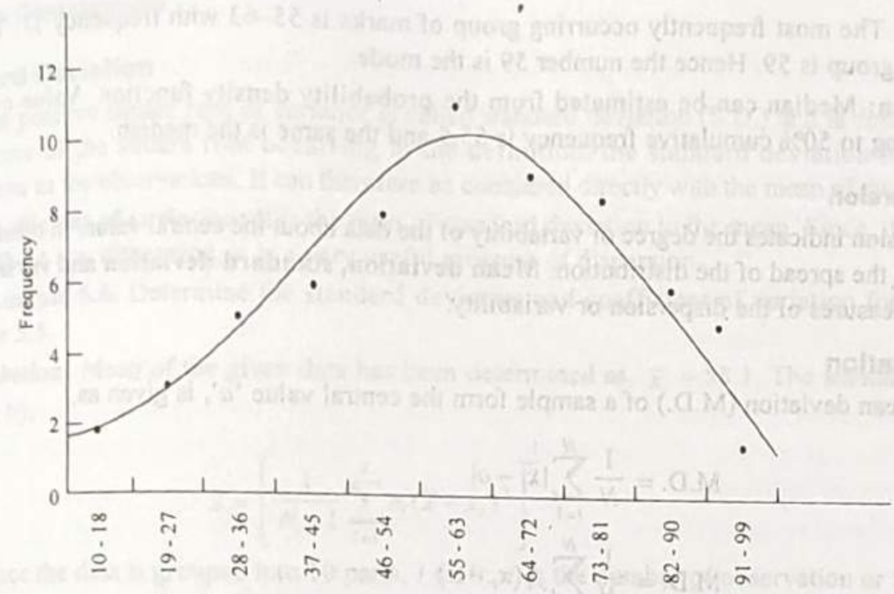


Fig. 5.9 (a)

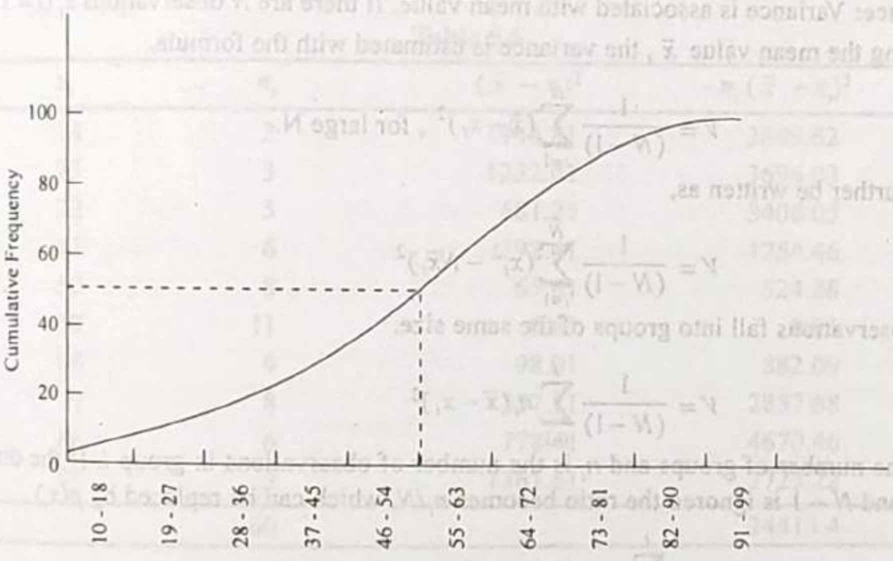


Fig. 5.9 (b)

Mean

$$\bar{x} = \frac{1}{60}\sum_{i=1}^{I} x_i n_i$$

$$\bar{x} = \frac{1}{60}\sum_{i=1}^{I} x_i n_i =$$

$$[14 \times 2 + 23 \times 3 + 32 \times 5 + 41 \times 6 + 50 \times 8 + 59 \times 11 + 68 \times 9 + 77 \times 8 + 86 \times 6 + 95 \times 2]$$

$$= \frac{3486}{60} = 58.1$$

Mode: The most frequently occurring group of marks is 55–63 with frequency 11. The mid point of the group is 59. Hence the number 59 is the mode.

Median: Median can be estimated from the probability density function. Value of marks corresponding to 50% cumulative frequency is 55.5 and the same is the median.

5.5.2 Dispersion

Dispersion indicates the degree of variability of the data about the central value. In other words, it represents the spread of the distribution. **Mean deviation, standard deviation and variance** are important measures of the dispersion or variability.

Mean Deviation

The mean deviation (M.D.) of a sample form the central value 'a', is given as,

$$\text{M.D.} = \frac{1}{N}\sum_{i=1}^{N}|x_i - a|$$

$$\text{M.D.} = \frac{1}{N}\sum_{i=1}^{N} f_i(x_i - a)$$

The central value 'a' may be mean, mode or median.

Variance: Variance is associated with mean value. If there are N observations x_i ($i = 1, 2 \ldots\ldots, N$) and having the mean value $\bar{x}$, the variance is estimated with the formula.

$$V = \frac{1}{(N-1)}\sum_{i=1}^{N}(\bar{x} - x_i)^2 \text{, for large N.}$$

which can further be written as,

$$V = \frac{1}{(N-1)}\sum_{i=1}^{N}(x_i^{\,2} - N\bar{x}_i)^2$$

when the observations fall into groups of the same size.

$$V = \frac{1}{(N-1)}\sum_{i=1}^{I} n_i(\bar{x} - x_i)^2$$

where I is the number of groups and n_i is the number of observations in group i. If the difference between N and $N - 1$ is ignored the ratio becomes n_i/N, which can be replaced by $p(x_i)$

$$V = \sum_{i=1}^{I} p(x_i)(\bar{x} - x_i)^2$$

Since $\sum_{i=1}^{I} p(x_i) = 1$ and $\sum_{i=1}^{I} p(x_i)\, x_i = \bar{x}$

$$V = \sum_{i=1}^{I} p(x_i)\, x_i^{\,2} - \bar{x}^2$$

For a continuous variable,

$$V = \int_{-\infty}^{\infty} f(x)\, x^2 dx - \bar{x}^2$$

Variance is also called **second moment of dispersion.**

Standard Deviation

The positive square root of variance is called standard deviation (S.D.) and is denoted by σ or S. Because of the square root occurring in the definition, the standard deviation has the same dimensions as the observations. It can therefore be compared directly with the mean of the distribution.

Coefficient of variation: It is the ratio of standard deviation to the mean. Since, it is a relative term without any dimension, it is a very useful measure of dispersion.

Example 5.6. Determine the standard deviation and coefficient of variation for the data of Example 5.5.

Solution: Mean of the given data has been determined as, $\bar{x} = 58.1$. The standard deviation is given by,

$$S = \left[\frac{1}{N-1}\sum_{i=1}^{I} n_i(\bar{x}-x_i)^2\right]^{\frac{1}{2}}$$

Since the data is grouped into 10 parts, $I = 10$; n_i is the number of observation or frequency in each group. $N = \sum_{i=1}^{10} n_i$. The computations are given in Table 5.6.

Table 5.6

x_i	n_i	$(\bar{x}-x_i)^2$	$n_i(\bar{x}-x_i)^2$
14	2	1944.81	3889.62
23	3	1232.01	3696.03
32	5	681.21	3406.05
41	6	292.41	1754.46
50	8	65.61	524.88
59	11	0.81	8.91
68	9	98.01	882.09
77	8	357.21	2857.68
86	6	778.41	4670.46
95	2	1361.61	2723.22
	60		24413.4

$$S = \sqrt{\frac{24413.4}{(60-1)}} = 20.34$$

$$\text{Coefficient of variation} = \frac{S}{\bar{x}} = \frac{20.34}{58.1} = 0.35$$

5.6 Generation of Random Variates

In Gambling game of Example 2.2, we have discussed the generation of random observations from discrete distribution and in Example 2.3 of Numerical Integration, from a continuous distribution. There are a large number of probability density functions, which describe the various random phenomena. Here we will discuss the generation of random variates from some continuous and discrete distributions, which are commonly encountered in simulation.

5.7 Bernoulli Trial

Bernoulli Trial is a trial which results in two outcomes, usually called 'a success' and 'a failure'. Bernoulli Process is an experiment performed repeatedly, which has only two outcomes, say success (S) and failure (F). Here the probability that an event will occur (success) remains constant. In rolling a die, the probability of face i (1, 2, ..., 6) coming up is always $\frac{1}{6}$. The probability of a head while tossing a coin is 0.5 in each flip. Firing a target (hit or miss), contesting an election (win or lose) etc. are examples of Bernoulli trials. It is very easy to simulate a Bernoulli case. For example, if the probability that a machine will fail in the next one hour is 0.3, we just draw a random number, if it is less than or equal to 0.3, machine will fail otherwise it will not fail in the next hour.

5.8 Binomial Distribution

If 'n' Bernoulli trials are made at a time, then the distribution of the number of successes, is the Binomial distribution. For example, if a sample of N balls is drawn from an infinitely large population of balls, having a proportion p of say red balls, then the distribution of red balls in each sample is given by Binomial distribution.

The Binomial probability mass function of a variable x is given by,

$$f(x) = \binom{n}{x} p^x (1-p)^{n-x}, \quad x = 0, 1, 2, ..., n.$$

where p is the probability of success in an independent trial and thus $1-p$ is the probability of failure. x, a positive integer represents the number of successes in n independent trials. To generate a random variate x, n random numbers are generated. A random number less than or equal to p i.e., $r \le p$ gives a success and the total numbers of such successes gives the value of x.

The properties of Binomial distribution are

$$\text{Mean } \bar{x} = n.p$$

$$\text{Variance } \sigma^2 = n.p.(1-p)$$

A computer program for generating the random variates (x) from a Binomial distribution with parameters n (number of trials) and p (probability of success in a trial) is given below.

```
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
main()
  { /* To generate variates from a BINOMIAL distribution having
       parameters - n (number of trials) and p (probability of success
       in a trial)*/
       int n,i,x,k,m,nn,nt;
       float y,p;
       printf("\n The parameter n (number of trials) nt=");
       scanf("%d",&nt);
       printf("\n Enter the value of p(<1.0) the probability of succss=");
       scanf("%f",&p);
       printf("\n Number of variates to be generated (nn)=");
       scanf("%d",&nn);
       printf("\n Number of trials=%d\n Probability of success=
%4.2f",nt,p);
       printf("\n Number of variates= %d \n",nn);
       printf("\n Values of variate x:");
       for(m=1;m<=nn;++m) {
```

```
    x=0; n=nt;
    for(i=1;i<=n;++i) {
    y=rand()/32768.0;
    if(y < p) x=x+1;
          }
        printf(" %d",x);
        }

  }
```

The output of this program is given below:

Number of trials : 10

Probability of success = 0.6

Number of values : 20

Values of x : 8 6 3 7 7 7 8 9 6 6 8 4 3 7 3 6 5 5 4 5

5.9 Negative Binomial Distribution

In Binomial distribution, the number of trials N is fixed and the number of successes (or failures) determines the random variate. In negative Binomial distribution, the random variable is given by the number of independent trials to be carried out until a given number of successes occur. If X denotes the number of trials to obtain the fixed number K of successes then,

Probability of X trials until K successes occur

= Probability of $(K-1)$ successes in $(X-1)$ trials × probability of a success in Xth trial.

$$p(X) = \binom{X-1}{K-1} P^{K-1}((1-P)^{X-K}P)$$

$$= \binom{X-1}{K-1} P^{K}(1-P)^{X-K}$$

where P is the probability of success in a trial.

When K is an integer, this distribution is also called a **Pascal distribution.** When K is equal to one, it is called a **geometric distribution.**

A computer program for generating random variable 'x' from a Negative Binomial distribution with parameters k (numbers of successes) and p (Probability of success in a trial) is given below.

```
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
main()
{    /* To generate  variates  from a NEGATIVE BINOMIAL (Pascal)
     distribution having parameters k (number of successes)
     and p (probability of success in a trial   */

     int x,j,nn,m,s,k;
     float p,r;

     printf("\n Enter the values of p(<1.0)=");
     scanf("%f",&p);
```

```
printf("\n Enter the values of k (>=1)=");
scanf("%d",&k);
printf("\n Value of Papameter k= %d",k);
printf("\n Probability of success in a trial=%4.2f",p);
printf("\n Number of variates to be generated=");
scanf("%d",&nn);
printf("\n Values of variates x:");

for(j=1;j<=nn;++j) {
x=0; s=0 ;

while(s<k) {
r=rand()/32768.0;
if(r<=p)     {s=s+1;}
x=x+1;
/* printf("\n %5.1f %d %d",x,s,n);*/
}
printf("%4d,",x);
}

}
```

An output obtained from the above program is as under :

Value of parameter $k = 3$

Probability of success in a trial = 0.45

Number of variates to be generated = 15

Values of variates x : 3, 3, 6, 5, 18, 7, 4, 8, 6, 4, 3, 5, 3, 4, 3

5.10 Geometric Distribution

It is a special case of Negative Binomial distribution with $K = 1$. Independent Bernoulli trials are performed until a success occurs. If X is the number of trials carried out to get a success, then X is the geometric random variable.

$$p(X) = P.(1 - P)^{X-1}$$

Mean $= \dfrac{1}{P}$ and variance $= \dfrac{1-P}{P^2}$, where P is the probability of success in a trial.

5.11 Hypergeometric Distribution

In case of Binomial distribution, the probability of success is same for all trials. In other words, the population with proportion P of the desired events is taken to be of infinite size. However, if this population is finite and samples are taken without replacement, the probability of proportion of desired events in the population will vary from sample to sample. The distribution, which describes the distribution of success X, in such a case, is called the hypergeometric distribution.

If N is sample size and M is the finite size of population to start with, and P is the probability of success in a trial in the beginning, then the hypergeometric probability function is given by,

$$p(X) = M \cdot \frac{P!}{X!}(M \cdot P - X)! \frac{(M \cdot Q)!(MQ - N + X)}{(N-X)! \dfrac{M!}{N!}(M - N)!}$$

where $Q = 1 - P$

The mean is given by $\bar{x} = N.P$ and variance $\sigma^2 = \frac{M-N}{M-1} N(P \cdot Q)$

5.12 Poisson Distribution

Poisson distribution is the discrete version of the Exponential distribution. Let x be a random variable which takes non-negative integer values only, then the probability mass function,

$$p(x) = \frac{\lambda^x e^{-\lambda}}{x!}$$

is called the Poisson distribution with parameter λ, where $\lambda > 0$

The properties of Poisson distribution are

$$\text{Mean} = \lambda$$

$$\text{Variance} = \lambda$$

One of the first applications of this distribution was in representing the probable number of Prussian cavalry troopers killed each year by being kicked in the head by horses. This distribution finds applications in a wide variety of situations where some kind of event occurs repeatedly but haphazardly. Number of customers arriving at a service station, number of fatal accidents per year per kilometer of highway, number of typographical errors on a page, number of flaws per square yard of cloth, number of α-particles emitted by a radioactive substance etc. are some of the situations.

To generate an observation X, a random number is generated. The cumulative probabilities of X for values 0, 1, 2, ... etc. are computed and each time compared with the random number. As soon as the cumulative probability becomes greater than or equal to the random number, the value of X is returned.

A computer program subroutine for generating the Poisson observations is given below :

```
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
main()
  {  /*  To generate  variates  from a POISSON distribution
     given the 'mean' that is parameter lemda(>0).
     fact - factorial;
     prob - probability and cumprob - cumulative probability  */

     int  k,j,x,nn;
     float lemda,cumprob,fact,prob,y,mean ;
     /* lemda=5.0;*/
     printf("\n Value of mean (lemda)=");
     scanf("%f",&lemda);
     printf("\n Number of variates to be generated=");
     scanf("%d",&nn);

     mean=lemda;

     for(j=1;j<=nn;++j) {
     fact=1.; x=0;
     cumprob=0.;
```

```
y=rand()/32768.0;
while(y > cumprob) {
prob= pow( 2.718282,-mean) * pow(mean,x)/fact  ;
cumprob=cumprob+prob;
x=x+1;
fact=fact*x;
/*printf("\n %4.3f %4.2f %4.2f %4.2f",y, prob,cumprob,fact);*/
}
printf(" %4d",x);
}

}
```

An output of the above program is given below:

Value of mean (lambda) = 5

Number of variates to be generated = 15

Values of variates x : 2 1 5 2 5 4 6 4 7 10 5 6 3 7 6

5.13 Empirical Distribution

It is a discrete distribution constructed on the basis of experimental observations. The outcomes of an activity are observed over a period of time, and the probability of each outcome is established from the observations. For example, the demand of an item as observed over a long period is 1, 2, 3 or 4 pieces per hour. Out of say 400 observations, 40 times the demand is for one piece, 160 times for 2 pieces, 180 times for three pieces and 20 times for four pieces. In other words the probabilities of demand being 1, 2, 3 or 4 pieces are $\frac{40}{400} = .1$, $\frac{160}{400} = .4$, $\frac{180}{400} = .45$ and $\frac{20}{400} = .05$ respectively. This discrete probability distributions is an empirical distribution.

The general approach of working with such a distribution is called the integral inverse approach. Integration, the process of adding the values is statistically called cumulative distribution. A variate is generated using a random number, that is taking inverse of the law.

If $F(x)$ represents C.D.F. of variable x, and a random number r is such that $F(i-1) \le r \le F_i$ then the discrete variable x assumes the value x_i

i	1	2	3	....	n
x_i	x_1	x_2	x_3	....	x_n
$F(x)$	F_1	F_2	F_3	....	F_n

A computer program in C language for generating a random variable x from a given Empirical distribution is given below.

```
#include<stdio.h>
#include<stdlib.h>
main()
  {
  /* To generate random variables from an EMPIRICAL distribution.*/

   int i,j,k,n,nn;
   float  rnd,y;
   n=4;
```

```
/*  Values of x[i] and cumulative probabilitiesf[i] with i= 0,4 */
    float x[] ={1.5,2.5,4.0,5.5,6.0};
    float p[]= {0.0,0.1,0.25,0.65,0.90,1.0};
    printf("\n x[i] =       ");
    for(i=0;i<=n;++i) {printf("%6.2f",x[i]); }
    printf("\n p[i]=");
    for(i=0;i<=n+1;++i) {printf("%6.2f",p[i]); }
    printf("\n Numbe of variates to be generateds(nn)=");
    scanf("%d",&nn);
    printf("\n Values of variates x:\n");
    for(k=1;k<=nn;++k) {

    rnd=rand()/32768.0;
    for(i=0; i<=n;++i) {
    j=i+1;
    if((rnd >p[i] ) && ( rnd <= p[j]))  y= x[i];

/*  printf("\n%d %d %5.4f %5.4f x=%5.4f %d",i,j,pr[i],pr[j],x,te);*/
        }

            printf("%5.1f",y);
            }

    }
```

An output obtained from this program is given below.

x[i]	=		1.50	2.50	4.00	5.50	4.00
p[i]	=	0.00	0.10	0.25	0.65	0.90	1.00

Number of variates to be generated = 12

Values of the variate x : 1.5 1.5 4.0 1.5 4.0 2.5 4.0 2.5 5.5 6.0 4.0 4.0

5.14 Continuous Distribution

A random variable can be generated from any of the continuous distributions by making use of the cumulative density function of the variable. If $F(x)$ is the C.D.F. of variable x, then the new random variable,

$$Y = F(x)$$

is uniformly distributed over 0,1 interval. If a random number r is generated, then

$$r = F(x)$$

or

$$x = F^{-1}(r)$$

The generation of random observations from some important continuous distributions is given below.

5.15 Normal Distribution

There are two probability laws, the normal probability and exponential probability, that describe most of the behavior that can be observed in real life systems. There are many other probability laws derived from these two, but they are used only when finer precision is needed in simulation.

The normal distribution is also called the Gaussian distribution, after the mathematician who first described it. Because of its bell type shape, normal distribution is also called the bell curve. Normal distribution is used most frequently to describe the distributions as the marks obtained by a class, dimensions of parts made on a machine, number of light bulbs which fuse per time period, heights of male or female adults etc.

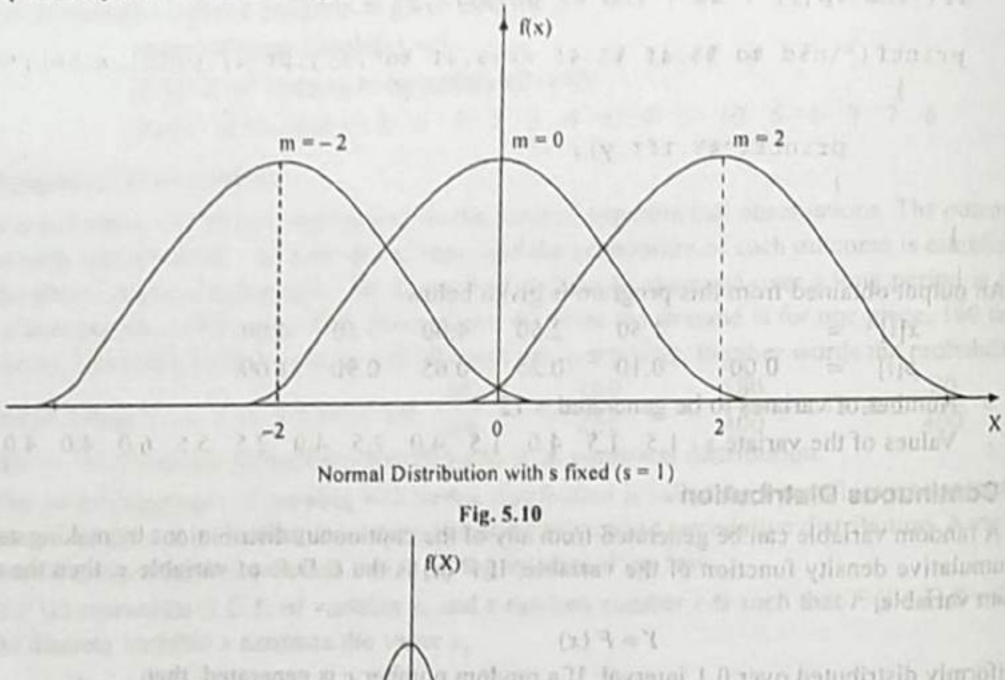
The density function of the normal curve is given by

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right], -\infty < x < \infty$$

where μ is mean of the distribution and σ is standard deviation.

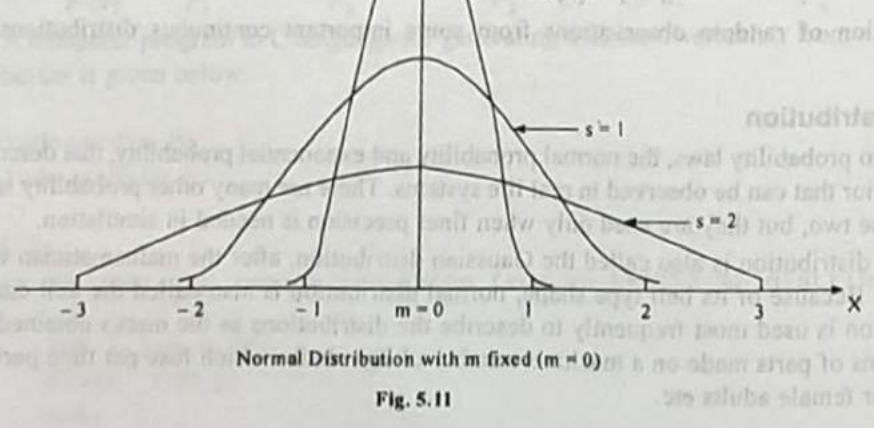
The parameter μ can take any real value, while standard deviation s is always positive. The shape of a normal curve depends upon the values of μ and σ. The effect of variations in the values of μ and σ is illustrated in Fig. 5.10 and Fig. 5.11.

The normal random variates can be generated by a number of methods. The commonly used methods are developed from the central limit theorem, which states that the probability distribution of the sum of n independent and identically distributed random numbers r_i with mean μ_i and variance σ_i^2 $(i = 1, n)$, approaches the normal distribution with mean $\sum_{i=1}^{n} \mu_i$ and variance $\sum_{i=1}^{n} \sigma_i^2$ asymptotically as N becomes large.



Normal Distribution with s fixed (s = 1)

Fig. 5.10



Normal Distribution with m fixed (m = 0)

Fig. 5.11

Let $r_1, r_2, \ldots\ldots, r_n$ be the random numbers ($0 < r_i < 1$) and

$$V = \sum_{i=1}^{n} r_i$$

Then $E(V) = \dfrac{n}{2}$

$$\text{Var}(V) = \sum_{i=1}^{n} \text{var}(r_i) = \frac{n}{12}$$

Thus V is asymptotically normal with mean $\dfrac{n}{2}$ and variance $\dfrac{n}{12}$.

Let
$$Z = \frac{V - \dfrac{n}{2}}{\sqrt{\dfrac{n}{12}}}$$

which is normal with zero mean and unit variance.

Now any random variate y, corresponding to the above n random numbers can be obtained from the expression.

$$\frac{y-\mu}{\sigma} = \frac{V - \dfrac{n}{2}}{\sqrt{\dfrac{n}{12}}}$$

or
$$y = \mu + \frac{\sigma}{\sqrt{\dfrac{n}{12}}}\left(V - \frac{n}{2}\right) = \mu + \frac{\sigma}{\sqrt{\dfrac{n}{12}}}\left(\sum_{i=1}^{n} r_i - \frac{n}{2}\right)$$

Since, according to Central Limit Theorem, the normality is approached quite rapidly even for small values of n, the value $n = 12$ is commonly used in practice.

$\therefore$
$$y = \mu + \sigma\left(\sum_{i=1}^{12} r_i - 6\right),\ i = 1, 2, \ldots\ldots, 12$$

A computer program to generate random variable x from a normal distribution with mean 'mue' and standard deviation 'sigma' is given below:

```
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
main()
  {
    /* To generate random variable from a Normal distribution
     having mean mue and standard deviation sigma.*/
     int i,j,m,nn;
     float t,sum,x,mue,sigma;
     printf("\n Enter the values of mue ");
     scanf("%f",&mue);
     printf("\n Enter the values of sigma");
     scanf("%f",&sigma);
     printf("\n Enter the number of variables to be generated nn=");
```

```
scanf("%d",&nn);
printf("\n   mue=%4.2f   sigma=%4.2f   nn=%4d",mue,sigma,nn);
printf("\n Values of variable x:");
printf("\n");
for(m=1;m<=nn;++m) {
sum=0.;
for(i=1;i<=12;++i) {
x=rand()/32768.0 ;
sum=sum+x;
}
t=mue+sigma*(sum-6.);
printf(" %6.2f",t);

}

}
```

An output of this program is given below.

Mue = 10.0 sigma = 1.5 $nn = 10$

Values of variate x: 7.09 10.42 12.45 8.89 9.33 7.92 7.95 9.11 9.93 10.77

5.16 Exponential Distribution

The exponential distribution is used to represent an activity where most of the events take place in a relatively short time, while there are a few which take very long times. The p.d.f. for this distribution is shown in Fig. 5.12.

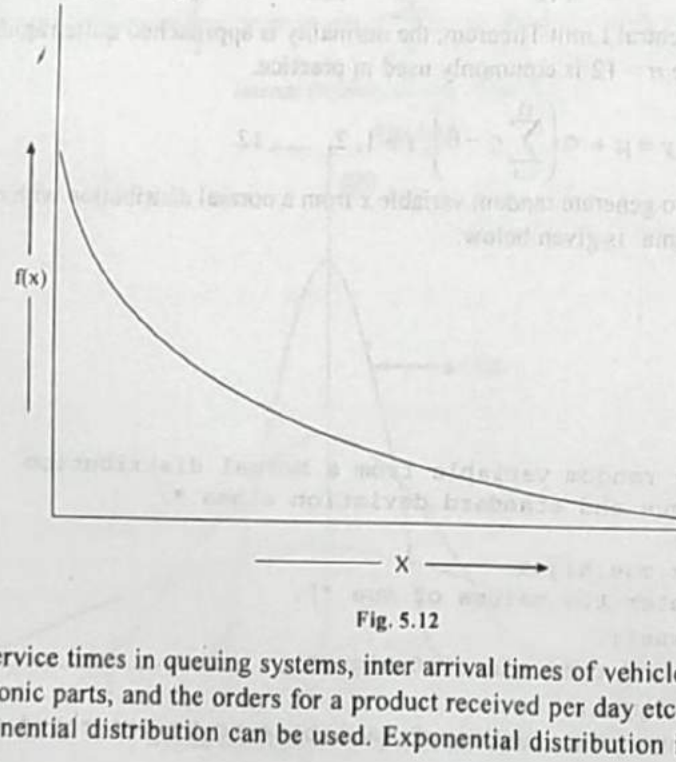


Fig. 5.12

The service times in queuing systems, inter arrival times of vehicles on a highway, the life of some electronic parts, and the orders for a product received per day etc. are some of the examples where exponential distribution can be used. Exponential distribution is analogous to Geometric distribution. While geometric distribution represents a variable as the number of customers arriving up to a specific time, the exponential distribution gives the time of next arrival of the customer.

The exponential p.d.f. is given by

$$f(t) = \begin{cases} \mu e^{-\mu t}, t > 0 \\ 0, \text{ otherwise} \end{cases}$$

The random variate from this distribution is generated by drawing a random number, r.

$$r = f(x) = \int_0^x \mu e^{-\mu x} dx = 1 - e^{-\mu x}$$

$$x = -\frac{1}{\mu} \ln(1 - r) = -\frac{1}{\mu} \ln r$$

The above expression is justified since $1 - r$ is also a random number and r and $1 - r$ are equally likely.

5.17 Erlang Distribution

There is a class of distribution functions named after A.K. Erlang, who found these distributions to be representative of certain types of telephone traffic. If there are k independent random variables $v_i (i = 1, 2, \ldots\ldots, k)$, having the same exponential distribution,

$$f(v_i) = \mu k e^{\mu k v_i}$$

with $v_i > 0$, $\mu > 0$ and k a positive integer then

$$V = \sum_{i=1}^{k} v_i \text{ has the Erlang distribution}$$

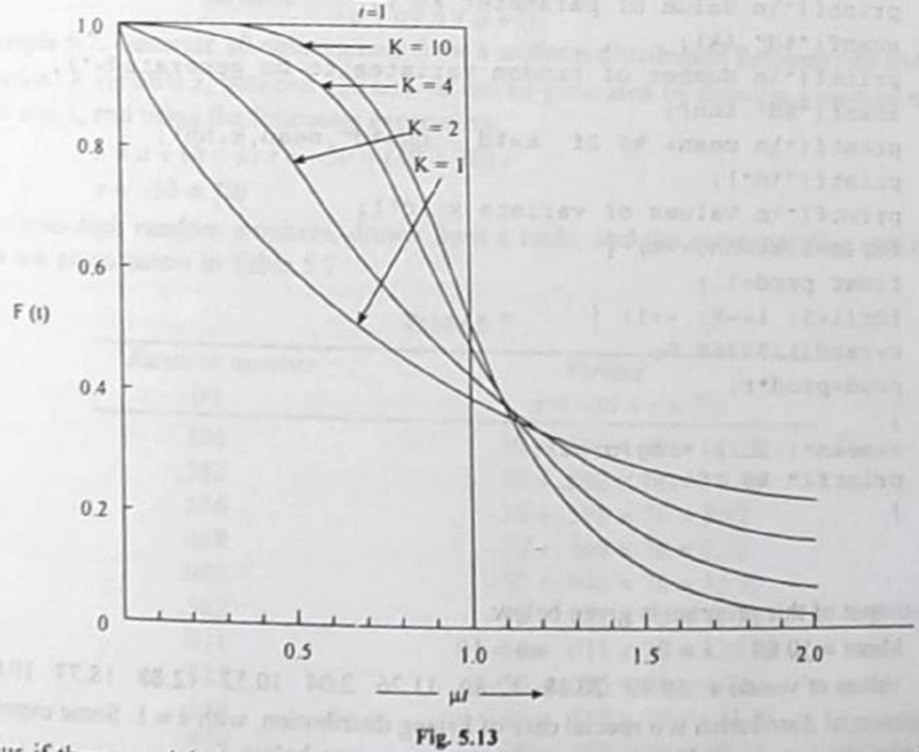


Fig. 5.13

Thus if there are k independent variates having the same exponential distribution, then their combined distribution is called Erlang distribution. Fig. 5.13 illustrates the Erlang distribution for several values of k. The parameter k is called the Erlang shape factor.

A random variate from Erlang distribution can be obtained by obtaining k random variates from the exponential distribution and adding them.

Thus, if $$v_i = -\frac{1}{\mu k} \ln r_i,\ i = 1, 2 \ldots\ldots k.$$

Then $$v = \sum_{i=1}^{k} v_i = \sum_{i=1}^{k} -\frac{1}{\mu k} \ln r_i = -\frac{1}{\mu k} \ln \prod_{i=1}^{k} r_i$$

A computer program in C language for generating Erlang variates is given below:

```
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
main()
    {
     /*To generate EARLANG distributed variates given the values of
      mean and earlang parameter k.
      If the value of k is taken as 1, than this program generates
      EXPONENTIALLY distributed variates*/

      int  i,j,k,m,nn;
      float x,r,mean;
      printf("\n Value of mean =");
      scanf("%f",&mean);
      printf("\n Value of parameter k=");
      scanf("%d",&k);
      printf("\n Number of random variates to be generated=");
      scanf("%d",&nn);
      printf("\n mean= %6.2f  k=%d   nn=%d",mean,k,nn);
      printf("\n");
      printf("\n Values of variate x:\n");
      for(m=1;m<=nn;++m) {
      float prod=1.;
      for(i=1; i<=k; ++i) {
      r=rand()/32768.0;
      prod=prod*r;
      }
      x=mean*(-1./k)*log(prod);
      printf(" %6.2f",x);
      }
    }
```

An output of this program is given below.

Mean = 10.00 $k = 2$ $nn = 10$

Values of variate x : 50.40 20.48 12.80 11.26 2.04 10.52 12.88 18.77 10.02 3.23

Exponential distribution is a special case of Erlang distribution, with $k = 1$. Some exponentially distributed variates generated by the above program are given below.

Mean = 10.00 $k = 1$ $nn = 10$

45.51 55.30 10.93 34.03 10.34 19.27 6.22 16.31 3.56 0.51

5.18 Uniform Distribution

The uniform probability law states that all the values in a given interval are equally likely to occur. Suppose, we say that service times are uniformly distributed between 30 and 80 seconds. It means that a service will not take less than 30 seconds and more than 80 seconds. On the average service time will be 55 seconds.

Let a variable x be uniformly distributed in the interval (a, b), that is all the values of x between a and b, $(b > a)$ are equally likely, then

$$x = a + (b - a)\, r$$

where r is a random number between 0 and 1. The uniform distribution is also called Rectangular distribution or Homogeneous distribution.

5.19 Beta Distribution

The beta distribution is an extension of Erlang distribution and exists only between the limits 0 and 1. A random variable x is said to have beta distribution with parameters p and q, if its density function is given by:

$$f(x) = \begin{cases} \dfrac{x^{p-1}(1-x)^{q-1}}{\beta(p,q)}, & \text{if } 0 < x < 1 \\ 0, & x \le 0 \text{ and } x \ge 1 \end{cases}$$

$$\text{with } p > 0,\ q > 0.$$

Properties of the beta distribution are:

$$\text{Mean} = \frac{p}{p+q}$$

$$\text{Variance} = \frac{p(p+1)}{(p+q)(p+q+1)}$$

Example 5.7. Generate 10 observations from a uniform distribution between –30 and 40.

Solution: A variable x, between –30 and 40 can be generated by drawing a random number r between 0 and 1, and using the following expression:

$$x = a + (b - a)\, r = -30 + (40 + 30)\, r$$

or

$$x = -30 + 70r$$

Ten three-digit random numbers, drawn from a table, and the corresponding values of the variable x are given below in Table 5.7.

Table 5.7

Random number (r)	*Variate* $x = -30 + r \times 70$
.396	$-30 + .396 \times 70 = -2.28$
.582	$-30 + .582 \times 70 = 10.74$
.556	$-30 + .556 \times 70 = 8.92$
.468	$-30 + .468 \times 70 = 2.76$
.940	$-30 + .940 \times 70 = 35.80$
.382	$-30 + .382 \times 70 = -3.26$
.011	$-30 + .011 \times 70 = -29.23$
.525	$-30 + .525 \times 70 = 6.75$
.117	$-30 + .117 \times 70 = -21.81$
.579	$-30 + .579 \times 70 = 10.53$

Example 5.8. A random variable X has the following empirical distribution.

X	:	1	2	4	6	8	10
$f(x)$	:	.10	.20	.25	.20	.15	.10

Plot the cumulative distribution and find the values of X corresponding to the following two-digit random numbers, 05, 45, 62, 93.

Solution: The cumulative distribution for variable X is computed below in Table 5.8, and is shown in Fig. 5.14.

Table 5.8

X	$f(x)$	$F(x)$
1	.10	0.10
2	.20	0.30
4	.25	0.55
6	.20	0.75
8	.15	0.90
10	.10	1.00

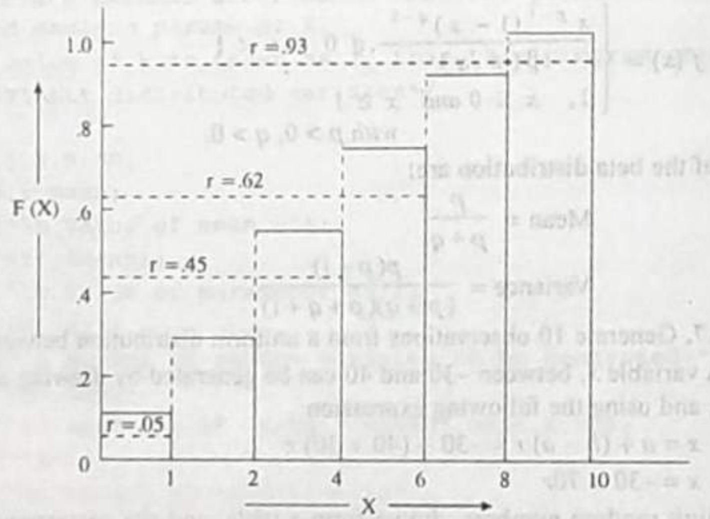


Fig. 5.14

Since CDF varies from 0 to 1, the y-axis also represents the random numbers from 0 to 1. Values of X, corresponding to random numbers 0.05, 0.45, 0.62 and 0.93, as obtained from the cumulative distribution are 1, 4, 6, and 10 respectively.

Example 5.9. Generate three random variates from a normal distribution with mean 20 and standard deviation 5. Take $n = 12$ for each observation. **[PTU B.Tech (Prod.) May 2006]**

Solution : A variate y, of a normal distribution is given by

$$y = \mu + \sigma\left(\sum_{i=1}^{n} r_i - \frac{n}{2}\right)$$

$$y = \mu + \sigma\left(\sum_{1}^{12} r_i - 6\right) \qquad \text{when } n = 12$$

where μ is mean and σ is standard deviation.

Since $n = 12$, we need 12 random numbers for generating one value of the variable.

(*i*) Twelve random numbers taken from a random number table are,

.483, .517, .063, .229, .807, .562, .066, .924, .511, .134, .657, .602

$$\sum_{i=1}^{12} r_i = 5.555$$

$$y_1 = 20 + 5\,(5.555 - 6) = 17.775$$

(*ii*) Twelve random numbers taken for the next value of y are,

.357, .944, .733, .345, .063, .546, .387, .935, .217, .816, .768, .183

$$\sum_{i=1}^{12} r_i = 6.294$$

$$y_2 = 20 + 5\,(6.294 - 6) = 21.47$$

(*iii*) From the same random number sequence, the next 12 random numbers are,

.488, .597, .183, .922, .731, .619, .232, .568, .421, .045, .897, .165

$$\sum_{i=1}^{12} r_i = 5.868$$

$$y_3 = 20 + 5\,(5.868 - 6) = 19.34$$

Thus three random variates from the given normal distribution are 17.775, 21.47 and 19.34.

Example 5.10: Generate three random variates from an exponential distribution having mean value 8.

Solution: A variate y, of an exponential distribution is given as,

$$y = -\frac{1}{\mu}\ \ln(1-r) \quad \text{or} \quad -\frac{1}{\mu}\ \ln(r)$$

where $\frac{1}{\mu}$ is the mean of the distribution

$$y = -\,8\ \ln(1-r)$$

Three random numbers taken from a table are

.513, .297 and .728

Then,

$$y_1 = -\,8\ \ln(1 - .513) = 5.756$$
$$y_2 = -\,8\ \ln(1 - .297) = 2.819$$
$$y_3 = -\,8\ \ln(1 - .728) = 10.416$$

The random variates could also be obtained as,

$$y_1 = -\,8\ \ln(.513) = 5.340$$
$$y_2 = -\,8\ \ln(.297) = 9.712$$
$$y_3 = -\,8\ \ln(.728) = 2.540$$

Example 5.11. Generate three random observations from an Erlang distribution having mean 8 and shape factors (*a*) 2, (*b*) 4.

Solution: A variate y of an Erlang distribution is given as

$$y = -\frac{1}{\mu k}\ \ln \prod_{i=1}^{k}(1-r_i)$$

or

$$-\frac{1}{\mu k}\ \ln \prod_{i=1}^{k} r_i$$

where $\frac{1}{\mu}$ is the mean and k is shape factor.

(*a*) Since $k = 2$, for each observation we need two random numbers. Six random numbers obtained from a table are,

.319, .461, .344, .548, .633, .822

$$y_1 = -\frac{8}{2} \ln \{.319 \times .461\} = -\frac{8}{2} \ln 0.14706 = 7.668$$

$$y_2 = -\frac{8}{2} \ln (.344 \times .548) = 6.674$$

$$y_3 = -\frac{8}{2} \ln (.633 \times .822) = 2.613$$

(*b*) Since $k = 4$, for each observation 4 random numbers are required. Twelve random numbers taken from the table are:

.320, .147, .638, .182, .527, .823, .627, .471, .220, .618, .292, .504

$$Y_1 = -\frac{8}{4} \ln (.320 \times .147 \times .638 \times .182) = 10.420$$

$$Y_2 = -\frac{8}{4} \ln (.527 \times .823 \times .627 \times .471) = 4.110$$

$$Y_3 = -\frac{8}{4} \ln (.220 \times .618 \times .292 \times .504) = 7.823$$

Example 5.12. Generate five random observations from the following distributions.

(*a*) Uniform distribution from 15 to 60

(*b*) Triangular distribution as

$$f(x) = \begin{cases} \dfrac{x^2}{2}, & \text{if } 0 \le x \le 1 \\ 0, & \text{otherwise} \end{cases}$$

(*c*) The distribution having the p.d.f. as

(*i*) $f(x) = \begin{cases} \dfrac{1}{50}(x-5), & \text{if } 10 \le x \le 20 \\ 0, & \text{otherwise} \end{cases}$ **[PTU, B.Tech May 2006]**

(*ii*) $f(x) = \dfrac{5}{7} x^2, \quad 1 \le x \le 2$

Solution: (*a*) If y is the random variable, then

$$y = f(x) = 15 + r \times (60 - 15)$$

where r is random number between 0 and 1.

Table 5.9

Random number	*Random observation* (y)
.526	38.67
.659	44.655
.136	21.12
.712	47.04
.348	30.66

(b) $y = f(x) = \begin{cases} \frac{x^2}{2}, & \text{if } 0 \le x \le 1 \\ 0, & \text{otherwise} \end{cases}$

Taking random number r between 0 and 1.

$$y = \frac{r^2}{2}$$

Random number	:	.181,	.963,	.652,	.892,	.137
Random observation	:	.016,	.464,	.213,	.398,	.0094

(c) (i) Again taking random numbers between 0 and 1,

$$y = f(x) = \frac{1}{50}(x-5)$$

where $x = 10 + (20 - 10)\,r$

Table 5.10

Random number	*x*	*y*
.386	13.86	0.1772
.501	15.01	0.2002
.967	19.67	0.2934
.764	17.64	0.2528
.447	14.47	0.1894
.293	12.93	0.1586

(ii) $f(x) = \frac{5}{7}x^2,\ 1 \le x \le 2$

Since x lies between 1 and 2, if we take a random number between 0 and 1, then $x = 1 + r$

$$y = f(x) = \frac{5(1+r)^2}{7}$$

Random number (r)	:	.163	.716	.279	.954	.934
Random observation (y)	:	.966	2.1033,	1.1685,	2.7935,	2.6717

Example 5.13. There are 10 equally reliable semiautomatic machines in a screw-manufacturing unit. Their breakdowns per day follow the **Binomial distribution** with probability of failure 0.15. Generate the number of breakdowns for the next seven days. Determine the mean and variance of the generated observations. What are the theoretical values of the mean and variance?

Solution: For generating one observation, that is the number of machines breaking down per day, 10 random numbers between 0 and 1 are drawn. The random number ≤ 0.15 gives a failure. The random numbers used here in Table 5.11 are drawn form the random number table given in Appendix.

Table 5.11

Day	*Random numbers*	*Breakdowns*
1	48, 51, **06**, 22, 80, 56, **06**, 92, 51, **13**	3
2	65, 60, 51, 50, **13**, 94, 57, 26, 78, 33	1
3	60, 31, **15**, 64, 89, 74, 39, 63, 58, 83	1
4	44, 64, 59, **03**, 59, 30, 16, 57, 87, 21	1
5	36, 60, 82, 37, 72, 33, 90, 76, 29, 66	0
6	40, **11**, 44, 74, 27, 16, 41, 20, 68, 95	1
7	28, 75, 16, **02**, 88, **13**, 74, **07**, 63, 56	3

$$\text{Mean } \bar{x} = \frac{1}{7}(3+1+1+1+0+1+3) = \frac{10}{7} = 1.4286$$

$$\text{Variance} = \frac{1}{6}[2(3-1.4286)^2 + 4(1-1.4286)^2 + (0-1.4286)^2]$$

$$\frac{1}{6}[4.9386 + .7348 + 2.0409] = \frac{7.7143}{6} = 1.2857$$

Theoretical mean $= np = 10 \times .15 = 1.5$

Theoretical variance $= np(1-p) = 10 \times .15 \times .85 = 1.275$

If the observations are carried out for a larger number of days, the observed mean and variance will approach the true mean and variance.

Example 5.14. A variable x has Negative Binomial distribution with parameters $k = 2$ and $p = 0.35$. Generate seven observations of the variable. Determine the observed mean and standard deviation. Also compute the true mean and standard deviation of the observations.

Solution: In Negative Binomial distribution, a random number is drawn to represent an event. If $r \le p$, it is success. The random numbers are drawn until number of successes becomes equal to k. The number of random numbers drawn represents x. The string of two-digit random numbers used below in Table 5.12 is taken from a random number table.

Table 5.12

S.No.	*Random number*	*Success*	*S.No.*	*Random number*	*Success*
1	39	0	1	24	1
2	58	0	2	05	2
3	55	0		$x_4 = 2$	
4	46	0	1	36	0
5	85	0	2	45	0
6	84	0	3	04	1
7	38	0	4	69	1
8	01	1	5	66	1
9	93	1	6	58	1
10	52	1	7	69	1
11	46	1	8	35	2
12	11	2		$x_5 = 8$	0
	$x_1 = 12$		1	29	1
1	57	0	2	29	2
2	75	0		$x_6 = 2$	
3	86	0	1	53	0
4	44	0	2	12	1
5	33	1	3	89	1
6	28	2	4	87	1
	$x_2 = 6$		5	67	1
1	93	0	6	30	2
2	58	0		$x_7 = 6$	
3	18	1			
4	91	1			
5	02	2			
	$x_3 = 5$				

Thus, values of x are 12, 6, 5, 2, 8, 2, 6

$$\text{Mean } \bar{x} = \frac{12+6+5+2+8+2+6}{7} = \frac{41}{7} = 5.857$$

Standard deviation,

$$\text{S.D.} = \left[\frac{1}{6}\sum(x_i - \bar{x})^2\right]^{\frac{1}{2}}$$

$$= \left[\frac{1}{6}\{(12-5.857)^2 + (6-5.875)^2 + (5-5.857)^{2+} + (2-5.857)^2\right.$$

$$= \left. +(8-5.857)^2 + (2-5.857)^2 + (6-5.857)^2\}\right]^{\frac{1}{2}}$$

$$= \left[\frac{1}{6}(37.736 + 0.1706 + 0.7344 + 6.5469 + 4.5924 + 14.8764 + 0.0205)\right]^{\frac{1}{2}}$$

$$= \left[\frac{64.6772}{6}\right]^{\frac{1}{2}} = 3.2832$$

$$\text{True mean} = \frac{k}{p} = \frac{2}{.35} = 5.7143$$

$$\text{True S.D.} = \frac{k\,(1-p)}{p^2} = \frac{2 \times .65}{(.35)^2} = 3.2576$$

Example 5.15. Generate the random variates, if the random variable of Example 5.14 follows geometric distribution with $p = 0.35$. Use the same string of random numbers. Determine the observed mean and S.D. as well as the true mean and S.D.

Solution:

Table 5.13

S.No.	*Random number*	*Success*	*S.No.*	*Random number*	*Success*
1	39	0	1	91	0
2	58	0	2	02	1
3	55	0		$x_6 = 2$	
4	46	0	1	24	1
5	85	0		$x_7 = 1$	
6	84	0	1	05	1
7	38	0		$x_8 = 1$	
8	01	1	1	36	0
	$x_1 = 8$		2	45	0
1	93	0	3	04	1
2	52	0		$x_9 = 3$	
3	46	0	1	69	0
4	11	1	2	66	0
	$x_2 = 4$		3	58	0
1	57	0	4	69	0
2	75	0	5	35	1
3	86	0		$x_{10} = 5$	
4	44	0	1	29	1

5	33	1		$x_{11} = 1$	
	$x_3 = 5$		1	28	1
1	28	1		$x_{12} = 1$	
	$x_4 = 1$		1	53	0
1	93	0	2	12	1
2	58	0		$x_{13} = 2$	
3	18	1	1	89	0
	$x_5 = 3$		2	87	0
			3	67	0
			4	30	1
				$x_{14} = 4$	

Thus random values of x are 8, 4, 5, 1, 3, 2, 1, 1, 3, 5, 1, 1, 2, 4

$$\text{Observed mean } \bar{x} = \frac{1}{n}\sum x_i = \frac{41}{14} = 2.9286$$

$$\text{Standard deviation S.D.} = \frac{1}{n-1}\sum (x_i - \bar{x})^2$$

$$\sum_i (x_i - \bar{x})^2 = 25.719 + 1.148 + 4.291 + 3.719 + .0051 + .8623 + 3.7195 + 3.7195 + .0051 + 4.291 + 3.7195 + 3.7195 + .8623 + 1.1479 = 57.9268$$

$$\text{S.D.} = \sqrt{\frac{57.9268}{13}} = 2.1108$$

$$\text{True mean} = \frac{1}{p} = \frac{1}{.35} = 2.857$$

$$\text{True S.D.} = \sqrt{\frac{1-p}{p^2}} = \sqrt{\frac{1-.35}{.35 \times .35}} = \sqrt{5.306} = 2.304$$

Example 5.16. Generate an Exponential distribution with mean value equal to 3.

Solution : An exponential variable x is given by

$$x = -\text{Mean} \times \ln(r)$$

where r is a random number between 0 and 1. The probability density function can be drawn by generating a large number of random observations. Larger the number of observations, more accurate will be the distribution. Fifty values of x have been generated in Table 5.14. The random numbers used have been taken from the table given in Appendix.

Based on these 50 values of x, a histogram showing the frequency distribution is plotted in Fig. 5.15. If the number of observations is increased, and the interval is further shortened, the histogram will result into a smooth curve like the one shown in Fig. 5.15.

The observed mean of the distribution,

$$\bar{x} = \frac{1}{\sum f}\sum f \cdot x = \frac{1}{50}\,[12 \times 0.5 + 11 \times 1.5 + 6 \times 2.5 + 5 \times 3.5 + 5 \times 4.5 + 3 \times 5.5 + 5 \times 6.5 + 1 \times 7.5 + 2 \times 8.5 + 0 \times 9.5]$$

$$= \frac{1}{50}\,[151] = 3.02$$

Obtained standard deviation is

$$= \left[\frac{1}{49}\ [12\,(3.02-0.5)^2 + 11\,(3.02-1.5)^2 + 6\,(3.02-2.5)^2 + 5\,(3.02-3.5)^2 + 5\,(3.02-4.5)^2 + 3\,(3.02-5.5)^2 + 5\,(3.02-6.5)^2 + (3.02-7.5)^2 + 2\,(3.02-8.5)^2]\right]^{\frac{1}{2}}$$

$$= \sqrt{\frac{274.48}{49}} = 2.367$$

Table 15.14

Rand	*x*	*Rand*	*x*	*Rand*	*x*
2182	4.57	7615	0.82	1955	4.90
1128	6.55	8508	0.48	2177	4.66
7112	1.02	6970	1.08	7471	0.87
6557	1.27	5799	1.63	8674	0.43
4199	2.60	6364	1.36	1092	6.64
3544	3.11	4165	2.63	9061	0.30
1749	5.23	8354	0.54	6438	1.32
1903	0.28	9130	0.27	1834	5.09
0764	7.72	5826	1.62	1884	5.01
3492	3.16	6285	1.39	6791	1.16
1292	6.14	7527	0.85	2068	4.73
4397	2.46	8976	0.32	7295	0.95
3807	2.90	2327	4.37	3440	3.20
4984	2.09	1182	6.41	5435	1.83
1340	6.03	3659	3.02	3090	3.52
0590	8.49	5924	1.57	0607	8.41
9566	0.13	3941	2.79	—	—

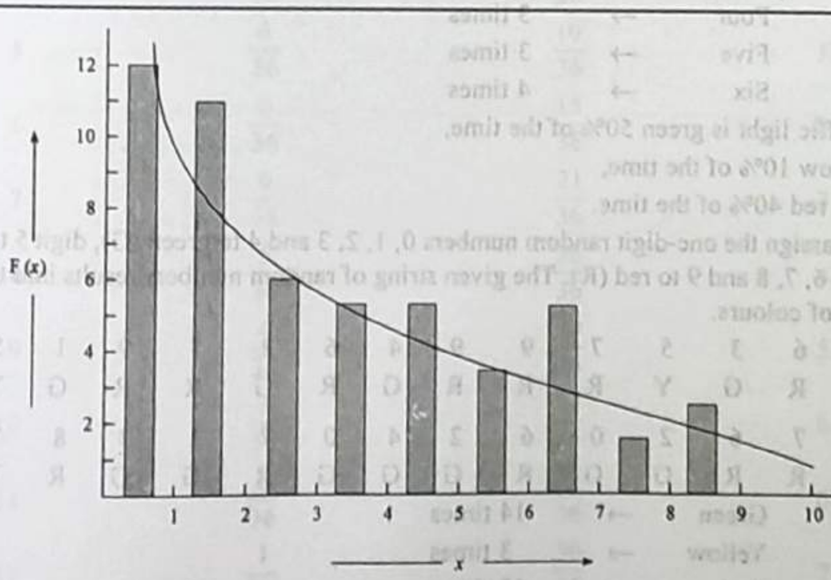


Fig. 5.15

Example 5.17. Use the one-digit random numbers 6, 3, 5, 7, 9, 9, 4, 6, 2, 7, 9, 1, 5, 8, 0, 7, 6, 2, 0, 6, 2, 4, 0, 7, 1, 3, 8, 5, 2, 4 ; to generate random observations for each of the following situations.

(*a*) Throwing an unbiased coin

(*b*) Throwing a dice

(*c*) The colour of a traffic light found by a randomly arriving car, when green is 50% of the time, yellow is 10% of the time, and red is 40% of the time. **[PTU. B.Tech Dec. 2005]**

Solution: (*a*) **Throwing an unbiased coin:** When an unbiased coin is tossed, the chances of getting the head (H) and tail (T) are equal that is 50 : 50. Using the one-digit random numbers between 0 and 9, let the first five digits 0, 1, 2, 3 and 4 stand for H and the remaining five digits 5, 6, 7, 8 and 9 stand for T. For the given string of random numbers, the observations are given below in Table 5.15.

Table 5.15

Rand. No. :	6	3	5	7	9	9	4	6	2	7	9	1	5	8	0
H or T :	T	H	T	T	T	T	H	T	H	T	T	H	T	T	H
Rand. No. :	7	6	2	0	6	2	4	0	7	1	3	8	5	2	4
H or T :	T	T	H	H	T	H	H	H	T	H	H	T	T	H	H

Number of heads = 14

Number of tails = 16

Thus out of 30 throws head comes 14 times, while tail comes 16 times.

(*b*) **Throwing a Dice:** A dice has six faces, and each face has same chance of coming up. Out of the given one-digit random numbers, let us use the numbers 1, 2, 3, 4, 5 and 6 to represent the six faces of the dice, and reject the remaining. Using the given string of random numbers the first throw results into six, while the second gives 3 and third through gives 5 etc.

We have;

One	→	2 times
Two	→	4 times
Three	→	2 times
Four	→	3 times
Five	→	3 times
Six	→	4 times

(*c*) Traffic light is green 50% of the time,
yellow 10% of the time,
and red 40% of the time.

We can assign the one-digit random numbers 0, 1, 2, 3 and 4 to green (G), digit 5 to yellow (Y) and the digits 6, 7, 8 and 9 to red (R). The given string of random numbers results into the following observations of colours.

Rand. No. :	6	3	5	7	9	9	4	6	2	7	9	1	5	8	0
Colour :	R	G	Y	R	R	R	G	R	G	R	R	G	Y	R	G
Rand. No. :	7	6	2	0	6	2	4	0	7	1	3	8	5	2	4
Colour :	R	R	G	G	R	G	G	G	R	G	G	R	Y	G	G

Green	→	14 times
Yellow	→	3 times
Red	→	13 times

Example 5.18. The game of craps requires the players to throw two dice one or more times until a decision has been reached. The player wins the game if the first throw results in a sum of 7 or 11 or if the first sum is 4, 5, 6, 8, 9 or 10 and the same repeats before a sum of 7 has appeared.

(*a*) Simulate 15 plays of the game, in the form of a table.

(*b*) Develop a simulation program in any computer language and run the simulation for 100, 200 and 300 games. **[PTU. B.E. 4th Sem. Elect. Dec. 2005]**

Solution: There are two dice, which are thrown at a time, and a throw can result into a sum of 2, 3, 4, 5, 6, 7, 8, 9, 10, 11 or 12. Sum 2 can appear only in one way *i.e.*, 1 + 1, the probability of the same is $\frac{1}{6}\times\frac{1}{6}=\frac{1}{36}$. Sum 3 can occur in two ways 1 + 2 and 2 + 1, and thus the probability of sum 3 is $2\left(\frac{1}{6}\times\frac{1}{6}\right)=\frac{2}{36}$. The ways in which a sum can occur and the corresponding probabilities were computed in Example 5.1, and the same are given below:

Sum (s) :	2	3	4	5	6	7	8	9	10	11	12
p (s) :	$\frac{1}{36}$	$\frac{2}{36}$	$\frac{3}{36}$	$\frac{4}{36}$	$\frac{5}{36}$	$\frac{6}{36}$	$\frac{5}{36}$	$\frac{4}{36}$	$\frac{3}{36}$	$\frac{2}{36}$	$\frac{1}{36}$

For simulating the game, let us take the random numbers from 00 to 72, and assign these to different outcomes of the throws that is sums, as in Table 5.16.

Table 5.16

Sum	*Probability*	*Cumulative probability*	*Random numbers*
2	$\frac{1}{36}$	$\frac{1}{36}$	01 to 02
3	$\frac{2}{36}$	$\frac{3}{36}$	03 to 06
4	$\frac{3}{36}$	$\frac{6}{36}$	07 to 12
5	$\frac{4}{36}$	$\frac{10}{36}$	13 to 20
6	$\frac{5}{36}$	$\frac{15}{36}$	21 to 30
7	$\frac{6}{36}$	$\frac{21}{36}$	31 to 42
8	$\frac{5}{36}$	$\frac{26}{36}$	43 to 52
9	$\frac{4}{36}$	$\frac{30}{36}$	53 to 60
10	$\frac{3}{36}$	$\frac{33}{36}$	61 to 66
11	$\frac{2}{36}$	$\frac{36}{36}$	67 to 70
12	$\frac{1}{36}$	$\frac{36}{36}$	71 to 72

The random numbers may be drawn from a random number table, or generated by some random number generation technique. The random numbers used in this simulation have been generated by a scientific calculator. In each number, the first two digits were retained. Taking the following string of random numbers 12, 62, 60, 33, 04, 33, 44, 64, 50, 72, 08, 60, 51, ..., etc. simulation is done as follows.

First game: The first throw of two dice, simulated by first random number 12, gives the sum as 4. It is neither 7 nor 11, hence take the 2nd throw, a random number 62 gives sum of 10. Third throw, random number 60, gives sum of 9, while 4th throw, random number 33, gives sum of 7. Since first sum '4' does not reappear before the occurrence of sum 7, player loses the first game.

Second game: Next random number is 04, which gives a sum of 3. Since, it is neither 7 or 11, nor any of 4, 5, 6, 8, 9 and 10, the player loses the second game too.

Third game: Next random number 33 results in a sum of 7. Player wins the game in the very first throw.

The simulation of the process of 15 games is given in Table 5.17. Seven times the player wins and eight times loses the game. Based on this small sample, the probability of a win is thus $\frac{7}{15} \times 100 = 46.6\%$.

However, such a small length of simulation run, cannot give very reliable results.

Table 5.17

Game No.	*Random numbers*	*Sum*	*Result*
1	12	4	
	62	10	
	60	9	
	33	7	Loses
2	04	3	Loses
3	33	7	Wins
4	44	8	
	64	10	
	50	8	Wins
5	72	12	Loses
6	08	4	
	60	9	
	51	8	
	01	2	
	17	5	
	11	4	Wins
7	30	6	
	44	8	
	71	12	
	04	3	
	20	5	
	34	7	Loses

8	46	8	
	18	5	
	42	7	Loses
9	56	9	
	12	4	
	02	2	
	44	8	
	48	8	
	36	7	Loses
10	59	9	
	24	6	
	65	10	
	44	8	
	57	9	Wins
11	05	3	Loses
12	37	7	Wins
13	49	8	
	66	10	
	06	3	
	54	9	
	07	4	
	43	8	Wins
14	11	4	
	55	9	
	51	8	
	16	5	
	39	7	Loses
15	67	11	Wins

Number of games won = 7

Number of games lost = 8

A computer program for simulating the game of craps, developed in C language is given below.

```
#include<stdio.h>
#include<stdlib.h>
#include<ctype.h>
int throw(void);
main()
   {  /* Simulation of Game of Craps*/
      /* n is the number of games */

      float x;
      int i,k,n,n1,n2,score1,score2,win=0,loss=0;
      printf("\n The number of games to play n=");
```

```
scanf("%d",&n);
for(i=0;i<=n;++i) {
score1=throw();
printf("\n i=%d  score=%d",i,score1);
if(score1==7 || score1==11) { win=win+1;
      printf("     It is a Win"); }
 else if(score1==2 || score1==3 || score1==12) { loss+=1;
      printf("     It is a loss"); }
 else {
      do  {
      score2=throw();
      printf(" score2=%d",score2);
        } while(score2 != score1 && score2 != 7);
      if(score2== score1) { win+=1;
      printf("     Win by matching first score");}
      else  { loss+=1;
      printf("     Lose due to failingto match first score");}
    }
}
printf("\n i=%d Wins=%d Losses=%d",i,win,loss);
/* printf("\n Any digit");
 scanf("%d",k); */
}
/* Function throw to generate the sum of two throws of dice*/
int throw (void) {
int n1,n2;
float x;
x=rand()/32768.;
n1=1+(int)(6*x);
x=rand()/32768.;
n2=1+(int)(6*x);
return(n1+n2);   }
```

Some results obtained from this program are as under.

Games	*Wins*	*Loses*
50	25	25
100	51	49
200	96	104
300	148	152

Example 5.19. The distribution of inter arrival times in a single server model is,

T :	1	2	3
$f(t)$:	$\frac{1}{4}$	$\frac{1}{2}$	$\frac{1}{4}$

and the distribution of service times is

S :	1	2	3
$f(s)$:	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{4}$

Complete the following table, using the two-digit random numbers 12, 40, 48, 93, 61, 17, 55, 21, 85, 88 to generate arrivals and 54, 90, 18, 38, 16, 87, 91, 41, 54, 11 to generate the corresponding service times.

Arrival number	*Arrival time*	*Time service begins*	*Time service ends*	*Waiting time in Queue*
1				
2				
3				
4				
etc.				

Solution : It is a simple case of a single server queuing simulation. Both the inter arrival and service times of the customers are random and follow the given discrete distributions.

The distribution of inter arrival times is

T	$f(t)$	$F(t)$	*Random numbers*
1	.25	.25	00 to 24
2	.50	.75	25 to 74
3	.25	1.00	75 to 99

Inter arrival times corresponding to the given random numbers are,

Random number	:	12	40	48	93	61	17	55	21	85	88
Inter arrival time	:	1	2	2	3	2	1	2	1	3	3

The distribution of service times is,

S	$f(s)$	$F(s)$	*Random numbers*
1	.50	.50	00 to 49
2	.25	.75	50 to 74
3	.25	1.00	75 to 99

Service times corresponding to given random numbers are;

Random number	:	54	90	18	38	16	87	91	41	54	11
Service time	:	2	3	1	1	1	3	3	1	2	1

Now the required table can be completed (Table 5.18). As soon as an arrival takes place, it will go into service, if the facility is idle that is the service on the previous arrival has been completed, otherwise it will wait till the facility becomes available.

Table 5.18

Arrival number	*Arrival time* $AT + T = AT$	*Time service begins SB*	*Time service ends* $SB + S = SE$	*Waiting time in Queue*
1	0 + 1 = 1	1	1 + 2 = 3	0
2	1 + 2 = 3	3	3 + 3 = 6	0
3	3 + 2 = 5	6	6 + 1 = 7	1
4	5 + 3 = 8	8	8 + 1 = 9	0
5	8 + 2 = 10	10	10 + 1 = 11	0
6	10 + 1 = 11	11	11 + 3 = 14	0
7	11 + 2 = 13	14	14 + 3 = 17	1
8	13 + 1 = 14	17	17 + 1 = 18	3
9	14 + 3 = 17	18	18 + 2 = 20	1
10	17 + 3 = 20	20	20 + 1 = 21	0

5.20 Central Limit Theorem

The central limit theorem is the most important theorem in statistical inferences. It sates that the average of sample of observations drawn from some population with any shape distribution approaches the normal distribution as the sample size increases. There are theoretical situations, where the central limit theorem fails, but these are rarely encountered in practice. In theoretical statistics there are many versions of the central limit theorem. These versions mainly depend upon the conditions under which the theorem is used and upon the assumptions about the distribution of parent population and the sampling techniques employed.

The most commonly used version of the central limit theorem states that, if there is a random sample of size n, n being larger than 30, drawn from an infinite population having a unit standard deviation, the standardized sample mean converges to a standard normal distribution or equivalently the sample mean approaches a normal distribution with mean equal to the population mean and standard deviation equal to standard deviation of population divided by the square root of sample size n.

The significance of the central limit theorem is that it enables us to use the sample statistics to draw inferences about the population parameters, without knowing anything about the probability distribution of the population.

Example 5.20: Fig. 5.16 (*a*) shows the distribution of annual commission earnings of all salesmen in a large corporation. It has a mean of Rs. 12500 and a standard deviation of Rs. 1500. The distribution is skewed to the right. A random sample of 30 salesmen has been drawn from population. What is the probability that their average earnings will be more than Rs. 13100?

According to the central limit theorem, the sampling distribution of the mean approaches normal distribution irrespective of the shape of the parent population distribution. The sampling distribution is shown on Fig. 5.16 (*b*).

Further, the central limit theorem states, that the sampling distribution has a standard deviation, which we also call a standard error, equal to the population standard deviation divided by the square root of sample size.

$$\sigma_{\bar{x}} = \frac{\sigma}{\sqrt{n}}$$

where $\sigma_{\bar{x}}$ is the standard error of the mean and σ is the standard deviation of the population, and n is sample size.

$$\sigma_{\bar{x}} = \frac{1500}{\sqrt{30}} = 273.87$$

Now, we determine the z statistics, the standard deviation from the mean of a standard normal probability distribution

$$z = \frac{\bar{x} - \mu}{\sigma_{\bar{x}}}$$

where, μ is the mean of the population distribution. μ is also the mean of the sample distribution. Z gives the distance of the given point from the mean.

$$z = \frac{13100 - 12500}{273.87} = 2.2$$

The area under the curve between the mean and a point 2.2 standard deviations away to the right of the mean is 0.4861 (Appendix Table A-7) the area beyond the point 2.2 is $0.5 - .4861 = 0.0139$, the shaded area on Fig. 5.16 (*b*).

Thus, we can say that there are hardly 1.4% chances that the average annual income from commission earnings of all the salesman exceeds Rs. 13100.

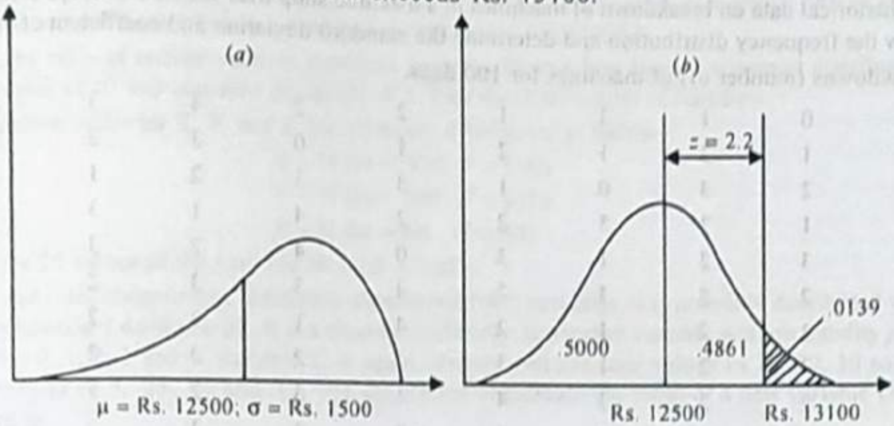


Fig. 5.16

Example 5.21: A mainframe computer includes a backup circuit to minimize the downtimes. After running the simulation of the working of a particular component, it has been found that its average service life is 4300 hours, with a standard deviation of 730 hours. The backup circuit contains two duplicate components so that in case of malfunction of one, the other component automatically gets switched on.

(*a*) What is the probability that the set of components will last 14000 hours ?

(*b*) At the most 12000 hours ?

Solution: The three identical components combined will have a life of $4300 \times 3 = 12900$ hours with the standard deviation of 730 hours.

$$\mu = 12900, \quad \sigma = 730$$

When $\bar{x} = 14000$

$$Z = (14000 - 12900)/730 = 1100/730 = 1.5.$$

The area of the tail beyond 14000 = 0.4332 from Appendix Table A-7.

Thus the possibility of life of components being more than 14000 $= .5000 - .4332$

$$= .0668 = 6.7\%$$

When $\bar{x} = 12000$,

$$Z = \frac{12000 - 12900}{730} = -1.23.$$

From Appendix Table A-7, the area under the normal curve between -1.23 and mean, μ, is 0.39. Thus the area under the curve up to the right tail is 0.89 as shown in Fig. 5.17. The probability of the life of components being 12000 hours or more is thus 89%.

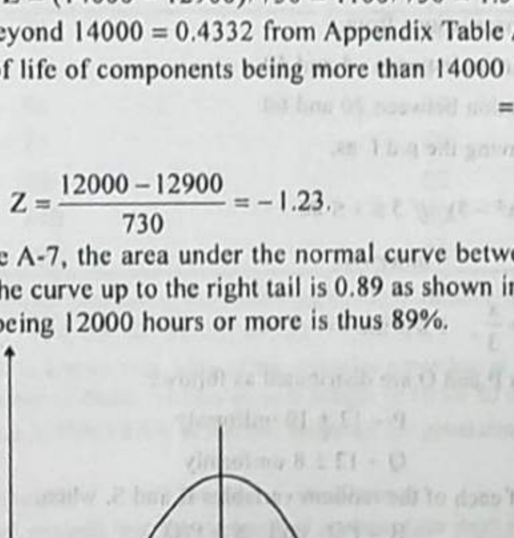


Fig. 5.17

5.21 Exercises

1. The historical data on breakdown of machines in a machine shop over the last 100 days is given below. Draw the frequency distribution and determine the standard deviation and coefficient of variation.
   Breakdowns (number of) of machines for 100 days.

2	0	1	3	1	2	2	4	3	2
3	1	2	1	2	1	0	3	2	5
4	2	3	0	1	3	3	2	1	2
0	1	3	5	2	2	4	1	3	4
1	3	2	1	3	0	4	2	3	2
4	2	5	3	2	4	3	3	2	1
2	3	2	1	2	4	1	3	2	5
0	3	2	1	3	4	2	3	2	1
1	2	2	3	4	1	3	2	5	0
3	4	0	1	3	2	3	3	2	1

2. Draw a frequency distribution from the following data, grouping the data in steps of 200. Find the mean and standard deviation of the data. Also determine the mode and median of the distribution.

196	565	3229	1575	288
411	224	420	1720	723
1128	945	519	508	36
852	380	321	3158	598
2056	1180	417	479	1350
1106	99	717	1530	518
417	599	662	2961	93
544	245	420	16	275
1268	133	715	149	3071
174	870	1500	1968	1217

3. Generate 5 random observations from,
   (*a*) a Uniform distribution between –5 and 40
   (*b*) a Uniform distribution between 20 and 60
   (*c*) the distribution having the p.d.f. as,

$$(i)\ f(x) = \begin{cases} \dfrac{1}{20}(x^2 - 5) & \text{if } 5 \le x \le 20 \\ 0, & \textit{otherwise} \end{cases}$$

$$(ii)\ f(x) = \frac{7}{8}x^2 + \frac{x}{3}, \quad 1 \le x \le 2$$

4. The random variables P and Q are distributed as follows:

$$P \sim 12 \pm 10 \text{ uniformly}$$
$$Q \sim 12 \pm 8 \text{ uniformly}$$

   Simulate 50 values of each of the random variables R and S, when,

$$R = PQ \quad \text{and} \quad S = P/Q$$

5. Generate five random observations from each of the following distributions.
   (*a*) Exponential with mean = 5
   (*b*) Erlang with mean = 5 and $k = 3$
   (*c*) Normal with mean = 5 and standard deviation = 4.

6. Generate five random observations from a Poisson distribution, if the mean is 4.0.
7. Determine the probability of their being n arrivals ($n = 0, 1, \ldots, 10$) in an inter arrival time of 10 seconds, when the arrivals have a Poisson distribution with mean value of 0.4.
8. Using the table of normal random numbers generate 50 numbers having a normal distribution, with mean value of 10 and standard deviation of 3. Plot the distribution of numbers.
9. The random variables X, Y, and Z are normally distributed as follows:

$$X \sim N\,(m = 100,\ s^2 = 100)$$
$$Y \sim N\,(m = 300,\ s^2 = 225)$$
$$Z \sim N\,(m = 40,\ s^2 = 64)$$

Simulate 25 values of the variable $W = (X + Y)/Z$

10. A, B, and C are independent identically distributed (IID) variables. A is normally distributed with mean 100 and standard deviation 20. B is a discrete uniformly distributed variable with probability $p(b) = 0.2$ with $b = 0, 1, 2, 3$ and 4. Variable C is again discrete and can take values as 10, 20, 30 and 40 with probabilities of .5, .25, .45 and .15. Use simulation to estimate the mean of a new variable D, which is defined as

$$D = (A - 25\,B) / (2C).$$

Use a sample of size 20.

11. The number of customers, who arrive at a repair shop can be described by a Poisson distribution that has a mean of 4 per hour. Generate arrivals for the first 100 hours. Plot the frequency distribution.
12. The lifetime, in years, of a satellite placed in orbit is given by the following pdf

$$f(x) = \begin{cases} 0.4e^{-0.4x}, & x \geq 0 \\ 0, & otherwise \end{cases}$$

(*a*) What is the probability of the life of satellite being more than 5 years?

(*b*) What is the probability of the life of the satellite being between 3 and 6 years?

13. Data have been collected on service times at a drive in bank window at the Ludhiana Bank. This data are summarized into intervals as follows:

Interval (*Seconds*)	*Frequency* y
10 – 20	10
20 – 35	20
35 – 50	30
50 – 75	40
75 – 100	50
100 – 140	20
140 – 200	10

Generate 10 values of service time using the following random number string.

74, 50, 03, 31, 21, 09, 23, 50, 97, 39.

14. From the past records, it is known that 30% of the vehicles reporting at a filling station have diesel engines. Generate the number of diesel vehicles in each sample of 10 for 50 samples. Plot the distribution.
15. Write a computer program in FORTRAN or BASIC language for generating and plotting the following probability distribution :

(*i*) Normal (*ii*) Exponential (*iii*) Erlang (*iv*) Poisson.

16. It is known from the past records that the demand of an item (per day) takes place according to the following distribution:

Demand	0	2	3	4	5
Probability	$\frac{1}{10}$	$\frac{1}{4}$	$\frac{1}{2}$	$\frac{1}{10}$	$\frac{1}{20}$

Simulate the demand for 15 days. **[PTU, B.Tech. (Prod.) May 2006]**

17. Collect the data on the inter arrival times and service times at a petrol pump. Approximate a probability density function for the inter arrival times and service times. How do the pdf vary over different time intervals of the day, like 7.00 AM to 10.00 AM, from 12.00 noon to 3.00 PM and from 5.00 PM to 8.00 PM.
18. Go to a major traffic intersection and determine the inter arrival times distribution, for the vehicles from each direction. Some arrivals want to go straight, some turn left, some turn right and some may take a U-turn. Based on this data design traffic lights system and develop its simulation model.
19. The electrical resistance of a copper wire increases with the increase in temperature. The experimental observations are given below :

Resistance (R)	19.10	25.00	30.10	36.00	40.00	45.10	50.00
Temperature (T)	76.30	77.80	79.75	80.80	82.35	89.90	85.10

By plotting a graph it can be seen that the relationship between R and T is almost linear. Develop the regression equation. **[Ans. $R = 71.14 + 0.2797$]**
20. Generate 10 random observations from the following probability density function assuming a set of 10 random numbers.

$$f(x) = \begin{cases} 2x, & \text{if } 0 \le x \le \\ 0, & \textit{otherwise} \end{cases}$$ **[P.U. ME (Mech.) 1988]**

21. Use the one-digit random numbers 2, 7, 9, 5, 8, 0, 7, 6, 2, 5, 6, 3, 5, 7, 8 to generate random observations for the colour of a traffic light found by a randomly arriving car, when green is 60% of the time, yellow is 10% and red 30% of the time. **[PTU B.Tech. (Prod.) Dec 2006]**

CHAPTER 6

SIMULATION OF QUEUING SYSTEMS

6.1 Introduction

Waiting line queues are one of the most important areas, where the technique of simulation has been extensively employed. The waiting lines or queues are a common site in real life. People at railway ticket window, vehicles at a petrol pump or at a traffic signal, workers at a tool crib, products at a machining center, television sets at a repair shop and consumers at a ration depot are a few examples of waiting lines. The waiting line situations arise, either because,

- There is too much demand on the service facility so that the customers or entities have to wait for getting service, or
- There is too less demand, in which case the service facility have to wait for the entities.

Thus, when facilities are inadequate, the entities wait and incur cost due to waiting time, and when the demand is inadequate, the facilities wait and incur cost due to idle time. The objective in the analysis of queuing situations is to balance the waiting time and idle time, so as to keep the total cost at minimum.

The queuing theory owe its development to an engineer named A.K. Erlang, who in 1920, studied waiting line queues of telephone calls in Copenhagen, Denmark. The problem was that during the busy period, telephone operators were unable to handle the calls, there was too much waiting time, which resulted in consumer dissatisfaction.

Many researchers carried on the research work in the telephone traffic further and it was only after World War II, that the queuing theory encompassed the waiting line situations from other fields as business and industry.

6.2 The Components of a Waiting Line System

(*i*) **Calling Source** or the population from which customers are drawn. The calling source may be finite or infinite. When the arrival of a customer does not affect the next arrival, the source of customers is said to be infinite. Population of vehicles arriving at a toll booth or patients arriving at a hospital have infinite calling source. In case of a repair crew looking after a group of 5 machines, if one breaks down and put under repair, then the next breakdown has to come from only 4 machines, the probability of which will be more than the previous. The population in this case is finite.

(*ii*) **Waiting Line** or queue, *i.e.*, the number of customers waiting to be served. Depending upon the space available, it may again be of finite or infinite length. In a barbershop having four chairs, the queue can have a maximum length of four, because people seldom queue up outside a barber shop. An important case of finite queue is the buffer used to decouple two sequential workstations, where the output of first station is not perfectly matched to the second workstation.

(*iii*) **Service Facility** or the number of service channels. The simplest case is of one service channel, where all the customers form one queue and are attended by one server, a single server model. In many cases, waiting line systems have more than one service facility.

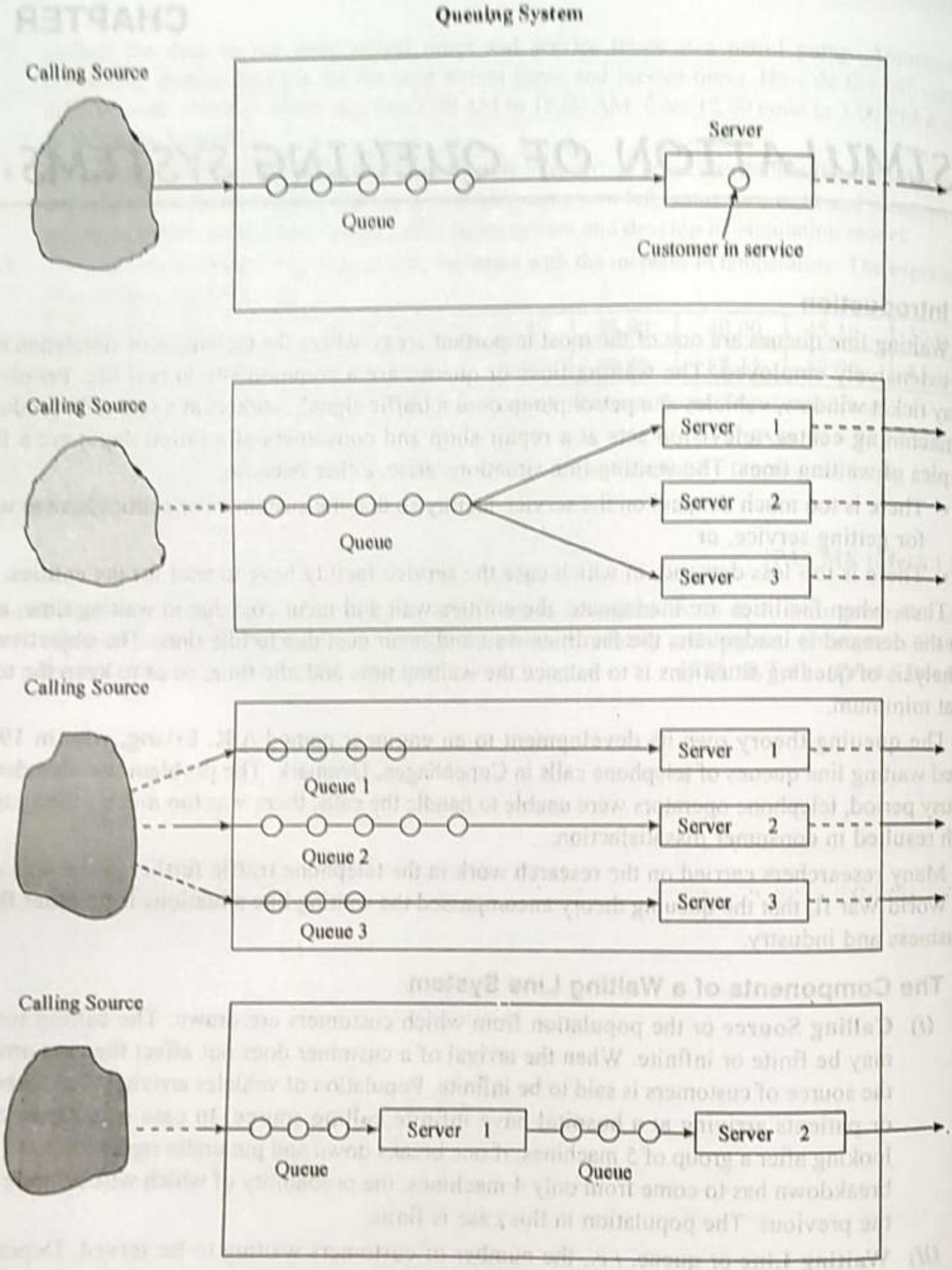


Fig. 6.1

These facilities may be working in parallel or in series. When in parallel, there may be a single queue or multiple queues. Some typical queuing systems are shown in Fig. 6.1.

The important ***attributes*** that determine the properties of a waiting line queue are,

- Input or arrival rate.
- Output or service rate.
- Service or queue discipline.

The arrival rate is the average number of entities, which join the waiting line per unit time. Depending upon the system, the time unit may be a second, a minute, an hour or a day etc. The average time between consecutive arrivals is called **inter-arrival time.** In most of the situations, the arrivals are a random phenomenon. Different probability density functions are applicable to different arrival patterns, but a commonly made assumption is that arrival rates follow the Poisson distribution.

The service rate or the departure rate is the average number of customers served per unit time. Average service time is the reciprocal of the service rate. Depending upon the situation, the service rate may be constant or variable and may follow any distribution as Uniform, Normal, Exponential, and Erlang etc. But in most of the situations service time is assumed to follow exponential distribution.

The third factor for describing the waiting line is the **queue discipline** that determines how the customers are selected from the queue for service. The most common queue disciplines are given below.

(*i*) **First-in, First-out (FIFO):** Also called the first come first served, is the most common service discipline, according to which the customers are served in the order of their arrival.

(*ii*) **Last-in, First-out (LIFO):** In some situations, the last arrival is served first, as in big go-downs the items coming last are taken out first, in crowded trains or elevators, passengers getting in last come out first.

(*iii*) **Priority:** An arrival may be given priority over the customers waiting in line. A particular machine in a production shop may be more important than the others, and when it breaks down, its repair may be taken up on priority as compared to the other broken down machines. When an arrival, not only goes to the head of the queue, but displaces any unit already in service, it is said to have **pre-emptive priority**. The new arrival is said to pre-empt or interrupt the service.

(*iv*) **Random:** The service discipline is said to be random, when all waiting customers have equal chance of getting selected for service. The selection follows purely random choice.

Some other terms commonly associated with waiting lines are given below.

- **Reneging:** When a queue grows excessively long, a customer waiting in the queue may become impatient and may leave the queue before it is due to enter the service facility. This process is called reneging.
- **Balking:** When a queue grows very long and an arrival refuses to join the queue, it is called balking.
- **Jockeying:** In multiple queues before multiple service channels, where all the channels are providing the same service, a customer may leave one queue and join the other looking faster. This process is called jockeying.
- **Polling:** When there are more than one queues forming for the same service, the action of sharing service between the queues is called polling. A bus picking up passengers from different stoppages along its route is an example of polling service. Separate queues for ladies and gents at a ticket window, is another example of polling service.

6.3 Stationary and Time Dependent Queues

The statistical properties of the arrival and service are generally assumed to be constant and independent of time. Such systems are called stationary systems. However in some situations, both the arrival and service rates may vary with time. For example, in the early morning the number of vehicles reporting at a toll office may be less, and the arrival rate may increase as the day advances towards noon. Similarly, the service rate may be higher in the morning and may come down near lunchtime. Such systems are said to be non-stationary or time dependent or time variant systems.

6.4 Transient and Steady States of the System

A system is said to be in transient state when its operating characteristics vary with time. If we start with an empty queuing system, the customer arriving first does not have to wait at all. The waiting time of customers goes on increasing as the queue builds up with the passage of time. After some time the system reaches steady state, where its behavior becomes almost steady. It is the steady state condition of the system, which is more important. For a system to reach steady state, the arrival rate should be less than the service rate. If the arrival rate is greater than the service rate, the queue length goes on increasing with time and the system never reaches the steady state. Such a state of the system is called **explosive.** In such a system, a maximum limit on the length of the queue should be imposed to ensure steady state.

6.5 Measures of System Performance

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

(*i*) Average number of customers in the queue or in the system.

(*ii*) Average waiting time of the customers in the queue or in the system.

(*iii*) System utilization.

(*iv*) The cost of waiting time and idle time.

Each of these measures has its own importance. The knowledge of average number of customers in the queue or in the system helps to determine the space requirements of the waiting entities. Also too long a waiting line may discourage the prospectus customers, while no queue may suggest that service offered is not of good quality to attract customers.

The knowledge of average waiting time in the queue is necessary for determining the cost of waiting in the queue.

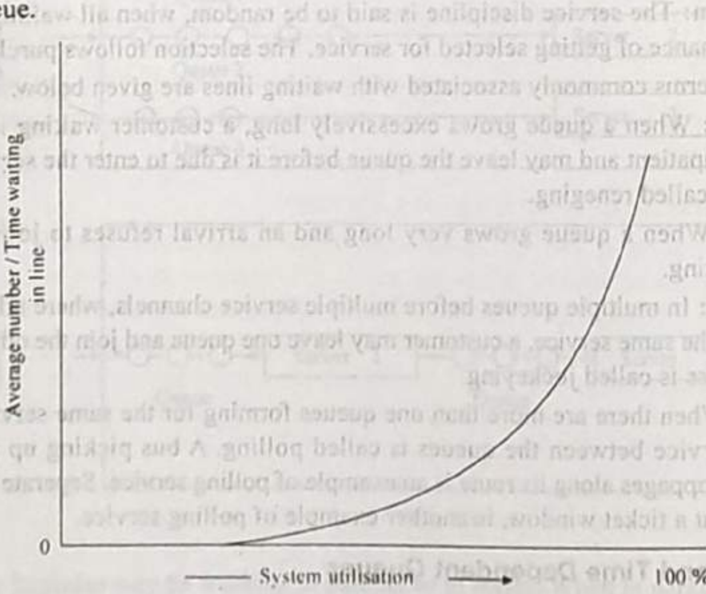


Fig. 6.2

System utilization, that is, the percentage capacity utilized reflects the extent to which the facility is busy rather than idle. System utilization factor (S) is the ratio of average arrival rate (λ) to the average service rate (μ).

$S = \lambda / \mu$ in case of a single server model.

$= \lambda / \mu n$ in 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. 6.2. Under normal circumstances 100% system utilization is not a realistic goal.

6.6 Costs of Customer Waiting Time and Idle Capacity

The customers in the waiting line who may be human being or physical objects, incur cost while waiting. It may be workers waiting at a tool crib or the trucks waiting for loading or unloading, they incur cost. The capacity cost relates to maintaining the ability to provide service. When a service facility is idle, the capacity is wasted. Both the costs are functions of system capacity, as illustrated in Fig. 6.3. The objective of analysis of any queuing system is to determine the capacity, which minimizes the total cost.

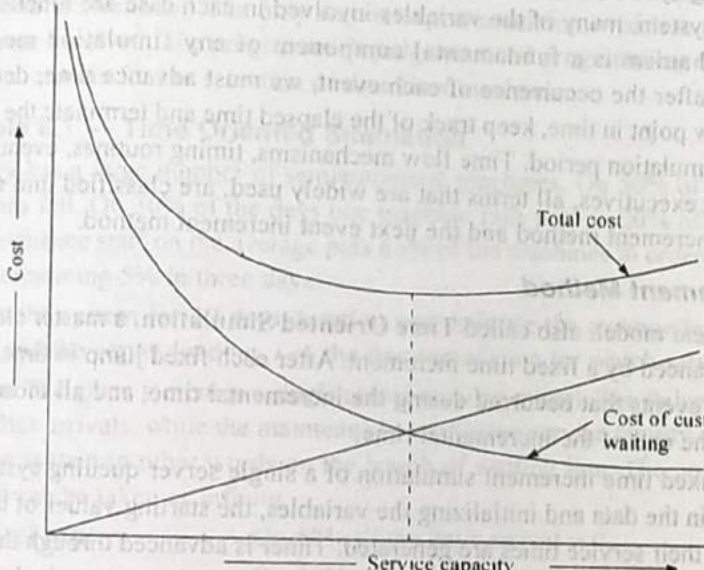


Fig. 6.3

6.7 Kendall's Notation

Kendall's Notation for specifying the characteristics of a queue is V/W/X/Y/Z, where,

- V indicates the arrival pattern,
- W indicates the service pattern,
- X gives the number of servers,
- Y represents the system capacity, and
- Z indicates the queue discipline.

The symbols used for the inter-arrival times, service times, and the queue discipline are,

Queue characteristic	*Symbol*	*Meaning*
Inter-arrival time or	D	Deterministic
Service time	M	Exponential
	Ek	Erlang with value of parameter *k*
		Any other distribution
Queue discipline	FIFO	First-in First-out
	LIFO	Last-in First-out
	SIRO	Service in random order
	PRI	Priority ordering
	GD	Any other specified ordering

If the capacity Y is not specified, it is taken to be ∞ (infinite) and if the queue discipline is not specified, it is taken as FIFO.

An M/D/2/5/FIFO stands for a queuing system having exponential arrival times, deterministic service times, two servers, with a capacity of 5 customers and having the first in first out queue discipline.

And an M/D/2 will mean exponential inter-arrival times, deterministic service times, two servers, infinite system capacity and FIFO queue discipline.

6.8 Time Flow Mechanism

Time is an essential element of most of the real life dynamic systems. It may be a queuing situation, a manufacturing system, an inventory control system, a maintenance and replacement problem or a computing system, many of the variables involved in each case are functions of time. Thus the time flow mechanism is a fundamental component of any simulation model. While conducting a simulation, after the occurrence of each event, we must advance time, determine the state of the system at a new point in time, keep track of the elapsed time and terminate the experiment at the end of a specified simulation period. Time flow mechanisms, timing routines, event scheduling procedures, or simulation executives, all terms that are widely used, are classified into two general methods, the fixed time increment method and the next event increment method.

6.8.1 *Fixed Time Increment Method*

In fixed time increment model, also called **Time Oriented Simulation,** a master clock or timer is simulated, which is advanced by a fixed time increment. After each fixed jump in time, the system is scanned to identify the events that occurred during the incremental time, and all those events are treated as if occurred at the end of the incremental time.

Flow chart for the fixed time increment simulation of a single server queuing system is given in Fig. 6.4. After reading in the data and initializing the variables, the starting values of the variables as number of arrivals and their service times are generated. Timer is advanced through the fixed time interval. The time increment is so selected that the probability of more than one arrival or more than one departure during this time is negligible. System is checked for arrival, if there is one it is added to the queue. Then the server is checked if busy or idle. If idle, time is updated, a customer is taken from the queue if available, otherwise, server remains idle. If server is busy, waiting time of the customer is updated. Then timer is advanced to next step. Process is continued until the simulation runs through the specified length of time.

6.8.2 *Next Event Increment Simulation*

In the next event increment model, also called the **Event Oriented Simulation,** the timer is advanced from event to event. At each point in time, the next earliest event is identified and the clock is advanced to that event. The state of the system is updated at each event.

Flow chart for the next event increment simulation of a single server queuing system is given in Fig. 6.5. After reading the data and initializing the variables, one arrival time and one service time are generated, assuming that the first arrival takes place at zero time. The next event, the event to occur at the earliest is then identified. In single server case, only two events are possible, the next arrival time and the next departure (service completion) time. If arrival time (AT) is earlier, clock is advanced to AT, the arrival is added to the queue, and next arrival is generated. If departure time (DT) is earlier, timer is advanced to DT and queue is checked, if a customer is available, it is put on service, clock is advanced to AT and next arrival and departure are generated. The process continues till the simulation ends.

6.8.3 *Comparison of the Two Methods*

The fixed time increment model is always used in the simulation of continuous systems, while both the fixed time and the next event time flow mechanisms are employed in discrete simulation. When the types of events encountered in the system are not very large, next event incrementing may be preferred. Since, in this case the system is examined only at those points in time, where some events happen, next event increment method proves economical. On the other hand in a complex system where the number of events occurring in a small interval of time is very large, it becomes difficult to develop a computer program by using the next event increment method. In such situations, the fixed time incrementing may prove more beneficial in programming as well as in computation time.

However, there is no yardstick to decide which method would be computationally more efficient for a given simulation model. The only way is experimentation, which is seldom justified by the amount of labour involved. The programmer's judgement and programming convenience are thus the most important criteria for the selection of time flow mechanism.

6.9 Example 6.1 — Time Oriented Simulation

A factory has a large number of semiautomatic machines. On 50% of the working days none of the machines fail. On 30% of the days one machine fails and on 20% of the days two machines fail. The maintenance staff on the average puts 65% of the machines in order in one day, 30% in two days and the remaining 5% in three days.

Simulate the system for 30 days duration and estimate the average length of queue, average waiting time, and the server loading, *i.e.*, the fraction of time for which server is busy.

Solution: The given system is a single server queuing model. The failure of the machines in the factory generates arrivals, while the maintenance staff is the service facility. There is no limit on the capacity of the system in other words on the length of waiting line. The population of machines is very large and can be taken as infinite.

Arrival pattern: On 50% of the days arrival = 0
On 30% of the days arrival = 1
On 20% of the days arrival = 2

Expected arrival rate $= 0 \times .5 + 1 \times .3 + 2 \times .2 = 0.7$ per day.

Service pattern: 65% machines in 1 day
30% machines in 2 days
05% machines in 3 days

Average service time $= 1 \times .65 + 2 \times .3 + 3 \times .05 = 1.4$ days

Expected service rate $= 1/1.4 = 0.714$ machines per day.

The expected arrival rate is slightly less than the expected service rate and hence the system can reach a steady state. For the purpose of generating the arrivals per day and the services completed per day the given discrete distributions will be used.

Random numbers between 0 and 1 will be used to generate the arrivals as under.

$0.0 < r <= 0.5$ Arrivals = 0
$0.5 < r <= 0.8$ Arrivals = 1
$0.8 < r <= 1.0$ Arrivals = 2

Similarly, random numbers between 0 and 1 will be used for generating the service times (ST).

$0.0 < r <= 0.65$ ST = 1 day
$0.65 < r <= 0.95$ ST = 2 days
$0.95 < r <= 1.0$ ST = 3 days

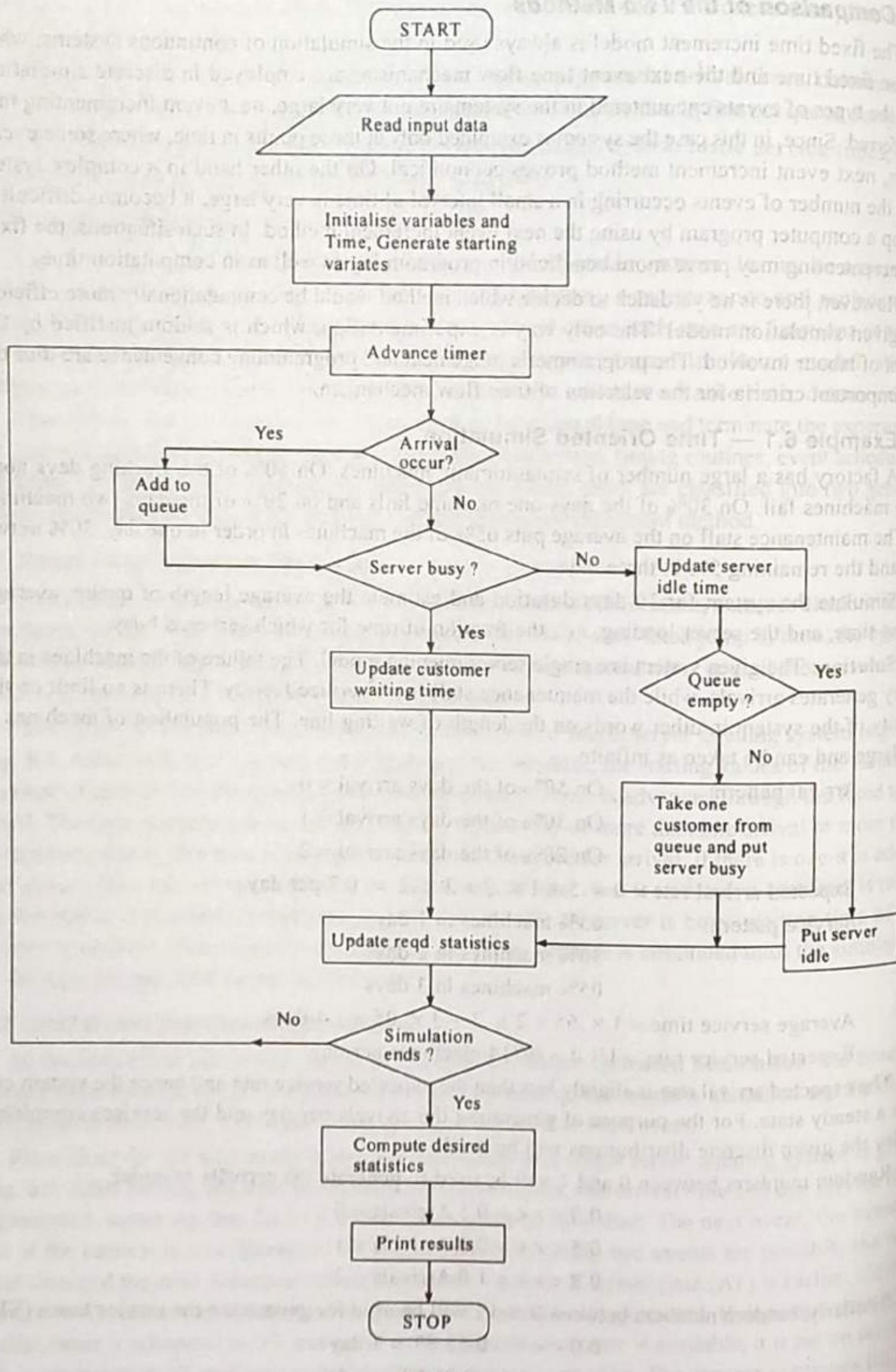

START
Read input data
Initialise variables and Time, Generate starting variates
Advance timer
Arrival occur?
Yes
Add to queue
No
Server busy ?
No
Update server idle time
Yes
Update customer waiting time
Queue empty ?
Yes
No
Take one customer from queue and put server busy
Put server idle
Update reqd. statistics
Simulation ends ?
No
Yes
Compute desired statistics
Print results
STOP

Fig. 6.4

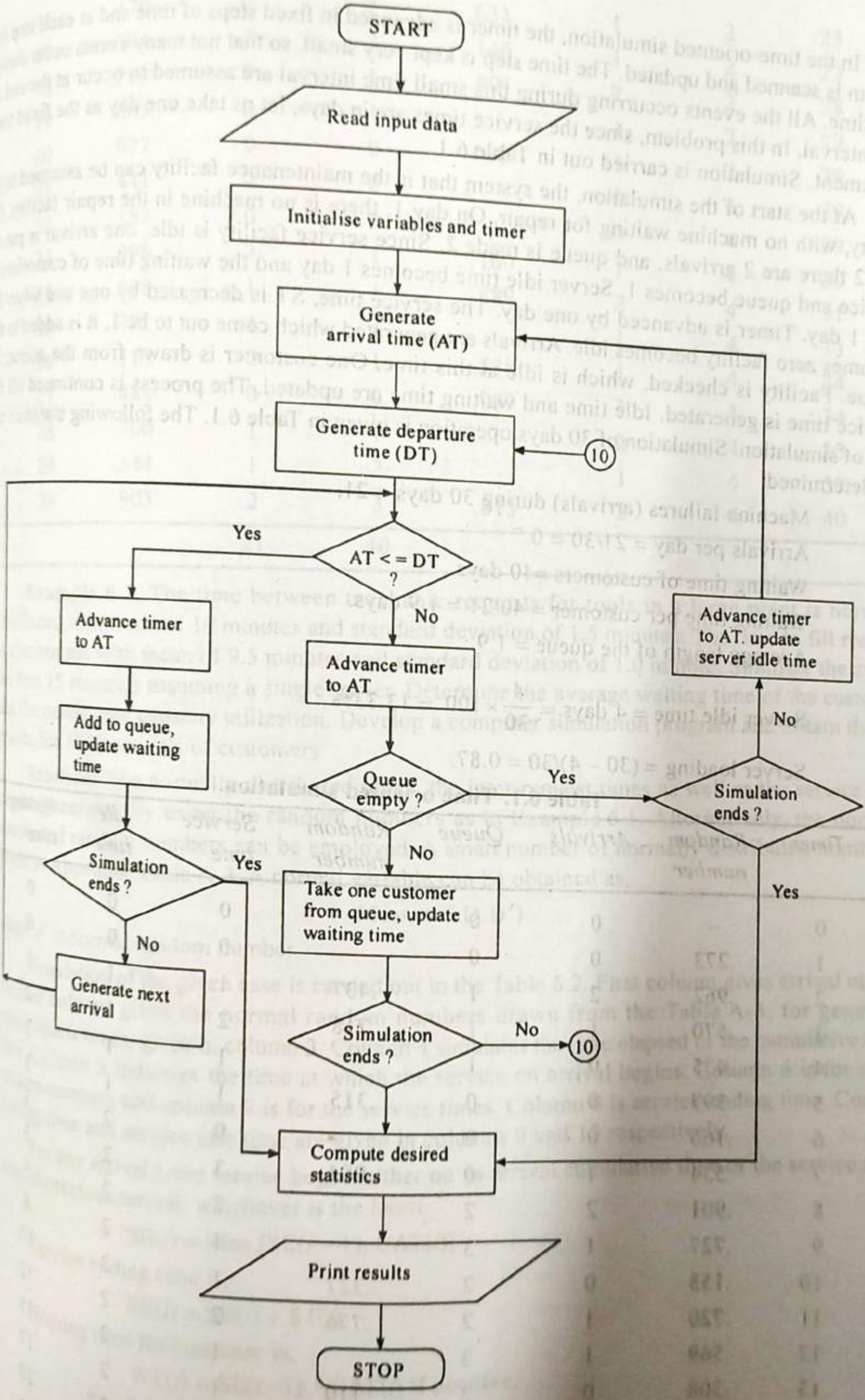

START
Read input data
Initialise variables and timer
Generate arrival time (AT)
Generate departure time (DT)
10
AT <= DT ?
Yes
No
Advance timer to AT
Add to queue, update waiting time
Simulation ends ?
Yes
No
Generate next arrival
Advance timer to AT
Queue empty ?
Yes
No
Take one customer from queue, update waiting time
Simulation ends ?
No
10
Advance timer to AT, update server idle time
No
Simulation ends ?
Yes
Compute desired statistics
Print results
STOP

Fig. 6.5

In the time-oriented simulation, the timer is advanced in fixed steps of time and at each step the system is scanned and updated. The time step is kept very small, so that not many events occur during this time. All the events occurring during this small time interval are assumed to occur at the end of the interval. In this problem, since the service times are in days, let us take one day as the fixed time increment. Simulation is carried out in Table 6.1.

At the start of the simulation, the system that is the maintenance facility can be assumed to be empty, with no machine waiting for repair. On day 1, there is no machine in the repair facility. On day 2 there are 2 arrivals, and queue is made 2. Since service facility is idle, one arrival is put on service and queue becomes 1. Server idle time becomes 1 day and the waiting time of customers is also 1 day. Timer is advanced by one day. The service time, ST is decreased by one and when ST becomes zero facility becomes idle. Arrivals are generated which come out to be 1, it is added to the queue. Facility is checked, which is idle at this time. One customer is drawn from the queue, its service time is generated. Idle time and waiting time are updated. The process is continued till the end of simulation. Simulation of 30 days operation is given in Table 6.1. The following statistics can be determined.

Machine failures (arrivals) during 30 days = 21.

Arrivals per day = 21/30 = 0."

Waiting time of customers =40 days.

Waiting time per customer = 40/21 = 1.9 days.

Average length of the queue = 1.9

Server idle time = 4 days = $\frac{4}{30} \times 100$ = 13.33%

Server loading = (30 − 4)/30 = 0.87.

Table 6.1. Time oriented simulation

Timer	*Random number*	*Arrivals*	*Queue*	*Random number*	*Service time*	*Idle time*	*Waiting time*
0	–	0	0		0	0	0
1	.273	0	0		0	0	0
2	.962	2	1	.437	1	1	1
3	.570	1	1	.718	2	1	2
4	.435	0	1		1	1	3
5	.397	0	0	.315	1	1	3
6	.166	0	0		0	2	3
7	.534	1	0	.964	3	2	3
8	.901	2	2		2	2	5
9	.727	1	3		1	2	8
10	.158	0	2	.327	1	2	10
11	.720	1	2	.776	2	2	12
12	.569	1	3		1	2	15
13	.308	0	2	.110	1	2	17
14	.871	2	3	.469	1	2	20
15	.678	1	3	.462	1	3	23

16	.470	0	2	.631	1	2	25
17	.794	1	2	.146	1	2	27
18	.263	0	1	.801	2	2	28
19	.065	0	1		1	2	29
20	.027	0	0	.086	1	2	29
21	.441	0	0		0	3	29
22	.152	0	0		0	4	29
23	.998	2	1	.160	1	4	30
24	.508	1	1	.889	2	4	31
25	.771	1	2		1	4	33
26	.115	0	1	.538	1	4	34
27	.484	0	0	.989	3	4	34
28	.700	1	1		2	4	35
29	.544	1	2		1	4	37
30	.903	2	3	.813	2	4	40
		21	40				

Example 6.2. The time between mechanic requests for tools in a large plant is normally distributed with mean of 10 minutes and standard deviation of 1.5 minutes. The time to fill requests is also normal with mean of 9.5 minutes and standard deviation of 1.0 minute. Simulate the system for first 15 requests assuming a single server. Determine the average waiting time of the customers and the percentage capacity utilization. Develop a computer simulation program and obtain the said results for 1000 arrivals of customers.

Solution: The normally distributed times, the inter-request times as well as the service times can be generated by using the random numbers as in Example 6.1. Alternatively, the normally distributed random numbers can be employed. A small number of normally distributed numbers is given in Appendix Table A-3. A normal variable can be obtained as,

$$x = \text{Mean} + \text{S.D.}(r')$$

where r' is normal random number.

Simulation of the given case is carried out in the Table 6.2. First column gives arrival number. Second column gives the normal random numbers drawn from the Table A-3, for generating inter-request times, given in column 3. Column 4 simulates the time elapsed or the cumulative arrival time. Column 5 indicates the time at which the service on arrival begins. Column 6 is for normal random numbers and column 7 is for the service times. Column 8 is service-ending time. Customer waiting time and service idle time are given in columns 9 and 10 respectively.

For any arrival i, the service begins either on its arrival cumulative time or the service ending time of previous arrival, whichever is the latest.

$$SB(i) = \text{Max.}[SE(i-1), CAT(i)]$$

Service ending time is,

$$SE(i) = SB(i) + ST(i)$$

Waiting time for customer is,

$$WT(i) = SE(i-1) - CAT(i) \text{ if positive.}$$

Service idle time is,

$$IT(i) = CAT(i) - SE(i-1), \text{ if positive.}$$

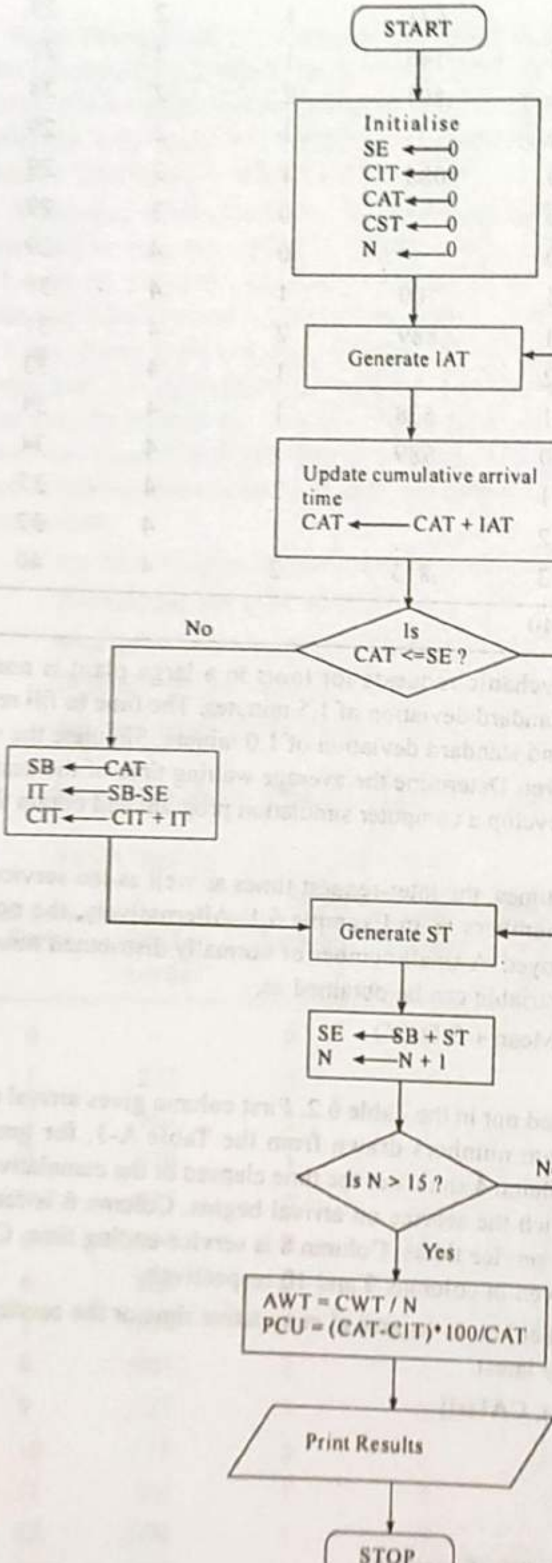

START
Initialise
SE ← 0
CIT ← 0
CAT ← 0
CST ← 0
N ← 0
Generate IAT
Update cumulative arrival time
CAT ← CAT + IAT
Is CAT <= SE ?
No
Yes
SB ← CAT
IT ← SB-SE
CIT ← CIT + IT
SB ← SE
IT ← SE-CAT
CIT ← CWT + WT
Generate ST
SE ← SB + ST
N ← N + 1
Is N > 15 ?
No
Yes
AWT = CWT / N
PCU = (CAT-CIT)*100/CAT
Print Results
STOP

Fig. 6.6

Flow chart for this simulation is given in Fig. 6.6.

At the end of 15 simulations,

Total time elapsed = 159.615 mins.

Total customer waiting time = 2.89 mins.

Total server idle time = 26.915 mins.

Since there are 15 arrivals,

Average waiting time = 2.89/15 = 0.193 per arrival.

Percentage capacity utilization = (167.755 – 26.915)/(167.755)*100 = 83.96%.

Table 6.2. Simulation of a single server queuing system

1	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*
Arrival number *i*	*Normal random number* *r'*	*Time between arrivals* *IAT*	*Cumm. arrival time* *CAT*	*Service begins* *SB*	*Normal random number* *r'*	*Service time* *ST*	*Service ends time* *SE*	*Cust. waiting time* *WT*	*Server idle* *IT*
1	–0.46	9.31	9.31	9.31	0.59	10.09	19.40	0.00	9.31
2	–1.15	8.275	17.585	19.40	–0.67	8.83	28.23	1.815	0.00
3	0.15	10.225	27.81	28.23	0.41	9.91	38.14	0.42	0.00
4	0.81	11.215	39.025	39.025	0.51	10.01	49.035	0.00	0.885
5	0.74	11.11	50.135	50.135	1.53	11.03	61.165	0.00	1.10
6	–0.39	9.415	59.55	61.165	–0.37	9.13	70.295	1.615	0.00
7	0.45	10.675	70.225	60.295	–0.27	9.23	79.525	0.07	0.00
8	2.44	13.66	83.885	83.885	–0.15	9.35	93.235	0.00	4.36
9	0.59	10.885	94.77	94.77	–0.02	9.48	104.25	0.00	1.535
10	–0.06	9.91	104.68	104.68	–1.60	7.90	112.58	0.00	0.43
11	0.09	10.135	114.815	114.815	–0.19	9.31	124.125	0.00	2.235
12	0.56	10.84	125.655	125.655	0.16	9.66	135.315	0.00	1.53
13	0.65	10.97	136.625	136.625	–0.07	9.43	146.055	0.00	1.31
14	3.10	14.65	150.275	150.275	0.24	9.74	160.015	0.00	4.22
15	–0.44	9.34	159.615	160.015	–1.76	7.74	167.755	0.04	0.00
							Total	2.89	26.915

These 15 observations cannot be expected to give a reliable estimate of the results. Much longer simulation run is required which is possible only with a computer simulation of the system. A computer program for the single server queuing model is given below in which both the inter-arrival and service times have been taken as normally distributed.

```
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
main()
  {
```

```
/*Single server queue:
Arrivals and service times are normally distributed.
Mean and standard deviation of arrivals are 10. and 1.5 minutes
mean and S.D. of service times are 9.5 and 1.0 minutes.*/
int kk,i,j,run=100;
float x,iat,st,awt,pcu,wt=0.,it=0. ;
    float mean=10. , sd=1.5 , mue=9.5 , sigma=1.0 ;
    float sb=0., se=0.,cit=0.,cat=0.,cwt=0.;
printf("\n  IAT     CAT      SB      ST      SE      CWT     CIT");
    for(j=1;j<=run;++j) {
    /* generate inter arrival time */
    float sum=0.;
    for(i=1;i<=12;++i) {
    x=rand()/32768.0;
    sum=sum+x; }
    iat=mean+sd*(sum-6.);
    cat=cat+iat;
    /*printf("\n iat= %6.2f cat=%6.2f",iat,cat);*/
    if(cat<=se) {sb=se;wt=se-cat;cwt=cwt+wt;}
    else{sb=cat; it=sb-se; cit=cit+it;}

    /*generate service time*/
    sum=0.;
    for(i=1;i<=12;++i) {
    x=rand()/32768.0;
    sum=sum+x;              }
    st=mue+sigma*(sum-6.);
    se=sb+st;
    printf("\n %5.2f %6.2f %6.2f %6.2f %6.2f%6.2f
%6.2f",iat,cat,sb,st,se,cwt,cit);
    }
    awt=cwt/run;
    pcu=(cat-cit)*100./cat;
    printf("\n Average waiting time=%6.2f",awt);
    printf("\n Percentage Capacity Utilisation=%6.2f",pcu);
  printf("\n any digit");
    scanf("%d",kk);
    }
```

The output of this program for 1000 arrivals is as below.

Inter-arrival time: Mean = 10.00 S.D.= 1.5 minutes

Service time: Mean = 9.5 S.D. = 1.0 minutes

Length of run (Max. Arrivals) = 1000

Average waiting time = 1.65 minutes

Percentage capacity utilization = 94.68%.

6.10 Example 6.3. Event Oriented Simulation

Simulate an M/M/1/∞ queuing system with mean arrival rate as 10 per hour and the mean service rate as 15 per hour, for a simulation run of about 3 hours. Determine the average customer waiting time, percentage idle time of the server, maximum length of the queue and average length of queue.

Solution: The given queuing system has exponential arrival and exponential service distributions. There is single server and the capacity of the system is infinite.

Mean arrival rate = 10/hr or 1/6 per minute.

Mean service rate = 15/hr or 1/4 per minute.

In actual simulation, the inter-arrival times and the service times are generated, as and when the particular events occur. However, to simplify the simulation table, the inter-arrival times for the first 25 arrivals and their service times (ST) are generated in Table 6.3. The random numbers (R) have been taken from a random number table.

Table 6.3

S. No.	*Inter-arrival times* $IAT = -6 \ln(1-r)$		*Service times* $ST = -4 \ln(1-r)$	
	R	*IAT*	*R*	*ST*
1	.100	0.63	.209	0.94
2	.375	2.82	.048	0.20
3	.084	0.53	.689	4.67
4	.990	27.63	.025	0.10
5	.128	0.82	.999	27.63
6	.660	6.47	.747	5.50
7	.310	2.23	.108	0.46
8	.852	11.46	.776	5.98
9	.659	6.45	.321	1.55
10	.737	8.01	.457	2.44
11	.985	25.20	.177	0.78
12	.118	0.75	.054	0.22
13	.840	10.99	.996	22.09
14	.886	13.03	.402	2.06
15	.995	31.79	.673	4.47
16	.654	6.37	.176	0.77
17	.801	9.69	.947	11.76
18	.743	8.15	.914	9.83
19	.159	1.04	.356	1.76
20	.283	2.00	.268	1.25
21	.699	7.20	.205	0.92
22	.098	0.62	.268	24.86
23	.802	9.71	.951	12.09
24	.813	10.05	.885	8.65
25	.543	4.70	.437	2.30

Simulation of the system is illustrated in Table 6.4. Event to event time flow mechanism has been employed. System is assumed to be empty at zero time. First arrival takes place at 0.63 minutes; clock is advanced to that time. Service on the customer is generated and the same ends at 1.57 mins. The next arrival is to occur at 3.45 mins. Clock is advanced to the earliest event, *i.e.*, to 1.57 mins.

As soon as an arrival occurs, the server is checked, if free, service on customer begins, otherwise, it is added to queue. As soon as a service ends, the queue is checked, if there is a customer waiting, service on the customer begins, otherwise the server becomes idle. For example, at time 31.61 service on a customer begins, while the previous service ended at 8.65 minutes giving 31.61 – 8.65 = 22.96 mins waiting time. The waiting time of customers is the number of customers waiting, multiplied by the incremental time. For example, between 41.3 and 52.59 minutes, 2 customers are in queue, giving waiting time 2 (52.29 – 41.13) = 22.92 minutes.

Flow chart for the given simulation is given in Fig. 6.7. It simulates the cumulated server idle time and the cumulated customer waiting time from which various results can be computed.

Following statistics are obtained from the simulation Table 6.4.

Time elapsed or Simulation run =183.26 mins.

Number of arrivals = 22

Cumulated waiting time for customers ≐ 106.55 mins.

Cumulated idle time of server = 85.01 mins.

Taking the different steps in time as the observations, the number of observations made is 39.

Length of the queue has been observed to be

0 on 23 times
1 on 6 times
2 on 5 times
3 on 4 times
4 on 1 time

From this information, following results are obtained.

(*a*) Average waiting time = 106.55/22 = 4.84 mins.

(*b*) Server idle time = (85.01/183.26) 100 = 46.39%

(*c*) Maximum length of the queue = 4

(*d*) Average length of the queue = (23 × 0 + 6 × 1 + 5 × 2 + 4 × 3 + 4)/39 = 32/39 = 0.82

Table 6.4. Event oriented simulation

Clock time	*Inter-arrival time IAT*	*Next arrival time NAT*	*Queue Q*	*Service begins SB*	*Service time ST*	*Service ends SE*	*Server idle time IT*	*Customer waiting time WT*
0.00	0.63	0.63	0	–		0.00	0.00	0.00
0.63	2.82	3.45	0	0.63	0.94	1.57	0.63	–
1.57		3.45	0					
3.45	0.53	3.98	0	3.45	0.20	3.65	1.88	–
3.65		3.98	0					
3.98	27.63	31.61	0	3.98	4.67	8.65	0.33	–
8.65		31.61	0					
31.61	0.82	32.43	0	31.61	0.10	31.71	22.96	–

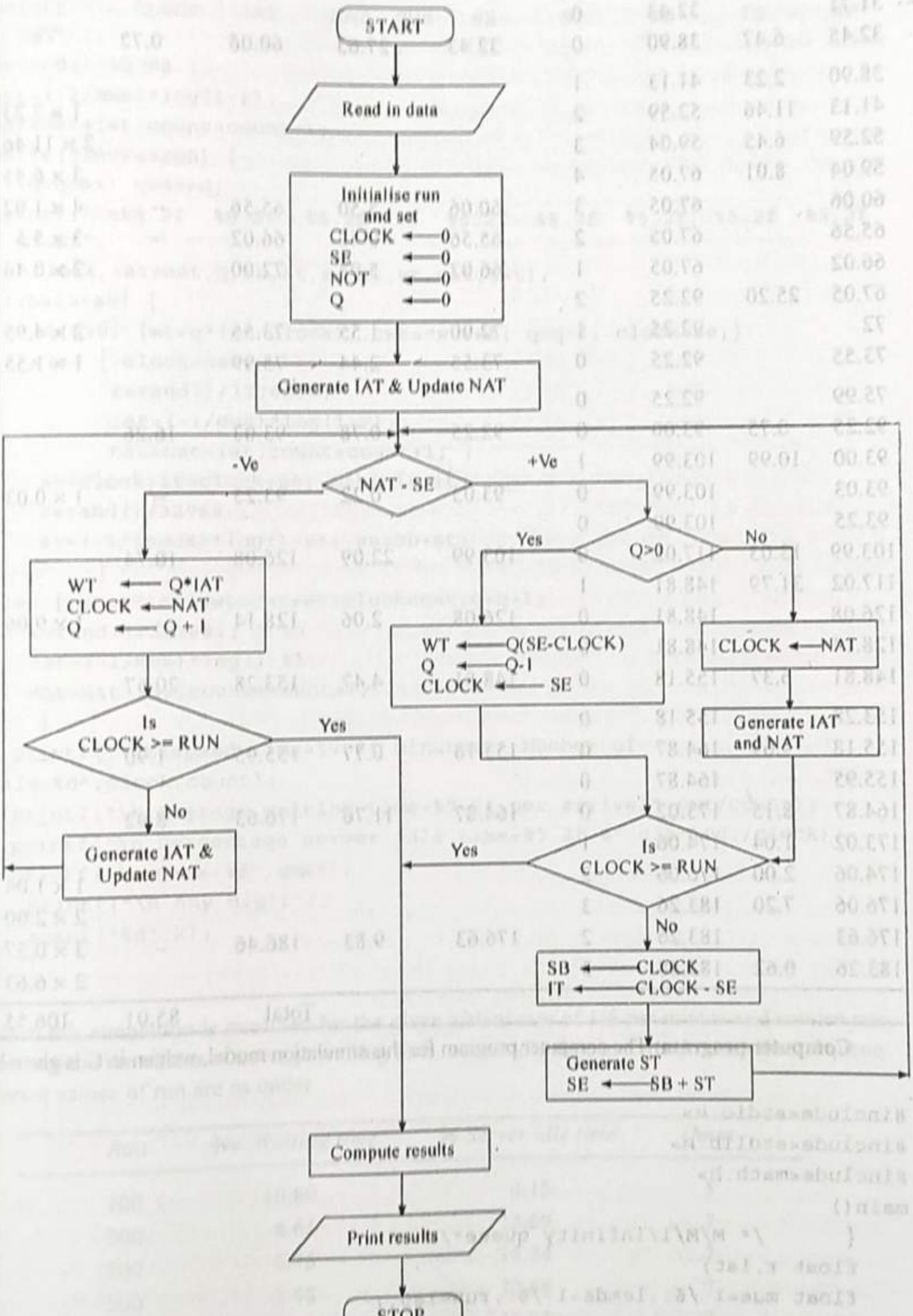

START
Read in data
Initialise run and set
CLOCK ⟵ 0
SE ⟵ 0
NOT ⟵ 0
Q ⟵ 0
Generate IAT & Update NAT
-Ve
+Ve
NAT - SE
Yes
No
Q>0
WT ⟵ Q*IAT
CLOCK ⟵ NAT
Q ⟵ Q + 1
WT ⟵ Q(SE-CLOCK)
Q ⟵ Q-1
CLOCK ⟵ SE
CLOCK ⟵ NAT
Is
CLOCK >= RUN
Yes
No
Generate IAT and NAT
Generate IAT & Update NAT
Yes
Is
CLOCK >= RUN
No
SB ⟵ CLOCK
IT ⟵ CLOCK - SE
Generate ST
SE ⟵ SB + ST
Compute results
Print results
STOP

Fig. 6.7

31.71		32.43	0					
32.45	6.47	38.90	0	32.43	27.63	60.06	0.72	–
38.90	2.23	41.13	1					
41.13	11.46	52.59	2					1 × 2.23
52.59	6.45	59.04	3					2 × 11.46
59.04	8.01	67.05	4					3 × 6.45
60.06		67.05	3	60.06	5.50	65.56	–	4 × 1.02
65.56		67.05	2	65.56	0.46	66.02	–	3 × 5.5
66.02		67.05	1	66.02	5.98	72.00	–	2 × 0.46
67.05	25.20	92.25	2					
72		92.25	1	72.00	1.55	73.55	–	2 × 4.95
73.55		92.25	0	73.55	2.44	75.99	–	1 × 1.55
75.99		92.25	0					
92.25	0.75	93.00	0	92.25	0.78	93.03	16.26	–
93.00	10.99	103.99	1					
93.03		103.99	0	93.03	0.22	93.25	–	1 × 0.03
93.25		103.99	0					
103.99	13.03	117.02	0	103.99	22.09	126.08	10.74	–
117.02	31.79	148.81	1					
126.08		148.81	0	126.08	2.06	128.14	–	1 × 9.06
128.14		148.81	0					
148.81	6.37	155.18	0	148.81	4.47	153.28	20.67	–
153.28		155.18	0					
155.18	9.69	164.87	0	155.18	0.77	155.95	1.90	–
155.95		164.87	0					
164.87	8.15	173.02	0	164.87	11.76	176.63	8.92	–
173.02	1.04	174.06	1					
174.06	2.00	176.06	2					1 × 1.04
176.06	7.20	183.26	3					2 × 2.00
176.63		183.26	2	176.63	9.83	186.46	–	3 × 0.57
183.26	0.62	183.88	3					2 × 6.63
						Total	85.01	106.55

Computer program: The computer program for this simulation model, written in C is given below.

```
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
main()
    {       /* M/M/1/infinity queue*/
    float r,iat;
    float mue=1./6.,lemda=1./5.,run=180.;
    float clock=0.,se=0.,sb=0.,nat=0.,cit=0.,cwt=0.,st=0.,it=0.,wt=0.;
    int q=0,cq=0,k,count=0,qmax=0;
```

```
  printf("\n CLOCK     IAT    NAT  QUE  SB     ST     SE     IT     WT
CIT  CWT" );
  r=rand()/32768.;
  iat=(-1/mue)*log(1-r);
  nat=nat+iat;count=count+1;
  while(clock<=run) {
  if(q>qmax) qmax=q;
  printf("\n%5.2f  %5.2f  %5.2f %3d  %5.2f  %5.2f  %5.2f  %5.2f  %5.2f
%5.2f %5.2f",
     clock,iat,nat,q,sb,st,se,it,wt,cit,cwt);
  if(nat>=se) {
     if(q>0) {wt=q*(se-clock); cwt=cwt+wt; q=q-1; clock=se;}
     else { clock=nat;
          r=rand()/32768.0;
          iat=(-1/mue)*log(1-r);
          nat=nat+iat;count=count+1; }
      sb=clock;it=clock-se; cit=cit+it;
      r=rand()/32768.;
      st=(-1/lemda)*log(1-r); se=sb+st;
      }
  else { wt=q*iat;cwt=cwt+wt;clock=nat;q=q+1;
     r=rand()/32768.;
     iat=(-1/mue)*log(1-r);
     nat=nat+iat;count=count+1; }
      }
   printf("\nElapsed time=%7.2f minutes Number of
arrivals=%d",clock,count);
   printf("\n Average waiting time=%7.2f per arrival",cwt/count);
   printf("\n Percentage server idle time=%7.2f %",cit*100./clock);
   printf("\n Qmax=%d",qmax);
    printf("\n Any digit");
    scanf("%d",k);
    }
```

When this simulation is executed for the given arrival rate of 1/6 per minute and service rate of 1/4 per minute the values of average waiting time, server idle time and maximum value of queue, for different values of run are as under.

Run	*Ave. Waiting time*	*% Server idle time*	*Qmax*
100	10.80	2.15	5
200	8.61	13.69	7
300	6.73	15.34	7
500	5.48	32.68	7
1000	7.63	35.07	7
2000	7.74	32.59	7

Thus the system stabilizes after some time with average waiting time of about 7.65 minute per arrival, the server idle time of about 35.00% and maximum queue at 7.

When the arrival rate and the service rate are made identical, because of the variability in times the average waiting time goes on increasing with passage of time while the server idle time goes on decreasing, the queue length as well as the maximum queue goes on increasing. The output of the program, for both the arrival and service rates of 1/6 per minute, is given below.

Run	*Ave. waiting time*	*% Server idle time*	*Qmax*
100	3.15	33.47	2
200	11.28	20.52	11
300	29.56	14.65	16
500	66.14	8.81	17
1000	61.32	7.29	17
2000	68.13	7.02	20

This is an unstable system or an explosive system, where the queue goes on building with time.

Example 6.4. In an M/D/2/3 system, the mean arrival time is 3 minutes and the Servers I and II take exactly 5 and 7 minutes respectively to serve a customer. Simulate the system for the first one hour of operation. Determine the idle time of servers and the waiting time of the customers.

Solution: In the M/D/2/3 system, the arrivals are distributed exponentially, while the services times are deterministic. There are two servers, Server I takes exactly 5 minutes to serve the customer, while Server II takes exactly 7 minutes. Capacity of the system is 3, *i.e.*, the number of customers in service as well as in the waiting line cannot exceed 3. The next customer in that case will be returned without service. The queue discipline will be taken as FIFO.

The mean inter-arrival time = 3 mins.

Or arrival rate $\lambda = 1/3$ per minute

The random observation x, for exponential distribution is generated as,

$$x = (-1/\lambda) \times \ln (1 - r)$$

where r is a random number between 0 and 1.

For operation time of 60 minutes, we will require about 20 arrivals; let us generate 25, for which 25 random numbers are required. Since r and $1 - r$ are equally likely, both the random number and its complementary random number can be used. This also helps to reduce the variance. The following string of random numbers has been taken from a table.

.218, .782, .119, .881, .711, .289, .344, .420, .580, .656, .354, .696, .175, .825, .913, .350, .087, .076, .944, .650, .130, .670, .440, .560, .569.

The inter-arrival times (in minutes) generated by using these random numbers are.

0.74, 4.57, 0.38, 6.39, 3.72, 1.02, 1.26, 1.63, 2.60, 3.20, 1.09, 3.12, 0.58, 5.23, 7.32, 1.29, 0.28, 0.24, 8.65, 3.15, 1.20, 6.12, 1.74, 2.46, 2.52.

Working of the system is simulated in Table 6.5. It is assumed that at zero time, there is no customer in the system. The first arrival takes place 0.74 minutes after the system starts. The first customer is assigned to Server I. The second customer arrives 4.57 minutes after the first, *i.e.*, at 5.31 minutes according to the simulated clock, and is assigned to Server II. Third arrival occurs at 5.69 minutes, and since both servers are busy it waits in queue. Server I completes service on customer one at 5.74 minutes, and customer 3 moves to Server I. This process continues. When customer 8 arrives, Server I is busy with customer 6 and Server II is busy with customer 5, while customer 7 is in queue. The system being full to capacity, customer number 8 returns without getting service. Similar situation occurs at clock times 31.02 and 45.38 minutes. When both servers are idle as at times 53.57 and 64.18 mins, the next arrival will go to the faster server, *i.e.*, Server I.

Table 6.5. Event oriented simulation

Clock	*Arrival No.*	*IAT*	*NAT*	*Servr I cstmr. in service*	*SE1*	*IT1*	*Servr II cstmr. in service*	*SE2*	*IT2*	*Cstmr. in queue*	*Cstmr. wtng time*
0.00	1	0.74	0.74	0	–	–	0	–	–	–	–
0.74	2	4.57	5.31	1	5.74	0.74	0	–	–	–	–
5.31	3	0.38	5.69	1			2	12.31	5.31	–	–
5.69	4	6.39	12.08	1			2			3	–
5.74				3	10.74	0.00	2			–	0.05
10.74							2			–	–
12.08	5	3.72	15.80	4	17.08	1.34	2			–	–
12.31				4			–			–	–
15.80	6	1.02	16.82	4			5	22.80	3.49	–	–
16.82	7	1.26	18.08	4			5			6	–
17.08				6	22.08	0.00	5			–	0.26
18.08	8	1.63	19.71	6			5			7	–
19.71	9	2.60	22.31	6			5			7,8	–
22.08				7	27.08	0.00	5			–	4.00
22.31	10	3.20	25.51	7			5			9	–
22.80				7			9	29.80	0.00	–	0.49
25.51	11	1.09	26.60	7			9			10	–
26.60	12	3.12	29.72	7			9			10,11	–
27.08				10	32.08	0.00	9			–	1.57
29.72	13	0.58	30.30	10			9			12	–
29.80				10			12	36.80	0.00	–	0.08
30.30	14	5.23	35.53	10			12			13	–
32.08				13	37.08	0.00	12			–	1.78
35.53	15	7.32	42.87	13			12			–	–
35.53	15	7.32	42.87	13			12			14	–
36.80				13			14	43.80	0.00	–	1.27
37.08				–			14			–	–
42.87	16	1.29	44.16	15	47.87	5.79	14			–	–
43.80				15			–			–	–
44.16	17	0.28	44.44	15			16	51.16	7.36	–	–
44.44	18	0.24	44.68	15			16			17	–
44.68	19	8.65	53.33	15			16			17,18	–
47.87				17	52.87	0.00	16			–	3.43

51.16				17			–			–	–
52.87				–			–			–	–
53.33	20	3.15	56.48	19	58.33	5.46	–			–	–
56.48	21	1.20	57.68	19			20	63.48	5.32	–	–
57.68	22	6.12	63.80	19			20			21	–
58.33				21	63.33	0.00	20			–	0.65
63.33				–			20			–	–
63.48				–			–			–	–
63.80	23	1.74	65.54	22	68.80	0.47	–			–	–
Total						13.80			21.48		13.58

From the simulation in Table 6.5, we observe that in the first 63.80 minutes of operation $21 - 3 = 18$ customers have been served, while 3 customers have been returned without service. The idle time of Server I is 13.80 minutes, while that of Server II is 21.48 minutes. The customer had to wait in line a total of 13.58 minutes. Thus, we find that while about 31.71% of the time, there had been a customer in the queue, Server I remained idle for about 21.63% of the time and Server II for about 33.67% of the time.

State of the system at 63.80 mins is,

Number of customers arrived = 22

Customers in service= one (number 22)

Number of customers returned without service = 3 (Nos. 8, 11 and 18)

Number of customers served = 21 − 3 = 18

Idle time of Server I = 13.80 minutes

Idle time of Server II = 21.48 minutes

Out of 41 observations, there was queue on 13 times

Waiting time of customers =13.58 mins.

% Idle time of Server I = (13.80/63.80) 100 = 21.63%

% Idle time of Server II = (21.48/63.80) 100 = 33.67%

The computer simulation program of this M/D/2/3 model is given below.

```
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
main()
{     /*M/D/2/3 queuing system*/
   float r,iat,clock,nat,it1,it2,run=150.,cit1=0.,cit2=0.;
   float mean=3.,lemda1=5.,lemda2=4.,se1=0.,se2=0.;
   int k,q=0, qmax=3,kont=0,counter;
   printf("\n  CLOCK    IAT     NAT     SE1     SE2  QUE  KONT cit1  cit2");
   /*generate first arrival*/
   r=rand()/32768.;
   iat=( -mean)*log(1-r);
   nat=nat+iat; se1=lemda1;   counter=1;

   printf("\n %6.2f %6.2f %6.2f  %6.2f %6.2f  %d    %d %6.2f %6.2f ",
            clock,iat,nat,se1,se2,q,kont,cit1,cit2);
   while(clock<=run) {
```

```
if(nat<=se1 && nat<=se2) {
   clock=nat;q=q+1;
   r=rand()/32768.;
   iat=(-mean)*log(1-r);
   nat=nat+iat; counter=counter+1;}
else if(se1<= nat && se1<=se2) clock=se1;
     else clock=se2;

if(q>qmax){kont=kont+1;
   q=q-1;}
if(q>=1 && se1<=clock) {
   it1=clock-se1; cit1=cit1+it1;
   se1=clock+lemda1;
   q=q-1;}

if(q>=1 && se2<=clock) {
   it2=clock-se2; cit2=cit2+it2;
   se2=clock+lemda2;q=q-1;}
if(q==0 && se1<=clock) {
   clock=nat;it1=clock-se1;cit1=cit1+it1;
   se1=nat+lemda1;
   r=rand()/32768.;
   iat=(-mean)*log(1-r);
   nat=nat+iat; counter=counter+1;}
if(q==0 && se2<=clock) {
   clock=nat;it2=clock-se2;cit2=cit2+it2;
   se2=nat+lemda2;
   r=rand()/32768.;
   iat=(-mean)*log(1-r);
   nat=nat+iat; counter=counter+1;
   }

printf("\n %6.2f %6.2f %6.2f  %6.2f %6.2f %d      %d %6.2f %6.2f ",
        clock,iat,nat,se1,se2,q,kont,cit1,cit2);
}
printf("\n clock=%8.2f cit1=%6.2f cit2=%6.2f counter=%d",
        clock,cit1,cit2,counter);
printf("\n\n                          Queuing System:M/D/2/3");
printf("\n\n Mean arrival time=%5.2f minutes exponentially
distributed",mean);
printf("\n Service time Server I=%5.2fminutes   Server II=%5.2f
minutes"
        lemda1,lemda2);
printf("\n Simulation Run (Elapsed time)=%7.2f minutes",clock);
printf("\n Number of customers arrived=%d",counter);
printf("\n Number of customers returned without service=%d",kont);
printf("\n Idle Time of Server I=%6.2f minutes",cit1);
```

```
    printf("\n Idle time of Server II=%6.2f minutes",cit2);
    printf("\n Percentage Idle time of Server I=%6.2f %",cit1*100./
clock);
    printf("\n Percentage Idle time of Server II=%6.2f %",cit2*100./
clock);


    printf("\n any digit");
    scanf("%d",&k);
    }
```

The output of this program when executed for 1000 minutes is given below, which shows that the idle time of the two servers tends to be identical in the long run.

Queue M/D/2/3

Mean arrival time = 3.00 minutes exponentially distributed.

Service time Server I = 5.00 minutes   Server II = 7.00 minutes.

Simulation run (Elapsed time) = 1001.13 minutes.

Number of customers arrived = 334.

Number of customers returned without service = 45.

Idle time of Server I = 177.72 minutes.

Idle time of Server II = 151.33 minutes.

Percentage Idle time of Server I = 17.75%

Percentage Idle time of Server II = 15.12%

6.11 Two Servers in Parallel Queuing System

The servers placed in parallel may be doing identical service as the railway ticket-windows or different types of service as clinics in OPD of a hospital, but all the servers draw customers from a single queue at the entry to the system. In systems, where different type of service is provided at parallel servers, the customers drawn from the single queue are sorted into different types and put further into different queue one before each server.

Let us consider the manufacturing system shown below in Fig 6.8, where a mixture of components A and B pass through workstation I. These are then sorted into two parts, the components A that are 90% and the components B that are 10%. Then components A are processed at workstation A and components B are processed at workstation B. Station I processes the components at a rate of one in 5 minutes with exponential distribution. Station A and B has normally distributed processing times with mean values of 4.0 and 10.0 minutes and standard deviations of 2.0 and 5.0 minutes respectively. There is no dearth of components before workstation I, and unlimited space is available before workstations A and B. Sorting and transportation consume negligible time. What is the percentage idle time of workstations A and B and what is the maximum length of queue before each of A and B?

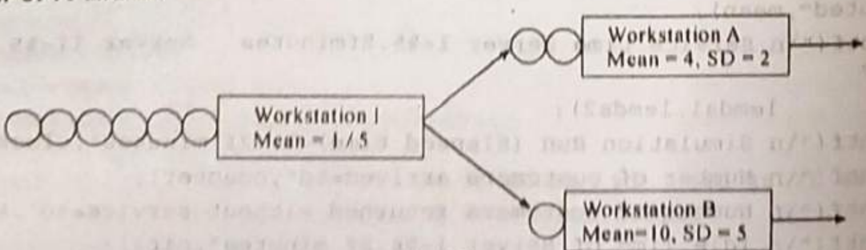


Fig. 6.8

Since there is no shortage of components at the head of workstation I, the workstation is always busy and have zero idle time. The processing rate of this workstation can be taken as the arrival rate of customers to the parallel Servers A and B from an infinite population. Each component processed at I is checked whether it is of type A or of type B and is added to the respective queue. The two work stations work independently, behaving like single server queues. The computer simulation model of this system, developed in C language, is given below.

```
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
main()
   {
   /* Simulation of a system having two servers in parallel.
      mue is arrival rate,
      meana and siga are mean and standard deviation
      of service time of server A,
      meanb and sigb are for server B.
      count is counter of arrivals while konta and
      kontb are counters for components served by A and B.
      qa(qb) is the queue length of components A(B)
      sea(seb) is the service ending on a component A(B)
      iat and nat are inter arrival time and next arrival time.
      wta(wtb) is the waiting time of components in queue A(B)
      ita(itb) is the idle time of server A(B).
      sta(stb) is the service time of server A(B).*/
      int i,count=0,konta=0,kontb=0,qa=0,qb=0;
      float r,sea=0.,seb=0.,iat,nat=0.,wta=0.,wtb=0.,
      clock=0.,ita=0.,itb=0.;
      float delt=0.1,run;
      float mue,meana,meanb,siga,sigb,time,sum,sta,stb;
      int qamax=0,qbmax=0;
      mue=1./5.;meana=4.0;siga=2.0;meanb=10.;sigb=5.;
     printf("\n\n Mean of expo.processing time of stationI=%6.2f",mue);
     printf("\n Processing time of station A: mean=%6.2f SD=%6.2f",
        meana,siga);
     printf("\n Processing time of station B: mean=%6.2f SD=%6.2f",
        meanb,sigb);

     printf("\n Enter the value of run");
     scanf("%f",&run);

     while(clock<=run) {
     /* Check the state of arrival and update queues*/
     if(clock>=nat)
      {
     /* Generate next arrival*/
     count=çount+1;
```

```
r=rand()/32768.;
iat=(-1/mue)*log(1-r);
nat=clock+iat;
/* Sort into A andB and add to respective queue*/
r=rand()/32768.;
if(r<=0.9) qa=qa+1;
else qb=qb+1;
if(qa>qamax) qamax=qa;
if(qb>qbmax) qbmax=qb;
}
/*    printf("\nclock=%6.2f iat=%6.2f nat=%6.2fqa=%d qb=%d
wta=%6.2f sta=%6.2f sea=%6.2f",
  clock,iat,nat,qa,qb,wta,sta,sea); */
/* check the state of the servers */
if(clock>=sea)
{
wta=wta+qa*delt;
if(qa>=1)
{
qa=qa-1;
 /*  Generate service time*/
sum=0.;
for(i=1;i<=12;++i){
r=rand()/32768.;
sum=sum+r;        }
sta= meana+siga*(sum-6.);
sea=clock+sta;
konta=konta+1;
}
  else ita=ita+delt;
 }
  if(clock<sea) wta=wta+qa*delt;
  /* Check state of srever B*/
  if(clock>=seb)
  {
 wtb=wtb+qb*delt;
if(qb>=1) {
qb=qb-1;
  /*  Generate service time*/
sum=0.;
for(i=1;i<=12;++i){
r=rand()/32768.;
sum=sum+r;        }
stb= meanb+sigb*(sum-6.);
seb=clock+stb;
kontb=kontb+1;
   }
```

```
    else itb=itb+delt;
     }
    if(clock<seb) wtb=wtb+qb*delt;
    clock=clock+delt;

    }
   printf("\n clock=%6.2f count=%d  Kont A=%d  Kont B=%d",
        clock,count,konta,kontb);
   printf("\nIdle time of server A= %6.2f%",100.*ita/clock);
   printf("\nIdle time of server B= %6.2f%",100.*itb/clock);
   printf("\nAverage waiting time of components A= %6.2fper
component",
          wta/konta);
   printf("\nAverage waiting time of components B= %6.2f per
component",
          wtb/kontb);
   printf("\n Maximum number in buffer A=%d",qamax);
   printf("\n Maximum number in buffer B=%d",qbmax);
   }
```

An output of this program for a run of 1000 minutes is given below.

Clock = 1000 count = 214 Kont A = 198 Kont B = 16

Idle time of workstation A = 19.58%

Idle time of workstation B = 82.51%

Average waiting time of components A = 7.88 per component.

Average waiting time of components B = 0.69 per component.

Maximum number in buffer A = 6.

Maximum number in buffer B = 1.

The simulation run of 1000 may not be sufficient. As the simulation run is increased the values of statistics recorded vary and stabilize at about 50,000 minutes length of run. The variation of the results with length of simulation run is given below.

Length of run	*1000*	*2000*	*5000*	*10000*	*20000*	*50000*
Idle time of A (%)	19.58	23.81	28.06	27.77	26.44	26.69
Idle time of B (%)	82.51	79.12	81.02	81.41	80.11	79.59
AWTCA	7.88	6.03	5.68	5.58	6.40	6.46
AWTCB	0.69	1.44	1.66	1.39	1.30	1.42
Max. buffer A	6	6	6	7	9	11
Max. buffer B	1	1	2	2	2	3

Example 6.5. Dr. Strong is a dentist who schedules all her patients for 30 minutes appointments. Some of the patients take more or less than 30 minutes depending upon the type of dental work to be done. The following table shows the various categories of work, their probabilities and the time actually needed to complete the work.

Category	*Time required in minutes*	*Probability*
Filling	45	25
Crowning	60	15
Cleaning	15	25
Extraction	45	10
Checkup	15	25

Simulate the dentist's clinic for about four hours and determine the average waiting time for the patients and the percentage idle time for the dentist. Assume that all the patients show up at the clinic at exactly their scheduled arrival time starting at 8:00 AM. Use the following random numbers for simulating the process: 40, 82, 11, 34, 52, 66, 17 and 70. **[P.T.U. B.Tech. (Prod.), Dec. 2006]**

Solution: The dentist's clinic represents a single server queuing system, where the arrival times of the patients are known to be at 30 minutes interval. The service times are random and can be determined using the given probability distribution and the sequence of random numbers. The category of work, service time required, probability of occurrence of the category and the cumulative probability are given in table below. The last column of the table gives the range of random numbers corresponding to each category of work.

Category of work	*Time required*	*Probability*	*Comm. Prob.*	*Random numbers*
Filling	45	25	25	00 – 24
Crowning	60	15	40	25 – 39
Cleaning	15	25	65	40 – 64
Extraction	45	10	75	65 – 74
Checkup	15	25	100	75 – 99

Now using the given sequence of random numbers, the service time of patients can be generated. The first random number is 40, which lies in the range 40 to 64, which corresponds to cleaning, for which time required is 15 minutes. Thus the service time for 1st patient is 15 minutes. The service times for the 8 patients are given below:

Patient No.	1	2	3	4	5	6	7	8
Random No.	40	82	11	34	52	66	17	70
Service time	15	15	45	60	15	45	45	45

The simulation of Dr. Strong's clinic is carried out in Table 6.6. Since the patients are scheduled at 30 minutes interval, total 8 patients will arrive at the clinic in 4 hrs. The simulation table comprises of columns for patients number, arrival time, service begin time, service end time, dentist's idle time, and patient waiting time.

Patient number 1, arrives exactly at 8:00 AM, service begins immediately, service time for first patient being 15 minutes, service ends at 8:15. Dentist's idle time is zero, and patient waiting time is zero. Patient two arrives after 30 minutes at 8.30 AM. The service on this patient begins at 8.30, causing dentist to ideal for 15 minutes. Service time again is 15 minutes. Service ends at 8:45 AM. There is no waiting time from the second patient. Similarly, third patient arrive at 9:00, service begins at 9:00, dentist is idle for 15 minutes. Service on 3rd patent ends at 9:45. The fourth patient arrives at his scheduled time of 9:30 but service can being only at 9:45, when the doctor is free from the previous patient. Thus 4th patient has to wait for 15 minutes.

The service on 7th patient begins at 11:45 and ends at 12:30. Since the simulation is to be done for about 4 hrs, it can be stopped here. Thus in a simulation of 270 minutes, seven patients have been serviced. Idle time of the dentist is 30 minutes, or $\frac{30}{270} \times 100 \approx 11\%$. The waiting time of all the patients is 135 minutes, which amounts to an average of $\frac{135}{7} \approx 19$ minutes per patient.

Table 6.6

Patient number	*Arrival time*	*Service begin time*	*Service time*	*Service end time*	*Dentist's idle time*	*Patient waiting time*
1.	8:00	8:00	15	8:15	00	00
2.	8:30	8:30	15	8:45	15	00
3.	9:00	9:00	45	9:45	15	00
4.	9:30	9:45	60	10:45	00	15
5.	10:00	10:45	15	11:00	00	45
6.	10:30	11:00	45	11:45	00	30
7.	11:00	11:45	45	12:35	00	45
8.	11:30	12:30	45	–	–	–
				Total	30	135

Example 6.6. The distribution of inter-arrival times in a single server model is

t :	1	2	3
$f(t)$:	.25	.50	.25

and distribution of service times is

s :	1	2	3
$f(s)$:	.50	.25	.25

Complete the following table using the two-digit random numbers 11, 20, 47, 68, 90, 62 and 35 to generate arrivals and 15, 86, 20,42, 11, 36 and 48 to generate the corresponding service times.

Arrival No.	*Arrival time*	*Service begin time*	*Service end time*	*Waiting time in queue*	*Server idle time*

[P.U. M.E. (Mech.), 1987]

Solution: The arrival times of the customers can be determined by generating the inter-arrival times, from the given discrete distribution. The inter-arrival times, probabilities of their occurrence, cumulative probabilities and the corresponding two-digit random numbers are given in Table 6.7.

Table 6.7

Inter-arrival time	*Probability*	*Comm. probability*	*Range of random number*
1	.25	.25	00 – 24
2	.50	.75	25 – 74
3	.25	1.00	75 – 99

Now using the given string of random numbers, the inter-arrival times can be generated. Corresponding to first random number, *i.e.*, 11, the inter-arrival times is 1, *i.e.*, first customer will arrive 1 time unit after opening the system for service. The inter-arrival times for seven arrivals are computed below.

Arrival number	:	1	2	3	4	5	6	7
Random number	:	11	20	47	68	90	62	35
Inter-arrival time	:	1	1	2	2	3	2	2

Similarly, the service times are generated, using the sequence of random numbers provided for this purpose.

Sr. No.	:	1	2	3	4	5	6	7
Random number	:	15	86	20	52	11	66	48
Service time	:	1	3	1	2	1	2	1

The given table can now be computed. It is assumed that service system opens at zero time.

Table 6.8

Arrival No.	*Arrival time*	*Service begin time*	*Service end time*	*Waiting time*	*Server idle time*
1.	0 + 1 = 1	1	1 + 1 = 2	0	1
2.	1 + 1 = 2	2	2 + 3 = 5	0	0
3.	2 + 2 = 4	5	5 + 1 = 6	1	0
4.	4 + 2 = 6	6	6 + 2 = 8	0	0
5.	6 + 3 = 9	9	9 + 1 = 10	0	1
6.	9 + 2 = 11	11	11 + 2 = 13	0	1
7.	11 + 2 = 13	13	13 + 1 = 14	0	0

Example 6.7. Simulate an M/D/2 system over the first 35 minutes of operation taking mean inter-arrival time as 3 minutes and the service times of Servers I and II as 5 and 6 minutes each. The inter-arrival times for the first 12 arrivals in min : sec have been generated as:

4:13, 2:00, 6:09, 1:37, 3:54, 6:09, 0:05, 2:49, 1:26, 0:52, 3:39, 8:54

Determine the percentage idle time of each server. **[P.U. M.E. (Prod.), 1991]**

Solution: The notation M/D/2 stands for the queuing system having exponential arrival distribution, deterministic service times, two servers, infinite system capacity and the queue discipline is first-in first-out. In present example, mean inter-arrival time is 3 min. Using exponential distribution, the inter-arrival times have already been generated. The two servers; Server I and Server II take exactly 5 and 6 minutes respectively to serve the customer. There is no limit on the number of customers in the system.

Working of the system is simulated in Table 6.9. It is assumed that at zero time, there is no customer in the system. The first arrival takes place at 4:13. When both the servers are idle, the choice of the customer is faster server, *i.e.*, Server I. Service begins at 4:13 and ends at 9:13 as Server I takes exactly 5 minutes. Second customer arrives after 2:00 minutes i.e., at 6:13 and goes to Server II, where service beings at 6:13 and ends at 12:13. Third arrival takes place at 12:22 minutes, both the servers are idle, customer goes to Server I. Here the idle time of Server I is 12:22 – 9:13 = 3:09 mins. Fourth arrival takes place at 13:59, goes to idle Server II. The idle time of Server II is 13:59 – 12:13 = 1:46 mins. The process continues, when customer No. 8 arrives at 26:56 minute both the servers are busy, customer waits till Server I completes pervious service at 29:02 minutes. Here waiting time of customer is 2:06 mins. During 35 minutes, service on 8 customer is completed, 9th and 10th customer are in service.

SIMULATION OF INVENTORY SYSTEMS

7.1 Introduction

Inventory management is one of the important areas of operations research and management science, which have been modelled, analyzed and simulated extensively. Though simple problems are amenable to the analytical techniques, but the real life inventory problems are generally so complex that Simulation is the only tool, which provides reasonably accurate solutions.

There is a wide range of situations where one or the other type of inventory is required for the smooth functioning of the system. The need of inventory arises out of difference in the timing or location of demand and supply. This applies whether one is dealing with the raw materials for a production process or finished goods stocked by a manufacturer, wholesaler, or retailer of any sort of goods. From a customer's perspective inventory should be such that the demand is always fulfilled. This is generally true in case of many grocery items, but rarely true for items like cars and special purpose machine tools. Inventories represent idle capital and thus cost money. In simple form inventory management, calls for maintaining adequate supply of specified items so as to meet the expected pattern of their demand and at the same time keeping the costs involved in maintaining the inventory and that of shortages at the minimum.

7.2 Classification of Inventory Systems

The inventory systems are generally classified into three types:

- Repetitive order-independent demand
- Single order-independent demand
- Repetitive order-dependent demand.

The first type is the most common situation faced in business. The second type, single-order systems like orders for Christmas trees and Diwali gifts occur but not very frequently. Systems with dependent demand are generally witnessed in manufacturing, where the parts procured from outside are used in the assembly of the final product. The demand for parts from outside vendors is dependent on the demand for finished items. The analysis of such situations is called the Material Requirements Planning (MRP).

Another way of classification of inventory, which is more relevant to the manufacturing systems, is according to their relation to the overall sequence of production operations.

- Supplies.
- Materials.
- In-process goods.
- Finished goods.

Supplies constitute of such staple items like office stationery, computer paper, floppy disks, compact discs, programming tapes, drawing office materials and various forms used in production planning. Such items generally have independent demand. Materials constitute all those items, which are worked upon or consumed in the manufacturing process. The raw materials, oils, lubricants, fuels, chemicals and paints etc. are materials. In-process goods are incomplete products held between various workstations, while the finished goods are the end product of the production system.

A large number of inventory models are available to deal with a wide variety of inventory situations. These can though be classified in a number of ways, but commonly these are grouped into two broad categories.

- Fixed-order quantity models
- Fixed-order period models

In a fixed-order quantity model the demand is met out of the inventory held in stock and the stock position is continuously (or perpetually) updated. If a particular demand is not fulfilled or is partially met, either a back order is taken or the sale is lost. As soon as the stock level touches the reorder point, order for fixed quantity is placed. In case of fixed-order period model, there is no perpetual updating of the stock; rather periodic reviews are made at fixed intervals of time. The amount balance in stock plus amount on order minus the backlog is compared with the desired maximum level and the order is placed for the difference. Thus in fixed-order quantity model the order quantity is fixed and the period between orders is variable, while reverse is true for the fixed order period model.

7.3 Inventory Costs

The costs involved in inventory systems are classified into three types.

1. ***Reorder cost*** or ***Set up cost***: It is the administrative cost associated with the placing of the order.
2. ***Inventory carrying cost*** or ***inventory holding cost***: It is the cost of maintaining the inventory, which includes storage charges, insurance, interest on tied capital, etc.
3. ***Shortage cost***: This cost is incurred when the system runs out of stock. This results in loss of profits as well as loss of goodwill and reputation.

Price of an Item: The purchase cost or price of an item includes the basic price of an item plus the taxes and transportation costs if any. In case of items produced by the firm itself, the full production cost is called the price.

Lead-Time: This is the time lag between the placing of order and obtaining the delivery of items. It is also called ***delivery lag***.

Service level: The service level for an inventory system is a measure of the customer satisfaction and may be defined as the ratio of units furnished to customers to the total number of units demanded by the customers. If the shortages are more the service level will naturally be low. No management will like to have an inventory policy with poor service level, even if it is the most economical operating policy. The service level is generally expressed as a percentage.

7.4 Single Item Constant Demand Inventory Model

Let us consider a very simple inventory system to have a feel of the basic inventory theory. Fig. 7.1 shows the variation in stock over time for a single item with constant demand. To start with the inventory level is Q, which goes down at a constant rate to zero, where it again jumps back to level Q. The cycle goes on repeating with average inventory level as Q/2 items.

In this model as soon as the inventory becomes zero it is replenished without any lag of time and the system never runs out of stock. Thus there is no shortage cost.

Let k be the carrying cost per item per day,

R be the order cost per order,

D be the volume of daily sale,

Then total inventory cost per day is

$$C = (Q/2)\cdot k + (D/Q)\cdot R$$

Q/D is the number of days between two consecutive deliveries.

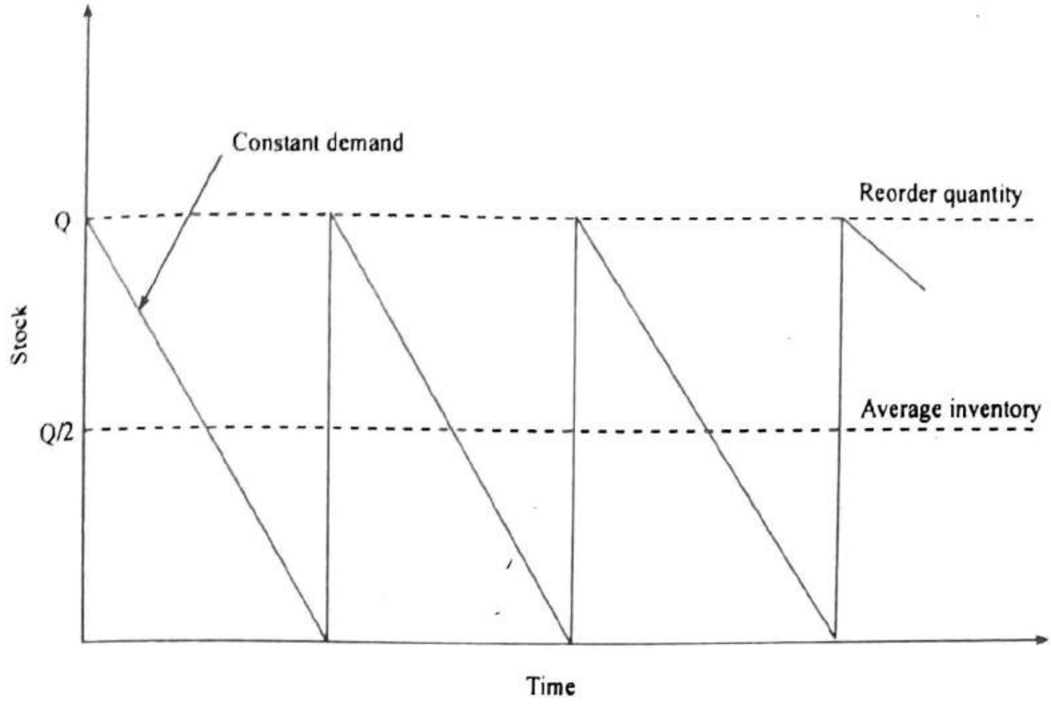


Fig. 7.1

On differentiating C with respect to Q and equating dC/dQ to zero, we get

$$\frac{dc}{dQ} = \frac{k}{2} - \frac{DR}{Q^2} = 0$$

$$Q^* = \sqrt{\frac{2DR}{k}}$$

The value of Q so obtained is also referred to as the Economic Order Quantity (EOQ).

Thus $$\text{EOQ} = \sqrt{\frac{2DR}{k}}$$

The EOQ gives the optimal reorder quantity, which minimizes the inventory cost C. Thus an order of Q* items should be placed after fixed regular intervals of Q*/D days.

This lot size formula is known as Wilson's Formula (named after R.H. Wilson) and has been extensively used in inventory theory.

To illustrate the use of this formula let us consider an inventory problem where;

k, the carrying cost per item per day is Re. 0.01,

D, the volume of daily sales is 100 units, and

R, the reorder cost per order is Rs. 200.

$$\text{EOQ} = \sqrt{\frac{2DR}{k}} = 2000 \text{ units}$$

$$Q/D = 2000/100 = 20$$

$$C = (2000/2)\cdot 0.01 + (100 \times 200)/2000 = \text{Rs. } 20 \text{ per day.}$$

Thus, 2000 units should be ordered after every 20 days for a minimum inventory cost of Rs. 20.00 per day.

7.5 EOQ with Constant Lead-Time

In the above model it was assumed that the stock was replenished immediately when it touched zero level, which is not practically feasible. There is always some *delivery lag*. If this delivery lag is constant say d days, it does not make any difference to the EOQ model. The order is placed exactly *d* days before the items desired arrival day. These reorder days are shown as reorder points in Fig. 7.2. However in case the delivery lag is not constant and is a random variable the analysis becomes complicated and the situation can best be handled by simulation.

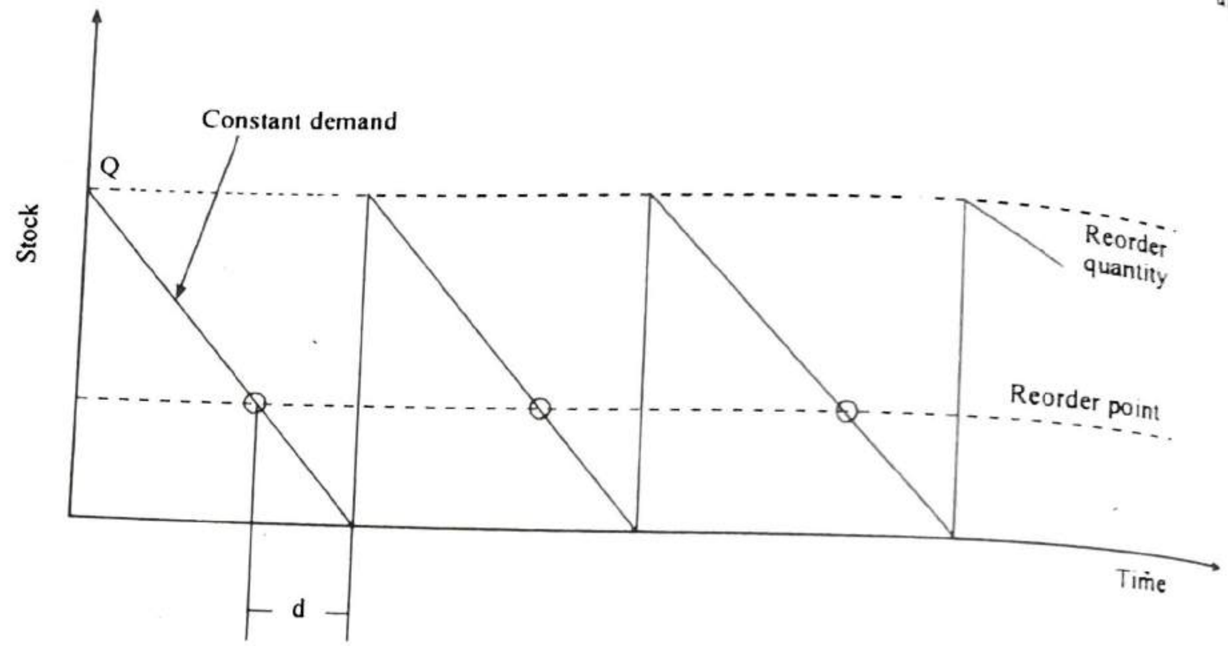


Fig. 7.2

7.6 EOQ with Shortage

The Wilson's EOQ Formula derived above assumed that the stock was replenished as soon as the stock touched zero level and that there was never any shortage that is system works at 100% service level. However in practice the 100% service level sometimes becomes uneconomical, as comparatively large inventory has to be maintained.

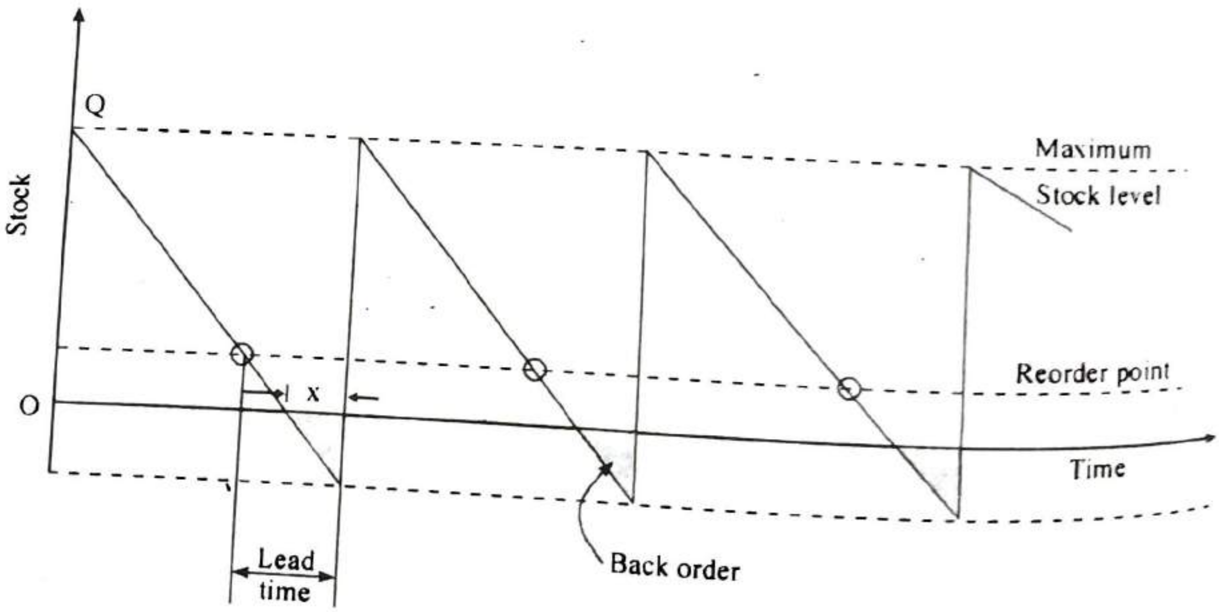


Fig. 7.3

Two types of situation occur when the system runs out of stock. First, the customer goes to some other supplier and the sale is permanently lost. Second, the customer leaves his order and the same is replenished as soon as the stock is received. This is called back order or backlog. In the case of back orders, though the sale is not lost but some penalty cost per item per unit time may occur. Penalty may occur in the form of cost of maintaining back orders, cost of shipping the stock to the customer, rebate in price and loss in goodwill, etc.

Let us modify the Wilson's model for the back ordering situation. Fig. 7.3 shows a system with constant daily demand D, a reorder quantity Q, a constant lead-time d. For a fraction of time x the system remains out of stock, while for the remaining time $(1 - x)$ the demand is met.The backlog is maintained which is filled as soon as the next delivery is received.

$$EOQ = \sqrt{\frac{2DR}{k}} \cdot \sqrt{\frac{k+b}{b}} \qquad \text{where } b = \frac{kx}{1-x}$$

7.7 Why Simulation of Inventory Systems?

In the previous section very simple inventory situations have been analyzed and simple inventory models derived. The real situations are much more complex than these situations. In a production environment we come across different types of inventories as raw materials, finished products and in-process inventories. Many complicated formulas are available in mathematical inventory theory to deal with problems of various types. However, no analytical formula is sufficiently versatile to solve the reasonably complex real life inventory problems. The real life inventory problems are generally so complex that simulation is the only tool, which can be employed to find the optimal operating policies. The following are some of the factors, which add to the complexities of inventory problems.

7.7.1 *Gradual Replenishment*

In deriving Wilson's Formula with or without lead-time, we assumed that the stock was replenished instantaneously as soon as the delivery was received. However this is not true in many situations especially in a production environment where the replenishment occur gradually. The demand is met from stock as well as from production. The particular item may be produced for some period and the same production facilities may then be used for the production of some other item. If the rate of production during the production period is P units a day and the demand is D units a day then during that period the inventory will be replenished at the rate of $(P - D)$ units a day. For such a situation following formula for the EOQ can be derived.

$$EOQ = \sqrt{\frac{2DR}{k\,(1 - D/P)}}$$

7.7.2 *Multi-item Production*

We derived the above formula, taking only one item into consideration. Generally a manufacturer uses the same production facility for the production of several different items one after the other. In such situations a batch (quantity of items produced) of one product should be sufficiently large so as to last for the duration when other products are being produced. Thus a production cycle is established such that one batch of each product is produced in one cycle. It can be shown that under such a situation the optimal number of production cycles per unit of time is given by

$$\sqrt{\frac{\Sigma D_j k_j (1 - D_j / P_j)}{2\Sigma r_j}}$$

where D_j = demand for the jth product per unit time,

k_j = carrying cost for the jth product per unit time,

r_j = set up cost for the jth product, and

P_j = production rate for the jth product.

7.7.3 *Capital Restriction*

In the previous sections it has been assumed that there was no restriction as for as the capital was concerned and as many units as determined by the EOQ formula could be procured. However, in many situations there are restrictions on capital available and limits are imposed, that the value of inventory should not exceed a specified amount.

7.7.4 *Quantity Discounts*

In many situations discounts on price are offered on making large purchases. The price reduction may occur at discrete points or it may be continuous. The rate of discount may also vary with the size of the order. The advantages of quantity discount are the direct savings due to reduced price; and saving in the interest on capital tied in inventory, which reduces the per unit carrying cost. On the other hand, to take this advantage large average stock has to be maintained, which results in increased carrying cost. Mathematical models have been developed to take into consideration the quantity discounts or price breaks but their application are limited only to very simple price structures.

7.7.5 *Varying Demand*

In deriving the Wilson's Formula it was assumed that the demand of the items was constant which is rarely true in real life situations. Generally the demand is a stochastic variable. The pattern of demand has to be determined by using the historical data and the forecasting techniques, which may give some discrete or continuous probability distribution. The effect of seasonal variation in demand has also to be superimposed. In addition to the number of arrivals the number of items demanded by each arrival also vary randomly. If the customers arrive independently, their inter arrival times generally follow exponential distribution and if each customer demands only one unit then the total demand in any period will be Poisson distributed. However, if the number of units demanded by the customers vary randomly then the probability density function of demand will depend upon the probability distribution of the number of units demanded.

7.7.6 *Varying Lead-time*

The lead-time, which we assumed to be a constant in the derivation of Wilson's Formula, is generally a random variable. The frequency distribution of lead-time has also to be established using the past data and the forecasting techniques. Lead-time generally follows Erlang distribution that is gamma distribution with integral value of parameter k, which becomes exponential for $k = 1$ and normal when k becomes large.

7.7.7 *Multiple Orders Outstanding*

In the derivation of various EOQ Formulae, it was assumed that at any time only one reorder could be outstanding. But in many inventory situations two or more than two reorders could be pending at a time. If, in addition to that, the lead-times are stochastic, the analysis of the situation becomes almost impossible for any analytical technique. In such a situation it is not essential that orders be delivered in the sequence in which they were placed. The reorder placed later can arrive earlier. And if the reorders must always arrive in the same sequence in which they were placed, then their lead times are not random variables.

The various factors discussed above make the inventory problems so complicated that it becomes impossible to find a reasonably good solution by employing the analytical techniques. The mathematical techniques at the most help to obtain the approximate guidelines. Since, inventory occupies an important place in any business or industrial operation, and has a significant bearing on the profits, its analysis and optimization is of utmost importance to any management. As already emphasized, simulation is the only tool to determine the optimal operating policy for a complex situation.

7.8 Example — Simulation of an Inventory Problem

In an inventory system the demand as well as the lead-times are random variables defined by discrete distributions as given below.

Demand	:	3	4	5	6
Probability	:	0.15	0.30	0.35	0.20
Lead-time	:	2	3	4	
Probability	:	0.2	0.6	0.2	

Two reorders each of quantity (Q) of 15 units can be outstanding at a time. There are two reorder points RP1 and RP2 at levels of 10 and 5 units respectively. The shortages are lost forever. The reorder cost is insignificant as compared to the carrying cost, and the objective is to determine the service level and the average stock held for the given reorder points and the reorder quantity. Taking an initial stock of 10 units, let us manually simulate the system for the first 20 days. This manual simulation will be helpful in understanding the mechanism of simulation and will make the computer programming of the model simple.

Though in computer simulation the demand and lead-times will be generated as and when required, here let us generate the demand for all the 20 days and some lead-times before starting the simulation process.

Generation of demand:

Demand	*Probability*	*Commu. Prob*
3	0.15	0.15
4	0.30	0.45
5	0.35	0.80
6	0.20	1.00

Day	:	1,	2,	3,	4,	5,	6,	7,	8,	9,	10,	11,	12,	13,	14,	15,	16,	17,	18,	19,	20
Random No.	:	64,	33,	18,	87,	74,	41,	53,	09,	67,	93,	56,	35,	48,	70,	32,	11,	87,	12,	23,	08
Demand	:	5,	4,	4,	6,	5,	4,	5,	3,	5,	6,	5,	4,	5,	5,	4,	4,	6,	3,	4,	3

Similarly the lead-times are generated. First 10 lead-times are,

3, 2, 4, 3, 3, 3, 2, 4, 3, 3

The simulation process is shown in Table 7.1. On day 1, the stock is equal to the initial stock of 10 units. Since the reorder point is at 10, on the very first day the reorder is placed for 15 units. The lead-time for this order is 3 days hence the delivery will take place on $1 + 3 = 4$th day in the morning. The reorder (ord 1) is assigned 15 and due date (ddate 1) is marked 4. Since the stock is more than the second reorder point hence ord 2 and due date (ddate 2) are marked zeros. Demand on day 1 is 5 units, which is completely met, and hence lost sale (short) on day 1 is zero. On day 2, stock available is 5 units, ord 1 is 15, and ddate 1 is 4. Now since the stock has touched the second reorder point, the second reorder is also placed with a lead-time of 2 days. With this ord 2 becomes 15 and the ddate 2 becomes $2 + 2 = 4$. Thus both the reorders will arrive at the same time, on the morning of 4th day. On day 2, demand is 4 units, which is completely met. Lost sale on day 2 again is zero. On day 3 stock available is only one unit while demand is 4 units. There is a shortage of 3 units, which is the lost sale on day 3. On day 4, both the reorders have been delivered and the stock becomes 30 units. Since stock is more than the reorder points hence, ord 1, ddate 1, ord 2 and ddate 2 are marked zeros. This way the simulation continues.

After 20 days,

Average stock = 11.9 units

Service level = 81%.

Table 7.1 : SIMULATION OF INVENTORY PROBLEM

day	*stk*	*cstk*	*ord 1*	*ddate 1*	*ord 2*	*ddate 2*	*dmd*	*cdmd*	*short*	*cshort*
1	10	10	15	**1 + 3 = 4**	0	0	5	5	0	0
2	5	15	15	4	15	**2 + 2 = 4**	4	9	0	0
3	1	16	15	4	15	4	4	13	3	3
4	30	46	0	0	0	0	6	19	0	3
5	24	70	0	0	0	0	5	24	0	3
6	19	89	0	0	0	0	4	28	0	3
7	15	104	0	0	0	0	5	33	0	3
8	10	114	15	**8 + 4 = 12**	0	0	3	36	0	3
9	7	121	15	12	0	0	5	41	0	3
10	2	123	15	12	15	**10 + 3 = 13**	6	47	4	7
11	0	123	15	12	15	13	5	52	5	12
12	15	138	0	0	15	13	4	56	0	12
13	26	164	0	0	0	0	5	61	0	12
14	21	185	0	0	0	0	5	66	0	12
15	16	201	0	0	0	0	4	70	0	12
16	12	213	0	0	0	0	4	74	0	12
17	8	221	15	**17 + 3 = 20**	0	0	6	80	0	12
18	2	223	15	20	15	**18 + 3 = 21**	3	83	1	13
19	0	223	15	20	15	20	4	87	4	17
20	15	238	0	0	15	20	3	90	0	17

Average Stock = 238/20 = 11.9 units

Service level = (90 – 17)/90 = 81%.

Computer Programme

In the simulation Table 7.1 the time progressed from day-to-day that is we used the fixed increment time flow mechanism. While developing a computer programme it is always helpful to draw the flow chart of the programme logic as given in Fig. 7.4. The variables used in the program have been defined as;

q1	Quantity of reorder 1
q2	Quantity of reorder 2
dmd	Demand
cdmd	Cumulative demand
stk	Stock held
cstk	Cumulative stock
avstk	Average stock
ord 1	State of Reorder 1
ord 2	State of Reorder 2
dd1	Due date of delivery of reorder 1
dd2	Due date of delivery of reorder 2
short	Shortage of demand.
cshort	Cumulative shortage of demand
todays	Length of simulation run in days

lt1	Lead-time for reorder 1
lt2	Lead-time for reorder 2
rp1	Reorder point 1
rp2	Reorder point 2
sl	Service level

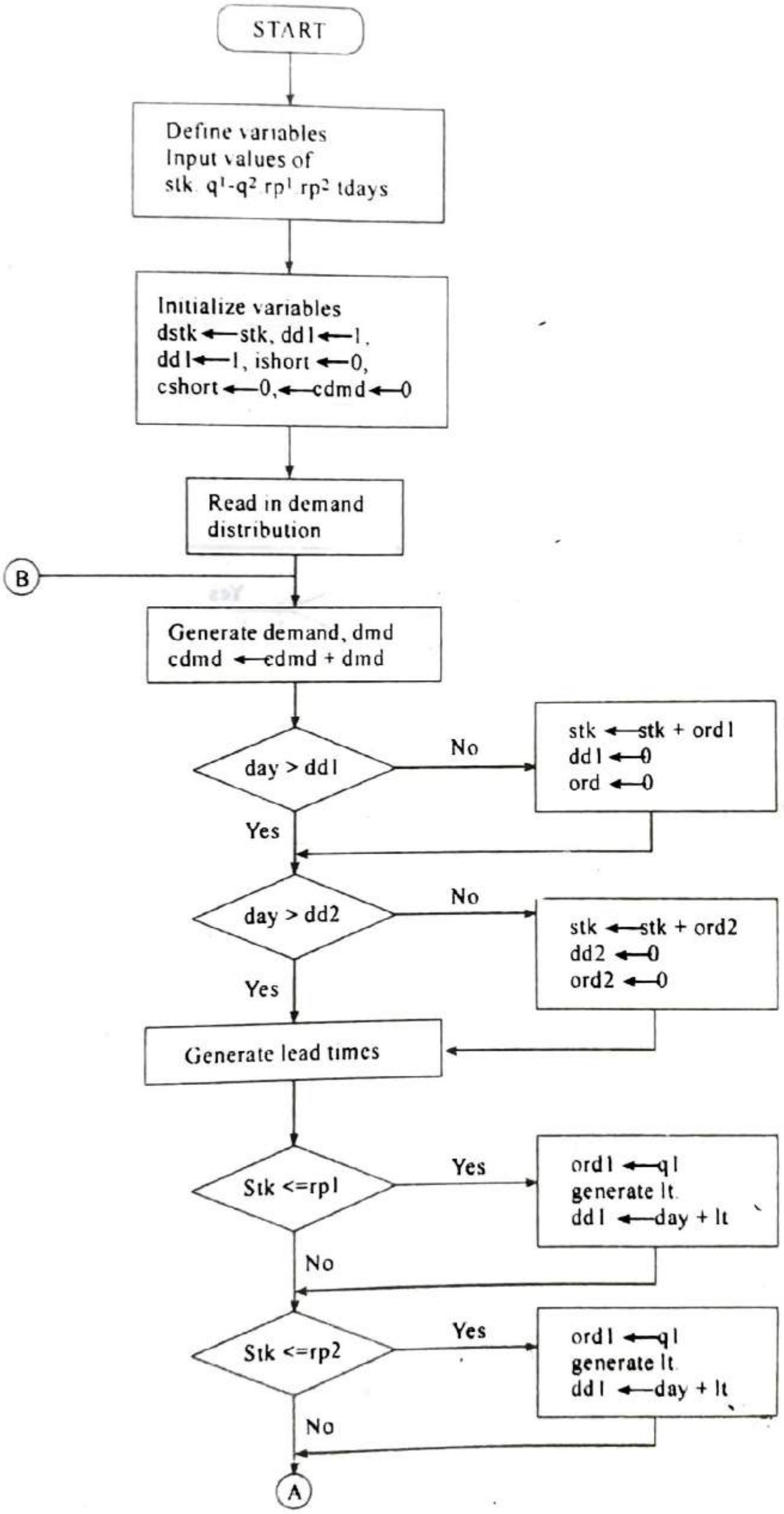


Fig. 7.4 (*Cont'd*)

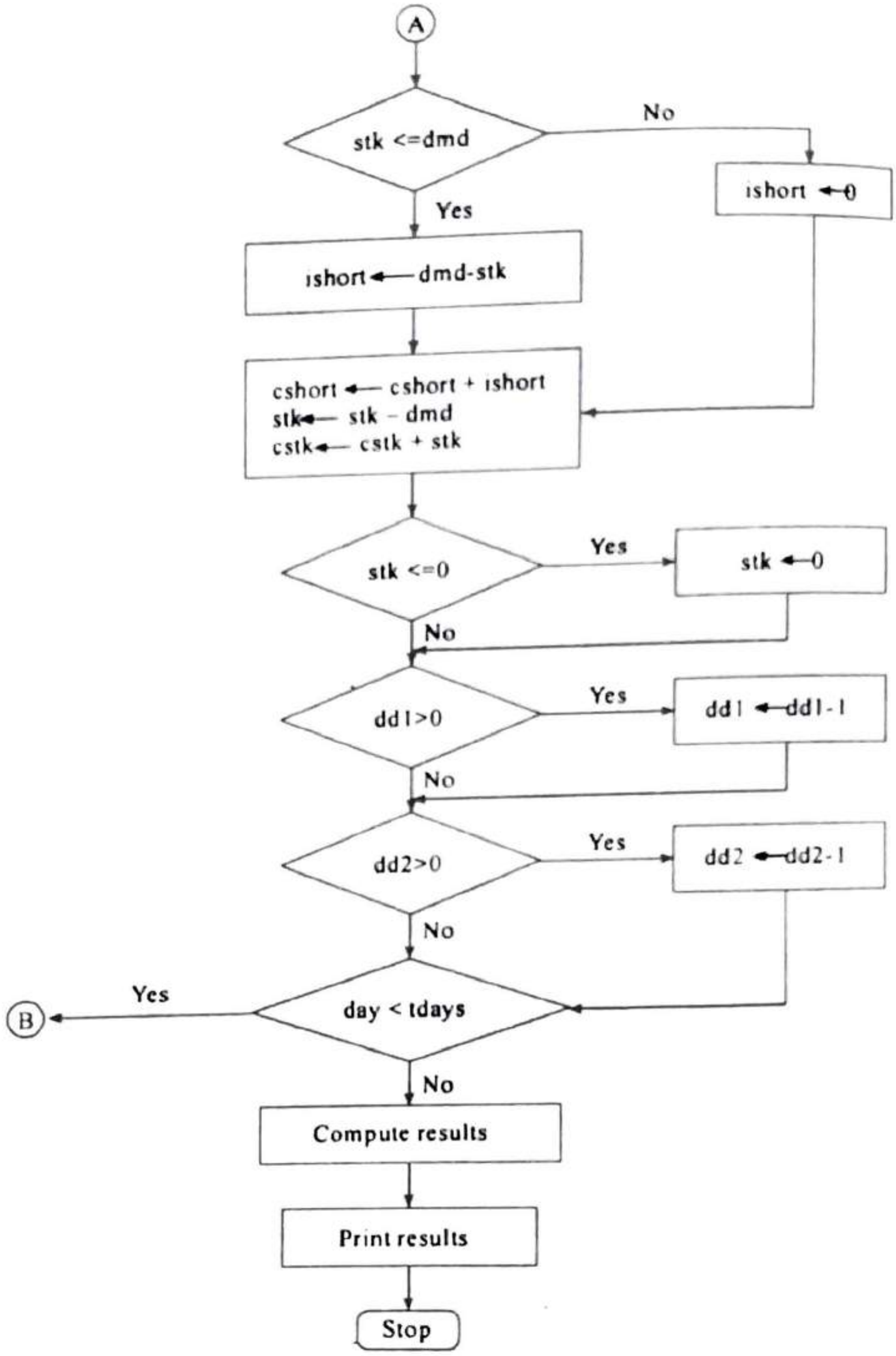


Fig 7.4

A computer programme for this simulation written in C language is given below.

```
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
main()
    {
    /* INVENTORY PROBLEM with two orders*/
    int q1,q2,rp1,rp2,tdays,dmd,cdmd,stk,cstk,day,dd1,dd2,lt1,lt2;
    int ishort, cshort,ord1,ord2;
    float x,avstk,sl;
```

```
q1=10;q2=10;
day=1;dd1=0;dd2=0;tdays=1000;
rp1=2; rp2=2;ord1=0;ord2=0;
ishort=0;cshort=0;cdmd=0;
printf("\n Enter valueof initial stk=");
scanf("%d",&stk);
printf("\n Initial stock=%d",stk);
cstk=stk;
/* printf("\n   Day stk cstk ord1 dd1 ord2  dd2 dmd cdmd ishort
   cshort\n");*/

   /*    printf("initial stock=%d q1=%d  q2=%d",stk,q1,q2);*/
for(day=1;day<=tdays;++day)
      {
   /* Generate Demand */
   x=rand()/32768.0;
   if(x<=0.15) dmd=4;
   else if((x>0.15) && (x<=0.45)) dmd=5;
   else if((x>0.45) && (x<=0.80)) dmd=6;
   else dmd=7;
   cdmd=cdmd+dmd;
   /*    printf("\n
%2d %4d %4d %3d %2d %3d   %2d   %3d   %4d  %3d  %4d",
   day,stk,cstk,ord1,dd1,ord2,dd2,dmd,cdmd,ishort,cshort); */
   if(day>=dd1){
    stk=stk+ord1;
    dd1=0;
    ord1=0; }
   if(day>=dd2){
   stk=stk+ord2;
   dd2=0;
   ord2=0; }
   /*generate lead time*/
   if( stk<=rp1) {
   ord1=q1;
   x=rand()/32768.0;
   if(x<=0.2) lt1=2;
   else if((x>0.2)&&(x<=0.8)) lt1=3;
   else  lt1=4;
   dd1=day+lt1;
   }
   /*    printf("\ndd1=%d lt1=%d",dd1,lt1); */

   if(stk<=rp2) {
   ord2=q2;
   x=rand()/32768.0;
```

```
if(x<=0.2) lt2=2;
else if((x>0.2)&&(x<=0.8)) lt2=3;
else  lt2=4;
dd2=day+lt2;
}

if(dmd<=stk) ishort=0;
else ishort=dmd-stk;
cshort=cshort+ishort;
stk=stk-dmd;
cstk=cstk+stk;
if(stk<=0) stk=0;
if(dd1>0) dd1=dd1-1;
if(dd2>0) dd2=dd2-1;
/*printf("\nday= %d cstk=%d cshort%d",day,cstk,cshort); */

}
float days=tdays;
avstk=cstk/days;
sl=(cdmd-cshort)*100.0/cdmd;
printf("\n Simulation run=%d days",tdays);

printf("\n Q1=%d Q2=%d  RP1=%d  RP2=%d",q1,q2,rp1,rp2);

printf("\n Average stock=%7.2f Service level=%5.2f%",avstk,sl);

}
```

The average stock and service levels obtained for some combinations of q1, q2, rp1 and rp2 are given below.

Length of Run = 100 days

Q1	*Q2*	*RP1*	*RP2*	*Initial Stk = 10*		*Initial Stk = 5*	
				Av.Stk	*SL*	*Av.Stk*	*SL*
15	15	10	7	5.78	74.65	5.93	73.61
15	10	10	7	4.16	72.81	3.17	68.15
10	15	10	7	3.99	68.90	4.90	73.66
10	10	10	7	2.34	64.53	2.32	66.26
15	10	7	7	1.47	66.28	1.48	66.40
10	10	5	5	1.04	62.70	0.93	62.03
10	10	2	2	0.38	56.76	0.37	56.68

7.9 Example – Single Order Outstanding

The simulation of an inventory problem with a single order outstanding is given in Table 7.2. In this case lead-time is 4, while the reorder point and reorder quantity are 8 and 15 respectively. The manual simulation carried out for 20 days gives the service rate as 62.5% and the average stock as 5.95 units.

The same inventory situation has been simulated again in Table 7.3 by reducing the lead-time to 3 days. By reduction in lead-time, the performance of the system improves, to service level of 69.32%, but this is at increased average stock of 6.45 units.

Similarly in Table 7.4 and Table 7.5 we find that service level can be increased by increasing the reorder quantity, but this also results in increased average stock held.

Table 7.2

Initial stock = 10 Lead-time = 2

Demand 3.4.5.6 p (.20,.30,.35,.15)

Reorder point = 8

Reorder quantity = 15

Single order outstanding.

Day	Stock	ordr.due	date.due	demand	shortage
1	10	0	0	6	0
2	4	15	4	3	0
3	1	15	3	4	3
4	0	15	2	3	3
5	0	15	1	4	4
6	15	0	0	3	0
7	12	0	0	5	0
8	7	15	4	4	0
9	3	15	3	5	2
10	0	15	2	6	6
11	0	15	1	5	5
12	15	0	0	4	0
13	11	0	0	3	0
14	8	15	4	5	0
15	3	15	3	5	2
16	0	15	2	4	4
17	0	15	1	3	3
18	15	0	0	5	0
19	10	0	0	5	0
20	5	15	4	6	1
	119			88	33

Service level = (88 − 33)/88 = 62.5%.

Average stock = 119/20 = 5.95

Table 7.3

Initial stock = 10 Lead-time = 3

Demand 3.4.5.6 p (.20, .30, .35, .20)

Reorder point = 8

Reorder quantity = 15

Single order outstanding.

Day	Stock	ordr.due	date.due	demand	shortage
1	10	–	–	6	–
2	4	15	3	3	–
3	1	15	2	4	3
4	0	15	1	3	3
5	15	–	–	4	–
6	11	–	–	3	–
7	8	15	3	5	–
8	3	15	2	4	1
9	0	15	1	5	5
10	15	–	–	6	–
11	9	–	–	5	–
12	4	15	3	4	–
13	0	15	2	3	3
14	0	15	1	5	5
15	15	–	–	5	–
16	10	–	–	4	–
17	6	15	3	3	–
18	3	15	2	5	2
19	0	15	1	5	5
20	15	–	–	6	0
	129			88	27

Service level = (88 – 27)/88 = 69.32%.

Average stock = 129/20 = 6.45

Table 7.4

Initial stock = 10 Lead-time = 3

Demand 3.4.5.6 p (.20, .30, .35, .15)

Reorder point = 8

Reorder quantity = 20

Single order outstanding.

Day	Stock	ordr.due	date.due	demand	shortage
1	10	0	0	6	0
2	4	20	3	3	–
3	1	20	2	4	3
4	0	20	1	3	3
5	20	0	0	4	0
6	16	0	0	3	0
7	13	0	0	5	0
8	8	20	3	4	0
9	4	20	2	5	1
10	0	20	1	6	6
11	20	0	0	5	0
12	15	0	0	4	0
13	11	0	0	3	0
14	8	20	3	5	0
15	3	20	2	5	2
16	0	20	1	4	4
17	20	0	0	3	0
18	17	0	0	5	0
19	12	0	0	5	0
20	7	20	3	6	0
	189			88	19

Service level = (88 – 19)/88 = 78.4%.

Average stock = 189/20 = 9.45 units.

Table 7.5

Initial stock = 10 Lead-time = 3

Demand 3.4.5.6 p (.20, .30, .35, .15)

Reorder point = 15

Reorder quantity = 20

Single order outstanding.

Day	Stock	ordr.due	date.due	demand	shortage
1	10	20	3	6	0
2	4	20	2	3	0
3	1	20	1	4	3
4	20	0	0	3	0
5	17	0	0	4	0
6	13	20	3	3	0
7	10	20	2	5	0
8	5	20	1	4	0
9	21	0	0	5	0
10	16	0	0	6	0
11	10	20	3	5	0
12	5	20	2	4	0
13	1	20	1	3	2
14	20	0	0	5	0
15	15	20	3	5	0
16	10	20	2	4	0
17	6	20	1	3	0
18	23	0	0	5	0
19	18	0	0	5	0
20	13	20	3	6	0
	248			88	5

Service level = (88 – 5)/88 = 94.32%.

Average stock = 248/20 = 12.4

A computer program for the above inventory system, having fixed order quantity, fixed reorder point, fixed lead-time and random demand with empirical distribution, is given below.

```
#include<stdio.h>
#include<stdlib.h>
#include<math.h>

main()

   /*Simulation of Inventory Problem- fixed reorder point, fixed Order
     quantity, fixed lead time and variable demand:3,4,5,6 having
     p=(.20,.30,.35,.15).
     Terms used:
     stk - stock held
     cstk - cumulative stock held
     ordd - order due
     ddue - date on which order due
     dmd - demand
     cdmd - cumulative demand
     shortage - unfulfilled demand
     cshortage - cumulative shortage
     rp - reorder point
     avstk - average stock
     sl - service level
     tdays - total days that is length of simulation run.     */

  {
   int day,stk,cstk,ordd,ddue,dmd,cdmd,shortage,cshort,rp,q,lt;
   float x,y,avstk,sl;
   int tdays;

   stk=10;lt=3,rp=10;q=15;day=1;cstk=0;
   ordd=0;ddue=0;shortage=0;dmd=0;cdmd=0;cshort=0;

   printf("\n Length of simulatión run (tdays)=");
   scanf("%d",&tdays);
   printf("\n Initial stock=%d",stk);
   printf("\n Lead time =%d",lt);
   printf("\n Reorder point=%d",rp);
   printf("\n reorder quantity=%d",q);
     /*printf("\n Day  Stock  Cstk   Ordd   Ddue  Demand  Cdmd Shortage
Cshort");*/
   for(day=1;day<=tdays;++day) {
     /*printf("\n %2d %5d %6d %6d %6d %6d %6d %6d
%6d",day,stk,cstk,ordd,
     ddue,dmd,cdmd,shortage,cshort);*/
   stk=stk-dmd;
```



```
   if(stk<=0) stk=0;
   if(ddue==0){
   stk=stk+ordd; ordd=0; }

   cstk=cstk+stk;
   if(ordd==0)   {
   if(stk<=rp){ ordd=q; ddue=lt;}
             }
   /*Generate demand*/
   x=rand()/32768.0;
   if(x<=0.20) dmd=3;
   else if((x>0.2)&&(x<=0.50)) dmd=4;
   else if((x>0.5)&&(x<=0.85)) dmd=5;
   else dmd=6;
   cdmd=cdmd+dmd;
   if(dmd>stk) shortage=dmd stk;
   else shortage=0;
   cshort=cshort+shortage;
     /*printf("\n %2d %5d %6d %6d %6d %6d %6d %5d
%6d",day,stk,cstk,ordd,
     ddue,dmd,cdmd,shortage,cshort);*/
   if(ddue>0)  ddue=ddue-1;
   }
   sl=(cdmd-cshort)*100.0/cdmd;
   float days=tdays;
   avstk=cstk/days;
   printf("\nAverage stock= %6.2f units  Service Level=%6.2f
%",avstk,sl);
  }
```

The program written in C language can be employed to experiment with various combinations of lead-time, reorder quantity and reorder point to study their influence on the system performance. To obtain reasonably reliable results, length of simulation run can be increased. The determination of simulation run and other issues related to the design of simulation experiment are discussed in Chapter 9.

Some results obtained from the above program are given below in Table 7.6.

Table 7.6

Lead-time	*Reorder point = 8*		*Reorder point =10*	
	Ave. stock	*SL(%)*	*Ave. stock*	*SL(%)*
1	11.70	98.97	13.58	99.96
2	8.23	82.22	9.48	91.42
3	6.66	66.80	7.48	74.84
4	5.54	55.67	6.07	60.77
5	4.77	47.86	5.16	51.88

Example 7.1. A bakery keeps stock of a popular brand of cake. Daily demand based on past experience is given below :

Daily Demand	0	15	25	35	45	50
Probability	.01	.15	.20	.50	.12	.02

Using the following sequence of random numbers, simulate the demand for the next 10 days.

48, 78, 09, 87, 99, 77, 15, 14, 68 and 89

Find out the stock situation, if the owner of the bakery decides to make 35 cakes every day. The unmet demand on any day is lost. [P.T.U., B.Tech. (Prod.), Dec. 2005]

Solution : This is a very simple inventory problem; Thirty five cakes are made every day. The demand is random and given by a discrete probability distribution. The demand for 10 days can be generated using the probability distribution and the given sequence of random numbers. The daily demand of cakes, probability and cumulative probability are given in Table 7.7. The last column of the table gives the range of two-digit random numbers, corresponding to the probability of occurrence of a particular daily demand.

Table 7.7

Daily Demand	*Probability*	*Commu. Prob.*	*Random Numbers*
0	.01	.01	00
15	.15	.16	01 – 15
25	.20	.36	16 – 35
35	.50	.86	36 – 85
45	.12	.98	86 – 97
50	.02	1.00	98 – 99

Now the daily demand of cakes for 10 days can be generated by using the given sequence of random numbers. Random No. 48 corresponds to day 1. It lies in the random number range 37 – 86, which corresponds to daily demand of 35 cakes. The demand for 10 day is simulated as given below, in Table 7.8.

Table 7.8

Day	1	2	3	4	5	6	7	8	9	10
Random No.	48	78	09	87	99	77	15	14	68	89
Demand	35	35	15	45	50	35	15	15	35	45

The production rate of cakes is 35 per day. It can be assumed that the left over stock of cakes will be carried over to next day, and any shortage in meeting the demand will amount to sales lost. The simulation of stock is carried out in Table 7.9.

On day 1, demand is 35 cakes, production or supply is also 35, hence no stock is left at the end of the day. Similarly on day 2 demand and supply are equal. But on day 3 demand is only 15 and hence 20 cakes are left in stock at the end of the day. One day 4 while demand is 45, the supply is 20 cakes of previous day and 35 made on 4th day, amounting to 55. Thus 55 – 45 = 10 cakes are left in stock on 4th day. On 5th day demand is 50, while supply is 10 + 35 = 45 cakes, resulting in a shortage of 5 cakes. This shortage cannot be carried to next day, and hence is lost sale. The process of simulation continues as in Table 7.9. At the end of 10th day, the cakes left in stock are 30.

Table 7.9

Day	*Demand*	*Supply*	*Stock left*
1	35	35	0
2	35	35	0
3	15	35	20
4	45	20 + 35	10
5	50	10 + 35	– 5
6	35	35	0
7	15	35	20
8	15	20 + 35	40
9	35	40 + 35	40
10	45	40 + 35	30

7.10 Exercises

1. An automobile tyre dealer operates his inventory on a single order model. As soon as the stock goes down to *P* units he places an order of *Q* units with the wholesaler. The order is placed at the end of the day and is received after three days that is at the beginning of 4th day. Only one order can be outstanding at a time. Any unfulfilled demand is lost forever. The demand on any day can be any number between 0 and 20, each equally probable. Write a simulation program to determine the service level for various combinations of P and Q. The levels of P can be taken as 15, 20, 25 and of Q as 20, 30, 40. Initial stock may be taken as 10 units. For each combination run the simulation for 200 days.
2. In the above problem, the lead-time is fixed at 3 days. Consider the more realistic case when the lead-time can vary slightly and can be any of 2, 3 and 4 days with probabilities of 0.15, 0.75 and 0.10 respectively. Modify the simulation model and run it for the given number of combinations of P and Q for 200 days.
3. In Exercise 1 it was assumed that the unfulfilled demand on any day is lost forever. But in many cases the orders can be held for some small durations if the customer agrees. Let us assume that 50% of the customers agree to hold unfulfilled part of their orders for one day. Modify the simulation model and run it for the said combinations of P and Q to determine the effect on service level.
4. In a single order inventory model the demand per day is distributed normally with mean of 50 units and standard deviation of 10. The lead-time follows Erlang distribution with mean of 3 days and Erlang parameter $k = 2$. The inventory carrying cost is Re. 1.00 per unit per day. Unfulfilled demand is demand lost and costs Rs. 20 per unit. Placement of each order costs Rs. 100. Taking the initial stock on hand as 100 units and no order outstanding, select the order quantity and reorder point so that the total cost is minimum. Reorder points between 115 and 200 and order quantity between 140 and 300 may be tested in some suitable steps.
5. A single order inventory model is given in Exercise 4. Develop a search procedure to determine the minimum cost inventory policy.
6. Simulate the inventory system described in Section 7.8 for two different starting conditions say with initial stock of zero in one, and with initial stock of 20 in second. Plot the variation in Average stock and service level for the first 50 days taking observations after every 10 days and give your comments on the curves obtained.
7. Develop a simulation model for a periodic review inventory system in which the highest inventory level is fixed say as Z units. The stock is checked after a fixed interval of time say *T* days, and order is placed for varying amount which is equal to the difference between the maximum stock level Z and the stock in hand at the time of review. The inventory carrying cost is Rs. c1 per unit per day, the cost of lost sales is Rs. c2 per unit and the ordering cost is Rs. c3 per order. The demand is stochastic and follows

a normal distribution with mean 'mus' and standard deviation 'sigma'. The lead-time follows exponential distribution with mean 'lt'. The simulation has to evaluate the various (I, Z) policies for a specified simulation run.

8. In Exercise 7 the order is placed for a varying order quantity after every fixed period of time. Now consider the situation in which the order is placed after the fixed interval of time only if the stock level is at or below a specified level P. If it is above P nothing is done till the next review time. Run the simulation for 500 days for the following data.

   Z = 100, P = 60, Demand mean = 10 and sigma = 3.

   Lead-time mean, lt = 3, c1 = 1.25, c2 = 20.0, c3 = 125.0

   What will be the effect on total cost if the value of P is made 50, 40 or 70?

9. A baker is trying to determine how many dozens of bread to bake each day. The probability distribution of the number of customers is as follows:

Number of customers per day	10	12	14	16
Probability	.35	.30	.20	.15

   Customers order 1, 2, 3, or 4 dozen breads according to the following distribution.

No. of Dozen ordered/customer	1	2	3	4
Probability	.4	.3	.2	.1

   Bread sells for Rs. 60.0 per dozen and cost Rs. 42 per dozen to make. The breads not sold at the end of the day are disposed off at half price. Determine, by simulating the system for 500 days, the dozens of bread, which should be baked each day.

10. A newsstand can buy a daily newspaper for 50 paise each and sell it for 85 paise. The unsold copies, if any, can be disposed of as waste paper at 20 paise each. The estimated daily demand distribution is as follows:

Demand (No. of copies)	*Probability*
100	.03
110	.07
120	.19
130	.28
140	.20
150	.10
160	.05
170	.05
180	.03

Develop a computer simulation model of the system to determine the optimal number of newspaper copies, which should be procured, so that the expected profit is maximum.