Chapter 1
Introduction to Simulation

System, Model and Simulation

System

- The term system is derive from the Greek word systema, which means an organized relationship among functioning units or components units or components.
- A system is defined as an aggregation of objects or components joined in some regular interaction or interdependence.
- Systems are designed to achieve one or more objectives.
- Interrelationship and interdependence must exists among the system components.
- The objectives of the organization as a whole have a higher priority than the objectives of its subsystems.

Model

- A model is a simplified representation of a system at some particular point in time or space intended to some particular point in time or space intended to promote understanding of the real system.

- Model is a conceptual framework that describes the system.
- Modelling is the process of representing a model which includes its construction and working. This model is similar to a real system, which helps the analyst predict the effect of changes to the system.

Simulation

- The representation of the behavior or characteristics of one system through the use of another system, specially a computer program designed for the purpose.
- It is a program that mimics (imitate) the behavior of the real system of the real system.
- 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.
- A model construct a conceptual framework that describes a system. The behavior of a system that evolves over time is studied by developing a simulation model.

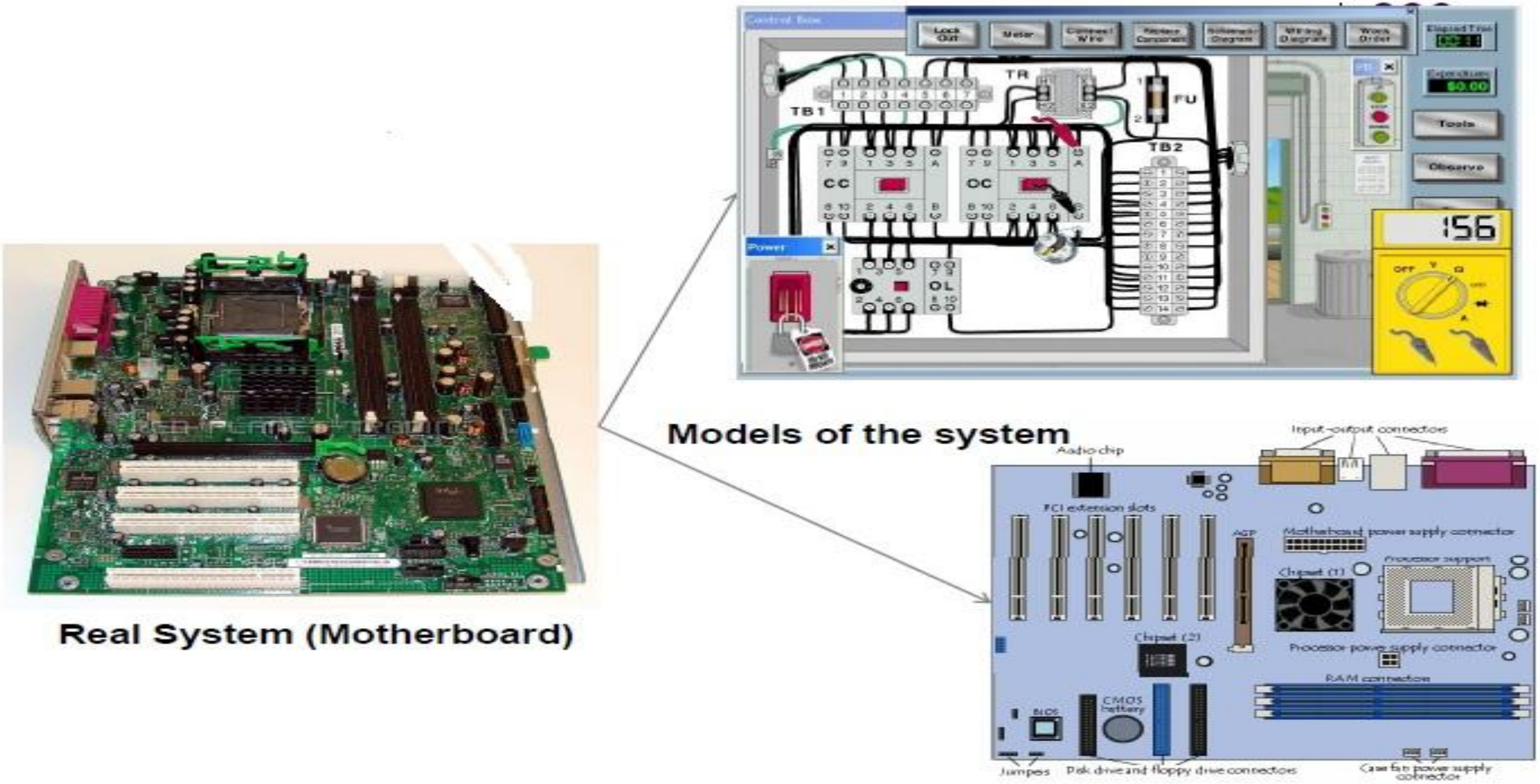


Figure: Example of model of a system

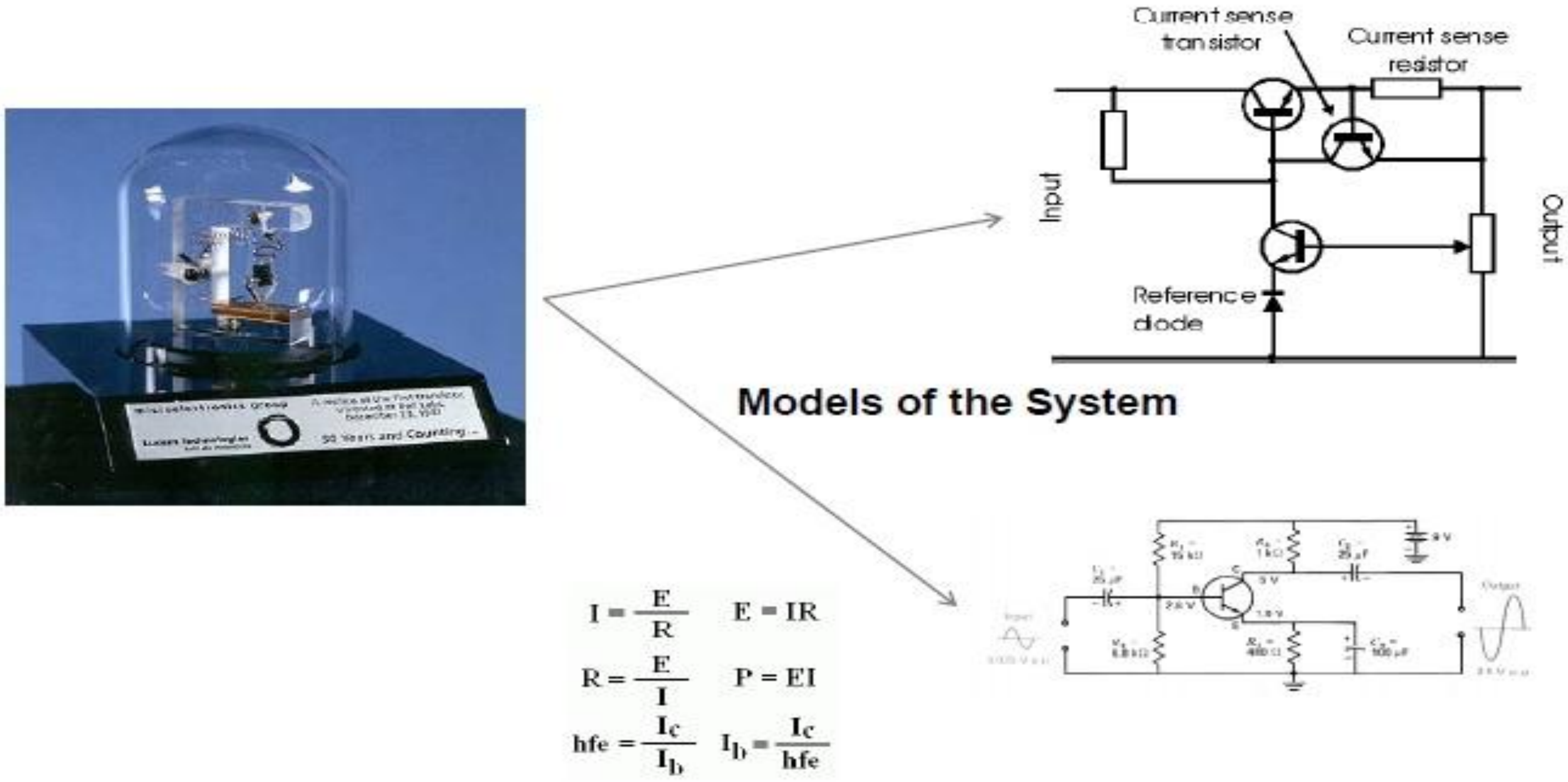


Figure: Example of model of a system

Why Simulation?

- It may be too difficult, hazardous, or expensive to observe a real, operational system
- A model can be used to investigate a variety of ‘what if’ questions about real-world system. Using simulation we can discover the change in system, output as the input parameter changes.
- Parts of the system may not be observable (e.g. internals of a silicon chip or biological system).
- Simulation can be used as an analysis tool for predicating the effect of changes.
- Simulation can be used as a design tool to predicate the performance of new system.

So it is better to do simulation before implementation.

When Simulation is Appropriate

- Simulation enable the study of internal interaction of a subsystem with complex system.
- Informational, organizational and environmental changes can be simulated and find their effects.
- A simulation model help us to gain knowledge about improvement of system.
- Finding important input parameters with changing simulation inputs.
- Simulation can be used with new design and policies before implementation.
- Simulating different capabilities for a machine can help determine the requirement.
- Simulation models designed for training make learning possible without the cost disruption
- The modern system (factory, wafer fabrication plant, service organization) is too complex that its internal interaction can be treated only by simulation

When Simulation is not Appropriate

- When problem can be solved analytically and easily.
- If it is easier to perform direct experiments.
- If the cost becomes too high such that cost exceeds saving.
- If resource and time are not available.
- If system behavior is too complex.

Discrete and Continuous System

Discrete System

- A discrete system is one in which the state variable(s) change only at a discrete set of points in time.
- Changes in the state variable(s) are predominantly discontinuous.
- Example: Number of customers waiting in line, Number of jobs in a queue, etc.

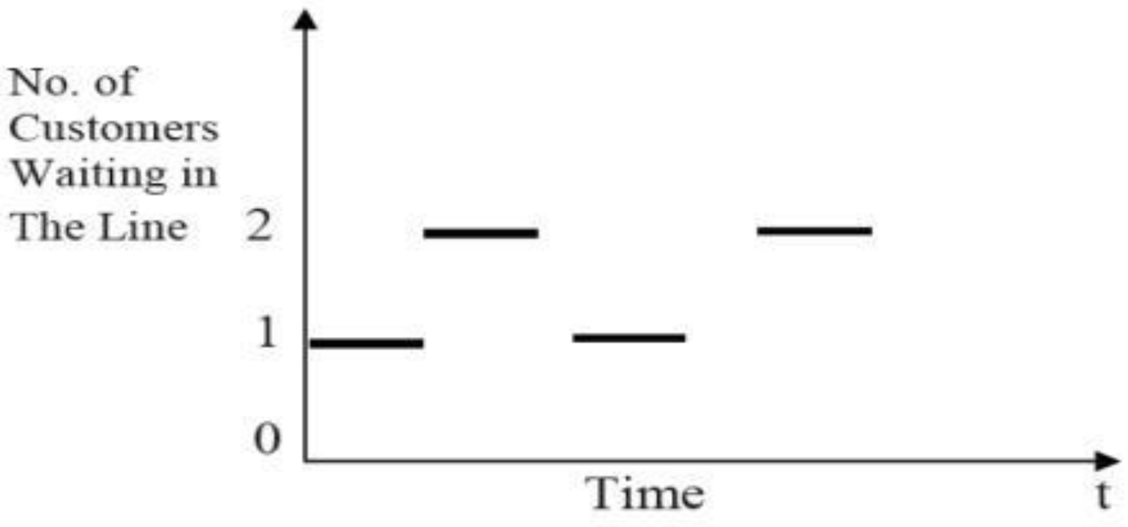


Figure: Discrete System Example

Continuous System

- A continuous system is one in which the state variable(s) change continuously over time.
- Changes in the state variable(s) are predominantly continuous and smooth without any delay.
- Example: Head of water behind the dam, etc.

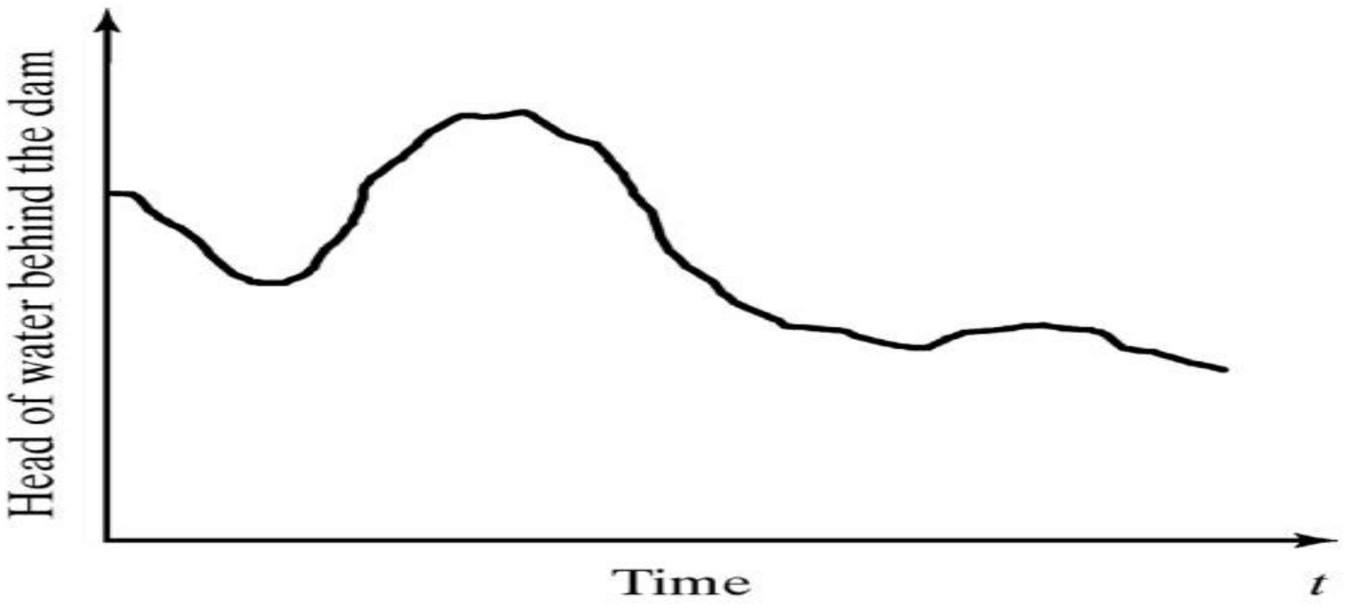
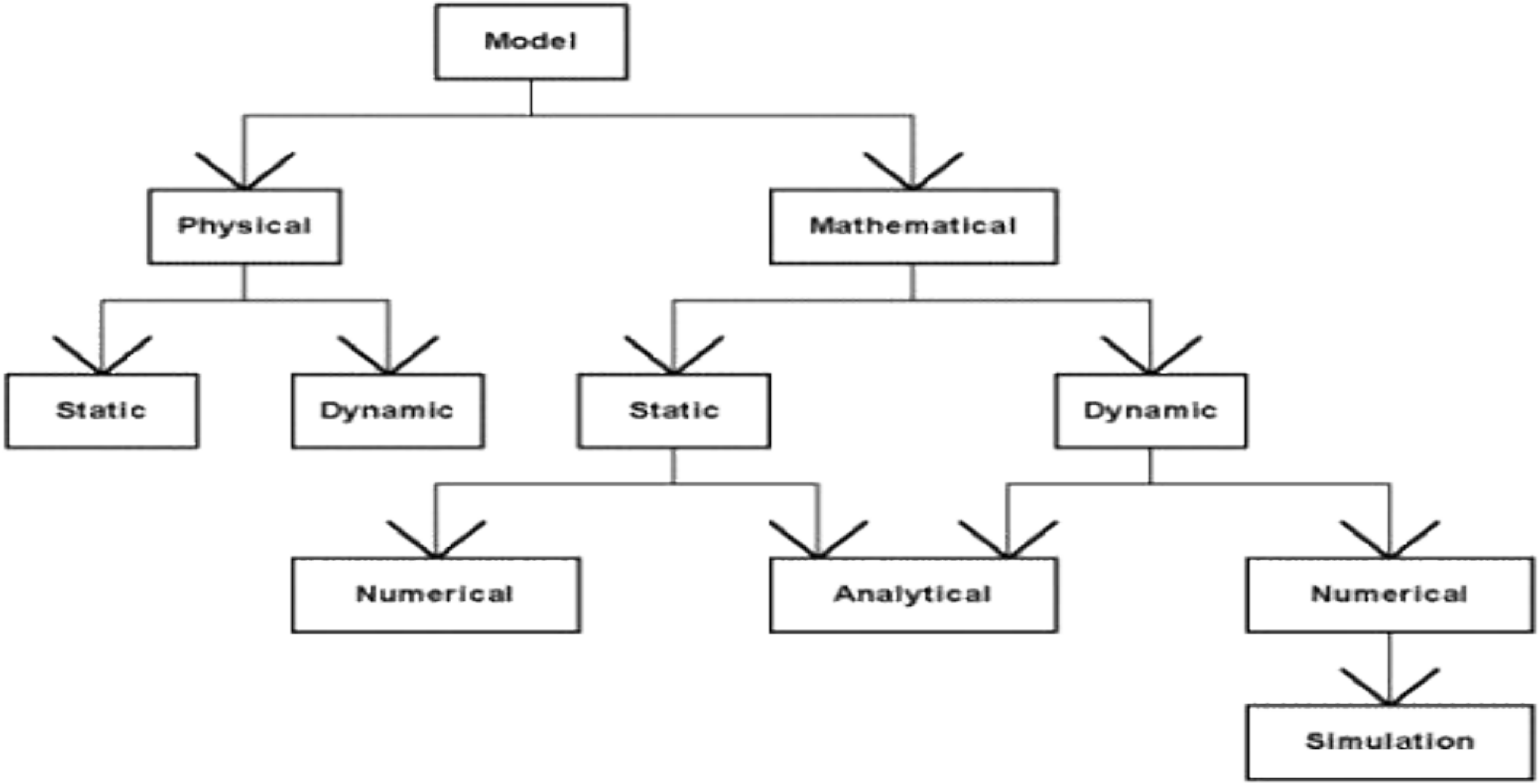


Figure: Continuous System Example

Types of Model



Physical Model

- Physical models are based on some analogy between such systems as mechanical and electrical or electrical and hydraulic.
- In a physical model of a system, the system attributes are represented by measurements such as voltage or the position of a shaft.
- The system activities are reflected in the physical laws that drive the model.
- For example the rate at which the shaft of a DC motor turns depends on the voltage applied to the motor.

Mathematical Model

- Mathematical models use symbolic notation and mathematical equation to represent a system.
- The system attributes are represented by variables, and the activities are represented by mathematical functions that interrelate the variables.

Static Model

- It is a type of model where time is not a significant variable.
- It is a representation of system at a particular point in time i.e. time plays no role.

Dynamic Model

- It is a type of model where time plays significant role and is a significant variable.
- It is the representation of a system that evolves over time.
- It describes the time-varying relationships.

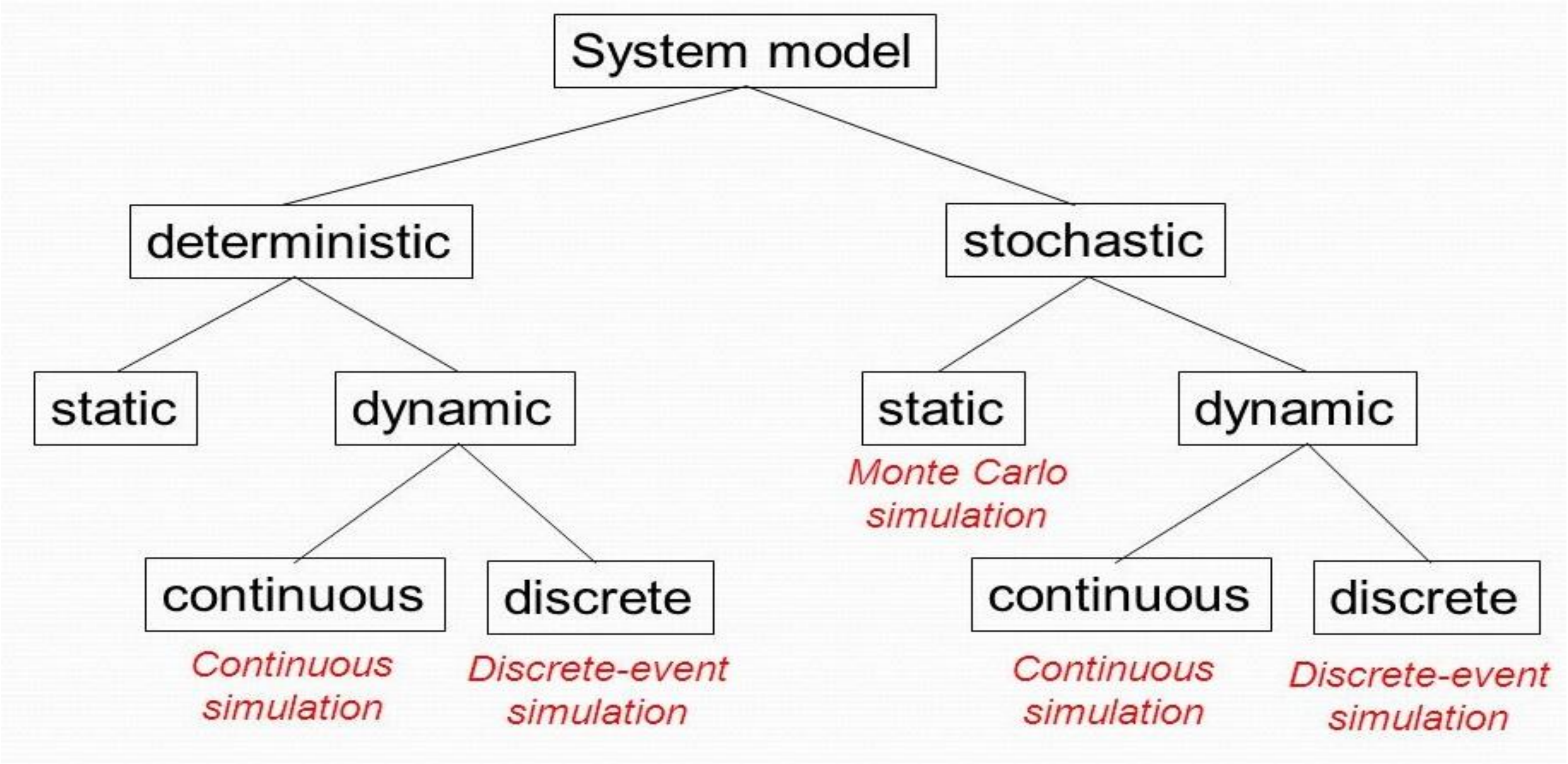
Analytical Model

- It is the one which is solved by using the deductive reasoning of mathematical theory.

Mathematical Model

- It is the one which is solved by using computational procedures.

Types of Simulation Model



Deterministic Simulation Model

- Deterministic models have a known set of inputs, which result into unique set of outputs.
- None of the system property is random.

Stochastic Simulation Model

- In stochastic model, there are one or more random input variables, which lead to random outputs.

Continuous Simulation Model

- Continuous simulation model represents system in which the state of the system changes continuously with time.

Discrete Simulation Model

- Discrete simulation model represents system in which the state of the system changes at discrete points.

Deterministic vs Stochastic Simulation Model

Deterministic Model	Stochastic Model
Deterministic models have known set of inputs which result in unique set of outputs.	Stochastic models have one or more random inputs which lead to random outputs.
Doesn't contain random elements. Output is deterministic quantity.	Contains random(probabilistic) elements. Output is random quantity.
The functional relationships that exists are known with certainty.	There are some uncertain functional relationships.
Examples: Simulation of chemical reaction based on differential equations, simulation of digital circuits, etc.	Examples: Queuing Models like arrival time of customers at a restaurant, amount of time required to service a customer, etc.

Static vs Dynamic Simulation Model

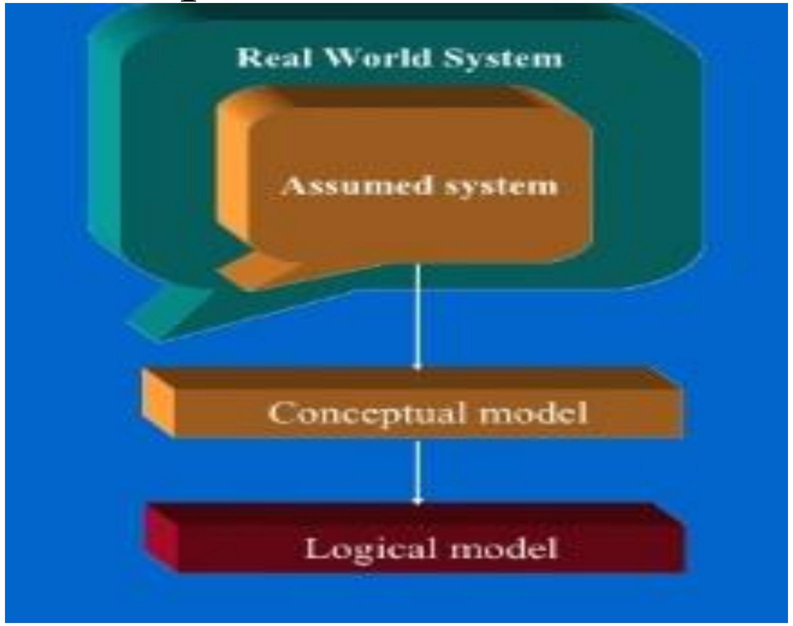
Static Model	Dynamic Model
Model represents a system that doesn't evolve over time.	Model represents a system that evolves over time.
Time doesn't play important role. Model represents system at a particular point of time.	Time plays a vital role.
Static model is more structural than behavioral.	Dynamic model is more behavioral than structural.
Static model is more rigid than dynamic modeling as it is a time independent view of a system.	Dynamic modeling is flexible as it can change with time.
Example: Monte Carlo Simulation, Model that calculates mechanical stress in a bridge, etc.	Example: Model of a processor, etc.

Continuous vs Discrete Simulation Model

Continuous Model	Discrete Model
Model represents system in which the state of the system changes continuously with time.	Model represents system in which the state of the system changes at discrete points.
The state variables change in a continuous way.	The state variables change only at a countable number of points in time.
Example: Model representing velocity of fluid in a pipe or channels, etc.	Example: Model of a system representing number of jobs in a queue, etc.

Steps in Simulation Study

1. **Problem formation**: The initial step involves defining the goals of the study and determing what needs to be solved. The problem is further defined through objective observations of the process to be studied
2. **Model Conceptualization:** This phase involves conceptualization of model which involves establishing a reasonable model. Essential features of the real world system are abstracted according to which an Assumed system is developed. From the Assumed system a conceptual model is developed which includes a more detailed specification of the system, important entities, relationships which is further developed into a logical model..



3. Data Collection: In this phase first the type of data to collect is determined and collection of data for input analysis and validation is done.

4. Model Translation: The model is translated into programming language. Choices range from general purpose languages such as C, C++, Fortran or simulation programs such as Arena.

5. Verification and Validation: Verification is the process of ensuring that the model behaves as intended. Validation is the process of determining whether the model accurately represents the system or not. Verification is performed before validation. Model verification answers for Did we build the model right? where as validation answers for Did we build the right model?

6. Experimental Design: The alternatives that are to be simulated must be determined. Factors such as number of simulations to run, length of each run, type of output data are determined.

7. Simulation Run and Analysis: The simulation is now run and the output of the simulation is collected and hence analyzed. The result is interpreted.

8. Documentation and Report: Documentation of the final model is prepared and the result of the simulation is reported. There are two types of documentation.

9. Implementation: The output of the simulation is analyzed and if expected output is achieved then finally the assumed system is implemented.

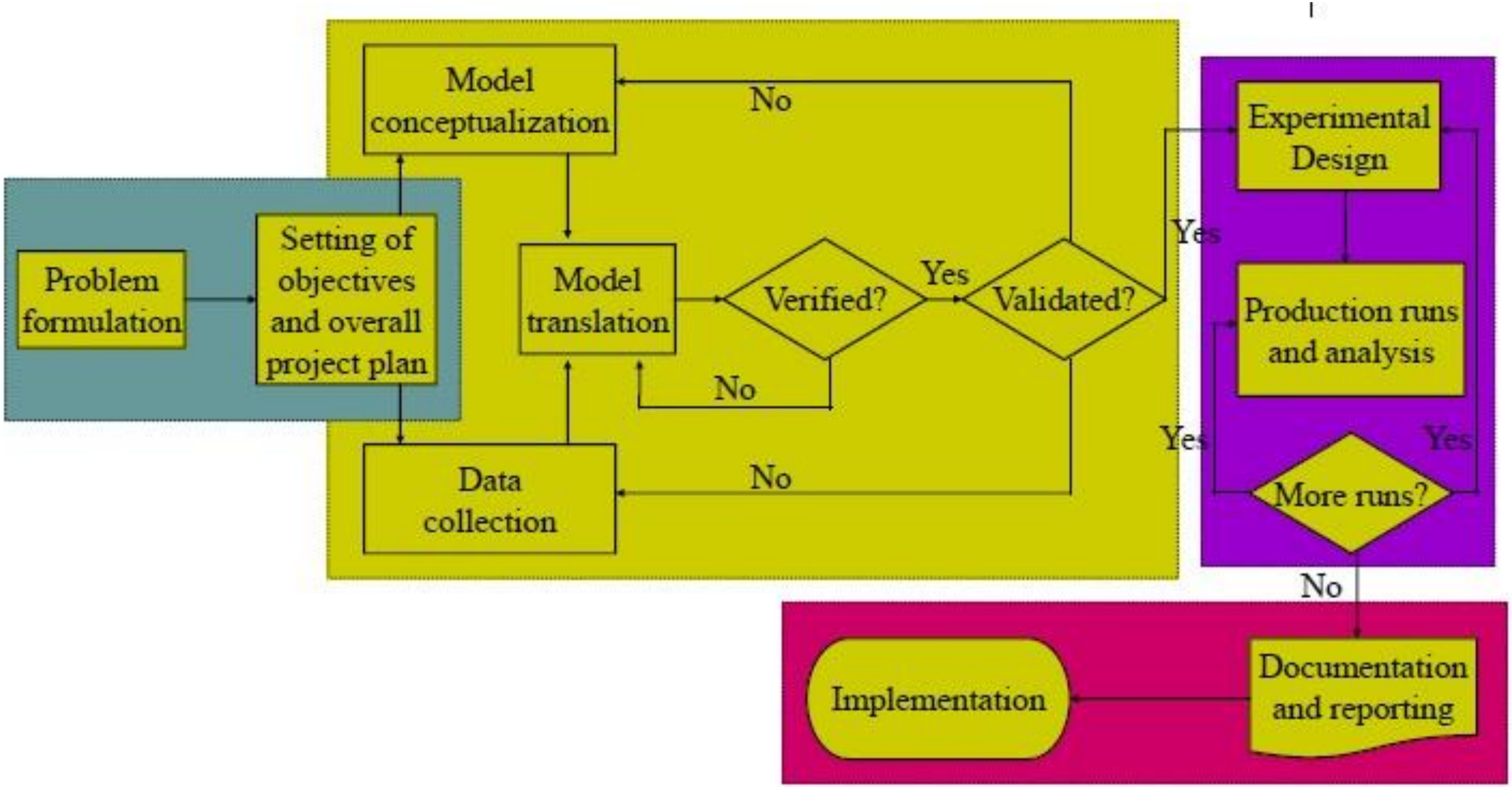
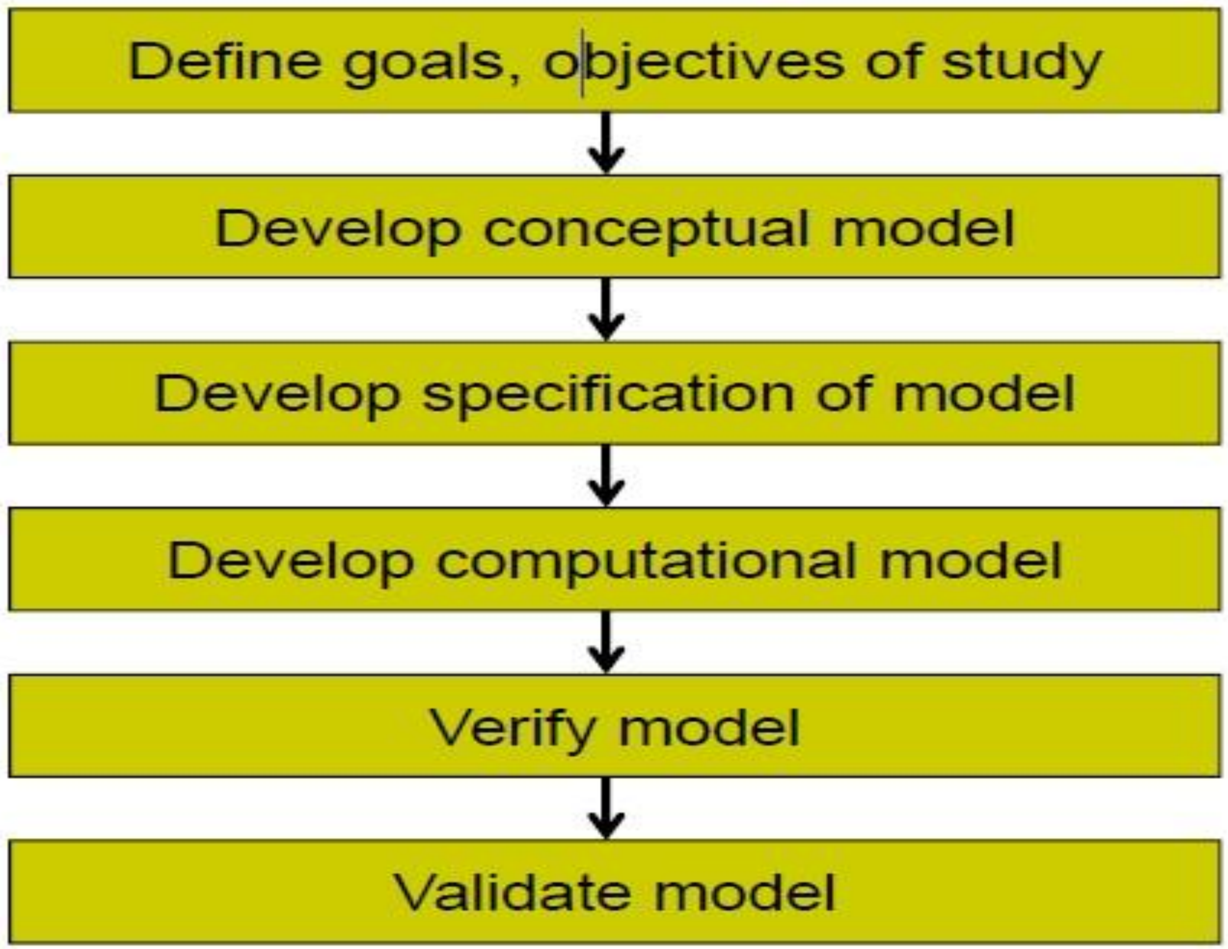


Figure: Flowchart For Steps In Simulation Study

Model Development Lifecycle



Steps in Model Development Life Cycle

1. **Define goals, objectives of study:** The goals and objectives for which the model is being developed should be identified and defined clearly.
2. **Develop Conceptual Model:** Once the goals and objectives are defined, the conceptual model should be developed now. A conceptual model is a representation of a system, made of the composition of concepts which are used to help people know, understand or simulate a subject the model represents. During the development of conceptual model, the main idea and concepts about the system for which the model is being developed must be found.
3. **Develop Specification of Model:** This phase involves a more detailed specification of the model. Collection of data, development of necessary algorithms are done in this phase. Empirical data or probability distributions often used in this phase.
4. **Develop Computational Model:** A computational model is a mathematical model in computational science that requires extensive computational resources to study the behavior of complex system. It is the executable simulation model. The specification model is developed into computational model in this phase.

5. Verify Model: This phase involves the verification of model. Verification is the process of ensuring that the model behaves as intended. Model verification answers for Did we build the model right?, Does the computational model match the specification model?, etc.

6. Validate Model: This phase involves the validation of model. Validation is the process of determining whether the model accurately represents the system or not. Model validation answers for Did we build the right model?, Does the computational model match the actual system?, etc.

Advantages and Disadvantages of Simulation

Main Advantages

- Simulation helps to learn about real system, without having the system at all. It helps to study the behavior of a system without building it.
- New hardware designs, physical layouts, transportation systems and various systems can be tested without committing resources for their acquisition.
- Simulation Models are comparatively flexible and can be modified to accommodate the changing environment to the real situation.
- Simulation technique is easier to use and can be used for wide range of situations.
- In systems like nuclear reactors where millions of events take place per second, simulation can expand the time to required level.
- Results are accurate in general, compared to analytical model.
- Help to find un-expected phenomenon, behavior of the system.
- Easy to perform ``What-If'' analysis.

Main Disadvantages

- Expensive and difficult to build a simulation model. Model building requires special training.
- Expensive to conduct simulation.
- Sometimes it is difficult to interpret the simulation results. Since most simulation outputs are essentially random variables, it may be hard to determine whether an observation is a result of system interrelations or randomness.
- Simulation results may be time consuming.

Applications of Simulation

- **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, and 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**: Applications such as planning of health services, expected patient density, facilities requirement, hospital staffing , estimating the effectiveness of a health care program.
- **Communication Applications**: Applications such as network design, and optimization, evaluating network reliability, manpower planning, sizing of message buffers.

- **Computer Applications**: Can be applicable in fields such as designing hardware configurations and operating system protocols, sharing networking, gaming.
- **Economic applications**: Can be used in 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.
- **Environment application**: Solid waste management, performance evaluation of environmental programs, evaluation of pollution control systems.
- **Biological applications**: Such as population genetics and spread of epidemics.

Chapter 2
Physical And Mathematical Models

Static Physical Models

- Physical models are such models where the system attributes are represented by physical measurements such as voltage or position of shaft.
- Static physical model is a scaled down model of a system which does not change with time.
- Static physical model is the physical model which describes relationships that do not change with respect to time.
- Such models only **depict the object's characteristics at any instance of time, considering that the object's property will not change over time.**
- Example : An architectural model of a house, scale models and so on.
- The best known examples of physical models are scale models.
- In shipbuilding, making a scale model provides a simple way of determining the exact measurements of the plates covering the hull, rather than having to produce drawings of complicated, three-dimensional shapes.

- Scientists have used models in which spheres represent atoms, and rods or specially shaped sheets of metal connect the spheres to represent atomic bonds.
- Scale models are also used in wind tunnels and water tanks in the course of designing aircraft and ships.
- **Sometimes, a static physical model is used as a means of solving equations with particular boundary conditions.**
- There are many examples in the field of mathematical physics where the same equations apply to different physical phenomena. For example, the flow of heat and the distribution of electric charge through space can be related by common equations.

Dynamic Physical Models

- Dynamic physical models rely upon an analogy between the **system being studied** and **some other system of a different nature**, the analogy usually depending upon an underlying similarity in the forces governing the behavior of the systems.
- To illustrate this type of physical model, consider the two systems shown in following figures below.

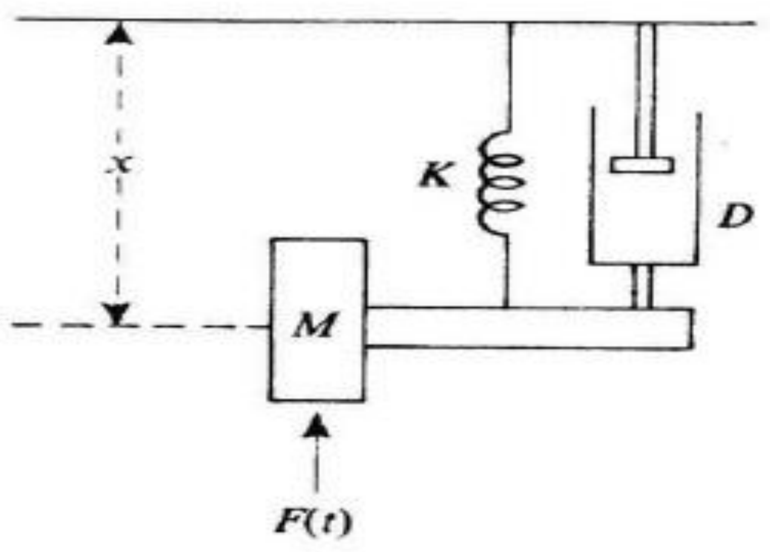


Figure 1: Mechanical System

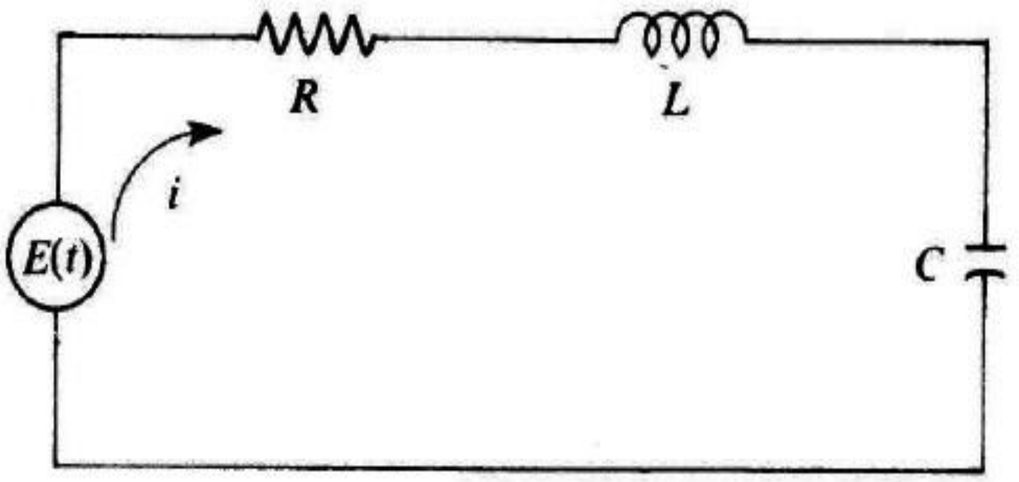


Figure 2: Electrical System

- Figure 1 shows a mass spring system where a mass M is subject to an applied force F(t) varying with time. The force on spring is directly proportional to expansion or contraction of the spring with spring constant k. There is also a shock absorber of damping constant D that exerts a damping force proportional to the velocity of mass.
- This system might represent an example of suspension of automobile wheel.
- The motion of the system can be described by the following differential equation

$$M\ddot{x} + D\dot{x} + Kx = KF(t)$$

Where x is the distance moved,

M is the mass,

K is the stiffness of the spring or spring constant,

D is the damping factor of the shock absorber

- Figure 2 shows an electrical system that contains Resistance R, Capacitor of capacitance C and Inductor of inductance L and are connected in series with a voltage source E(t) which varies with time.
- If q is the charge on the capacitance, it can be shown that the behavior of the circuit is governed by the following differential equation

$$L\ddot{q} + R\dot{q} + \frac{q}{C} = \frac{E(t)}{C}$$

- Both these mechanical and electrical system exactly have the same form of equation and hence following equivalences occur between these system.

Mechanical System	Electrical System
Displacement (x)	Charge (q)
Force (F)	Voltage (E)
Mass (M)	Inductance (L)
Damping Factor (D)	Resistance (R)
Spring Stiffness (K)	1/Capacitance(C)

Hence the mechanical and electrical systems are analogous to each other.

- Since these systems are analogous, the performance of one can be studied with the other.
- In practice, it is simpler to modify the electrical system than to change the mechanical system, so it is more likely that the electrical system will have been built to study the mechanical system.

- If, for example, a car wheel is considered to bounce too much with a particular suspension system, the electrical model will demonstrate this fact by showing that the charge (and, therefore, the voltage) on the condenser oscillates excessively.
- To predict what effect a change in the shock absorber or spring will have on the performance of the car, it is only necessary to change the values of the resistance or condenser in the electrical circuit and observe the effect on the way the voltage varies.
- If in fact, the mechanical system were as simple as illustrated, it could be studied by solving the mathematical equation derived in establishing the analogy.
- However, effects can easily be introduced that would make the mathematical equation difficult to solve.

Static Mathematical Models

- Mathematical models use symbolic notation and mathematical equation to represent a system.
- A static model gives the relationships between the system attributes **when system is in equilibrium.**
- Static Mathematical Models are such mathematical models that give the relationships between the system attributes when system is in equilibrium.
- These are the mathematical models that represent system at a particular point of time.
- If the point of equilibrium is changed by altering any of the attribute values, the model enables the new values for all the attributes to be derived but does not show the way in which they changed to their new values.
- For example, in marketing a commodity there is a balance between the supply and demand for the commodity

- Both factors demand and supply depend upon price.
- **Let S represent Supply, Q represent Demand and P represents Price.**
- Demand for the commodity will be low when the price is high and it will increase as the price drops.
- The relationship between demand and price might be represented by the straight line marked "Demand“ in following figure.
- Similarly the supply can be expected to increase as the price increases, because the suppliers see an opportunity for more revenue.
- The relationship between supply and price might also be represented by the straight line marked “Supply” in following figure.

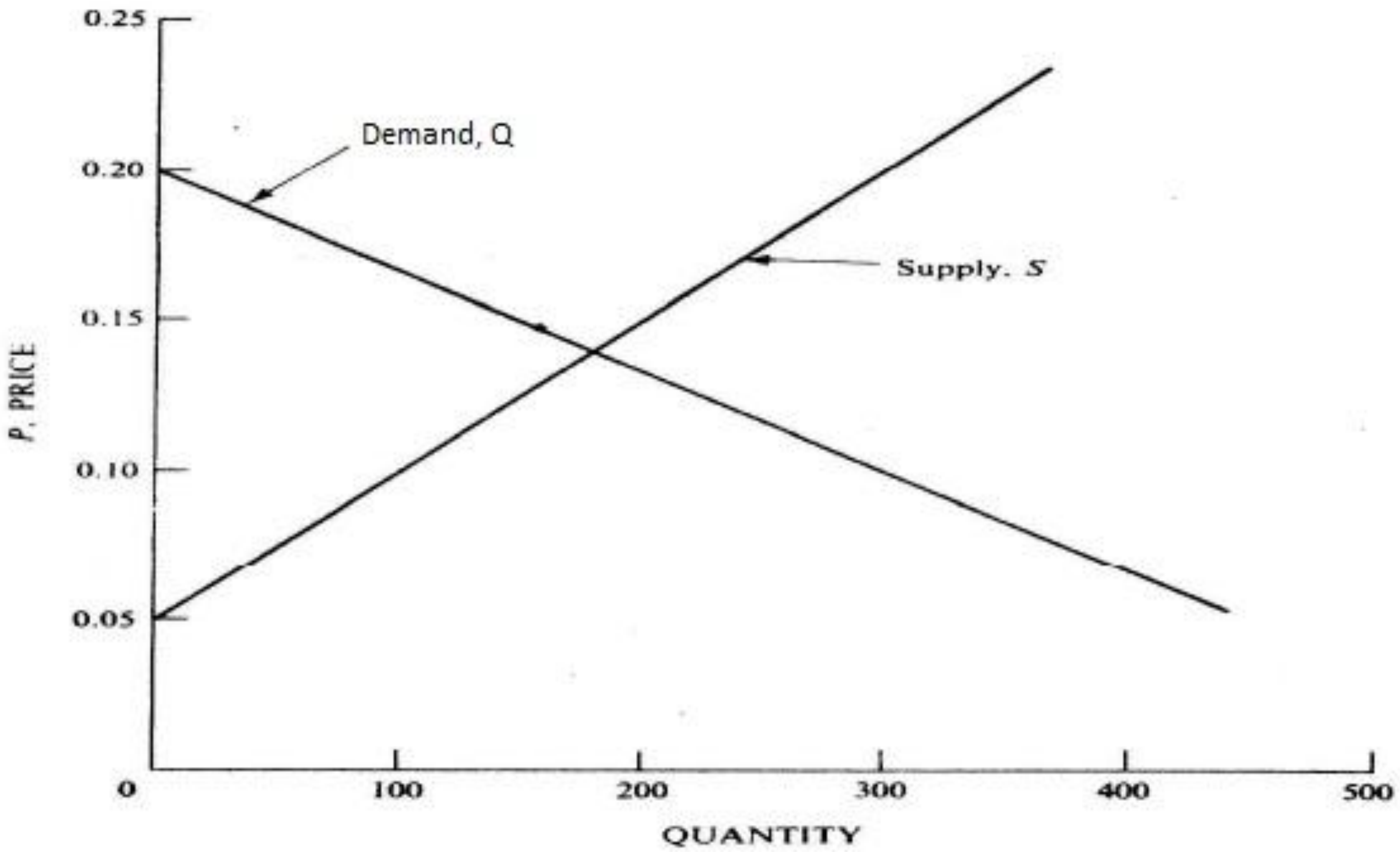


Figure: Linear Market Model

 If conditions remain stable, the price will settle to the point at which the two lines cross, because that is where the supply equals the demand.

 Since the relationships have been assumed linear, the complete market model can be written mathematically as follows:

$$Q = a - bP$$

$$S = c + dP$$

$$S = Q$$

 For the model to correspond to normal market conditions in which demand goes down and supply increases as price goes up the coefficients b and d need to be positive numbers.

 For realistic, positive results, the coefficient a must also be positive. Above figure has been plotted for the following values of the coefficients:

- a=600
- b=3000
- c=-100
- d=2000

 The fact that linear relationships have been assumed allows the model to be solved analytically. The equilibrium market price, in fact, is given by the following expression:

$$P = \frac{a-c}{b+d}$$

 With the chosen values, the equilibrium price is 0.14, which corresponds to a supply of 180.

 More usually, the demand will be represented by a curve that slopes downwards, and the supply by a curve that slopes upwards as shown below.

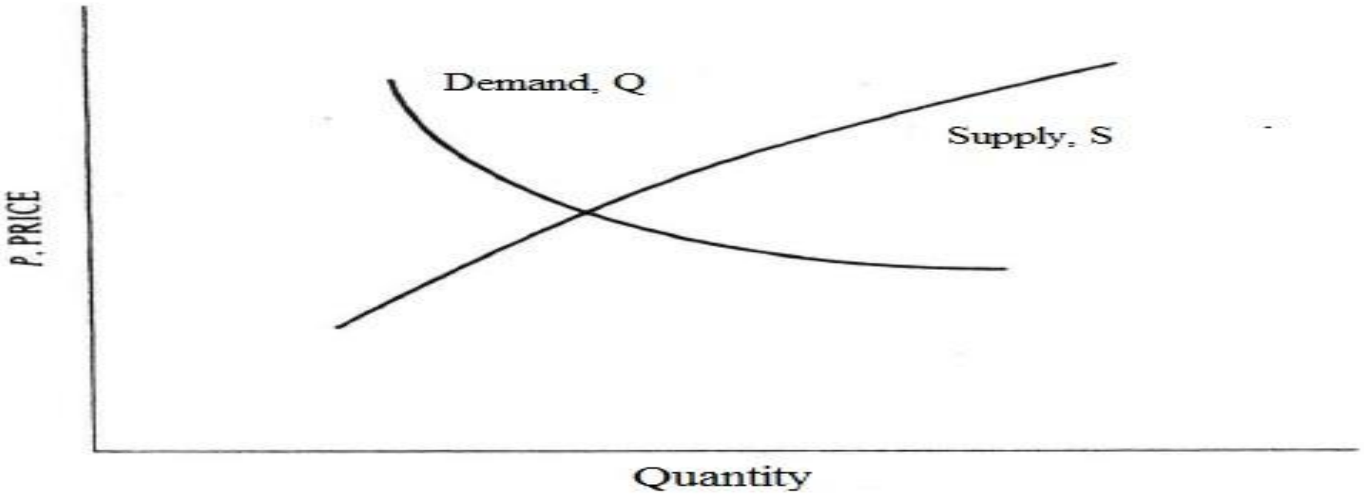


Figure: Non-Linear Market Model

- For such scenario, it may not then be possible to express the relationships by equations that can be solved.
- **Drawing the curves to scale and determining graphically where they intersect** is one such method. In practice, **it is difficult to get precise values** for the coefficients of the model.
- **Observations over an extended period of time can be performed.** The values depend upon economic factors, so the observations will usually attempt to correlate the values with the economy allowing the model to be used as a means of forecasting changes in market conditions for anticipated economic changes.

Dynamic Mathematical Models

- A dynamic mathematical model allows the changes of system attributes to be derived as a function of time.
- The derivation may be made with an analytical solution or with a numerical computation, depending upon the complexity of the model.
- The equation that was derived to describe the behavior of a **car wheel is an example of a dynamic mathematical model**. In this case, an equation that can be solved analytically.
- The equation is written as,

$$\ddot{x} + 2\zeta\omega\dot{x} + \omega^2 x = \omega^2 F(t)$$

where $\zeta \rightarrow$ Zeta, $2\zeta\omega = \frac{D}{M}$ and $\omega^2 = \frac{K}{M}$

- Expressed in this form, solutions can be given in terms of the variable ωt.
- The plot between x and ωt for various values of ζ is shown in figure below.

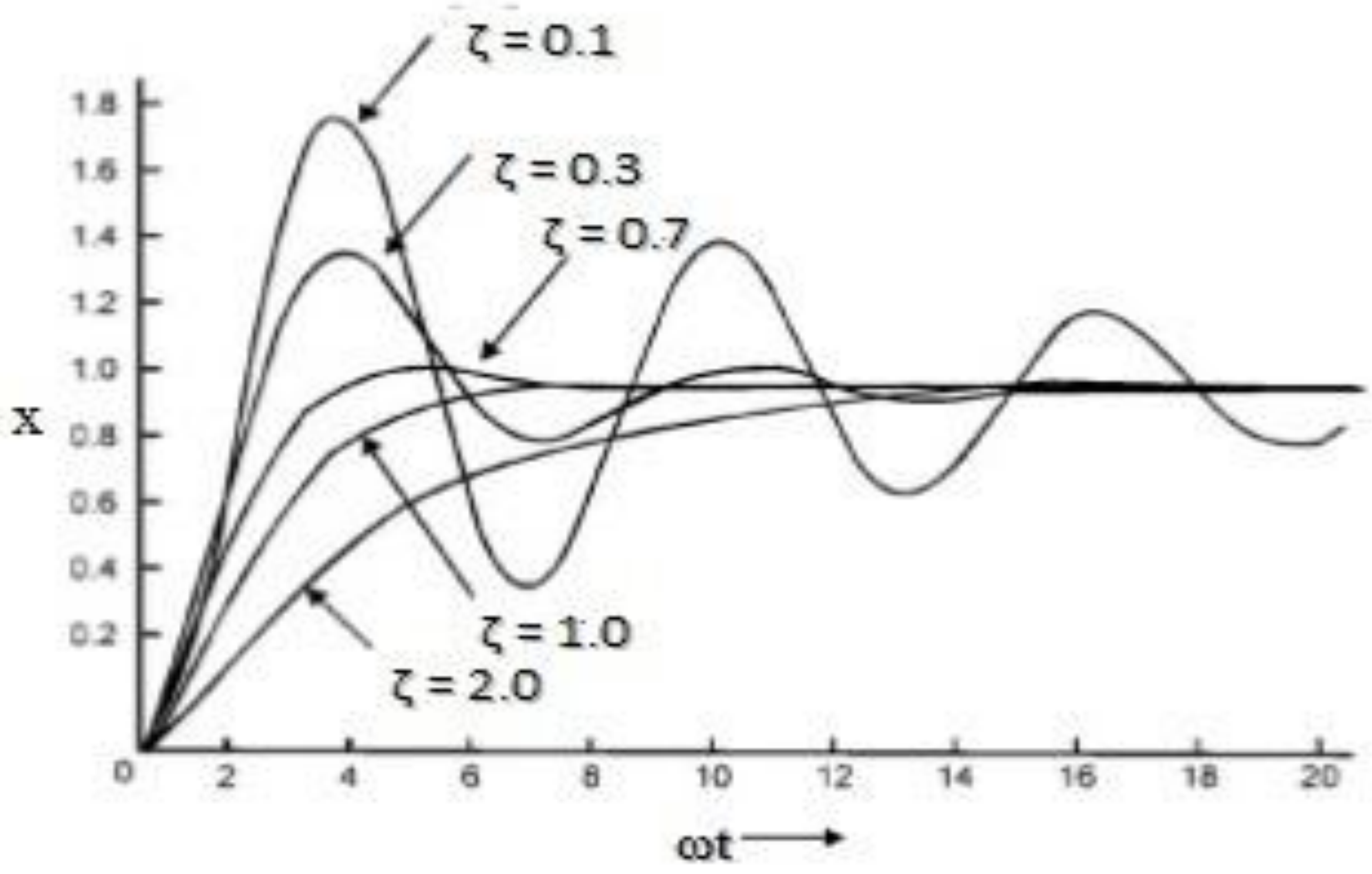


Figure: Solution of Second Order Differential Equations

- Solutions are shown for several values of ζ and it can be seen that when ζ is less than 1, the motion is oscillatory.
- The factor ζ is called the **damping ratio** and, when the motion is oscillatory, the frequency of oscillation is determined from the formula:

$$\omega = 2\pi f$$

where f is the number of cycles per second.

- The condition for the motion to occur without oscillation requires that $\zeta \geq 1$. It can be deduced from the definition of that the condition requires that $D^2 \geq 4MK$ since $2\zeta\omega = \frac{D}{M}$ and $\omega^2 = \frac{K}{M}$.

Principles Used In Modeling

1. **Block-Building**: The description of the system should be organized in **a series of blocks**. The main aim of these blocks is **to simplify the specifications of the interactions within the system.** The system as a whole can be described as the interconnections within the system.
2. **Relevance:** The model should only include those aspects that are relevant to the study objectives. Irrelevant information in the system might not do any harm, it should be excluded because it increases the complexity of the model.
3. **Accuracy:** The accuracy of the information gathered for the model should be considered. For example in the aircraft system the accuracy with which movement of the aircraft is described depends upon the representation of the airframe.
4. **Aggregation:** A further factor to be considered is the **extent to which the number of individual entities can be grouped together into larger entities.** In some cases, it may be necessary to construct artificial entities through the process of aggregation. Similar considerations of aggregation should be given to representation of activities.

Chapter 3
Continuous System Simulation

Continuous System

- A continuous system is one in which the state variable(s) change continuously over time.
- A continuous system is one in which the predominant activities of the system causes smooth changes in the attributes of the system entities.
- Changes in the state variable(s) are predominantly continuous and smooth without any delay.
- When such systems are modeled mathematically, the attributes of the system are controlled by a continuous functions.
- The continuous system is modeled using the differential equations.

Differential and Partial Differential Equations

Differential Equation

- The equation that consists of the higher order derivatives of the dependent variable is known as differential equations.
- A differential equation is a mathematical equation that relates some function with its derivatives.

Non-Linear Differential Equation

Let us assume an equation $M\ddot{x} + D\dot{x} + Kx = KF(t)$. Here M,D,K are constants, x is dependent variable and t is independent variable.

- The differential equation is said to be non-linear if any product exists between the dependent variable and its derivative or between the derivatives themselves.
- The differential equation is said to be non-linear if the dependent variable or any of its derivatives are raised to a power or are combined in other way like multiplication.

- $$\frac{d^3y}{dx^3} - \left(\frac{dy}{dx}\right)^2 + 4y = 4e^x \cos x$$

Product between two derivatives ---- non-linear

$$\frac{dy}{dx} + 4y^2 = \cos x$$

Product between the dependent variable themselves ---- non-linear

Linear Differential Equation

- The differential equation is said to be linear if no any product exists between the dependent variable and its derivative or between the derivatives themselves.
- The differential equation is said to be linear if any of the dependent variables and its derivatives have power of one(i.e. no higher powers) and are multiplied by the constant.
- Example: $\frac{d^2y}{dx^2} + 3\frac{dy}{dx} + 9y = 0$, $\frac{dy}{dx} + y = 0$, etc.

Partial Differential Equation

- When more than one independent variable occurs in a differential equation, the equation is said to be partial differential equation.
- A partial differential equation (PDE) is an equation involving functions and their partial derivatives.

Necessity of Differential Equation

1. Most physical and chemical process occurring in the nature involves rate of change, which requires differential equations to provide mathematical model.
2. It can be used to understand **general effects of growth trends** as differential equations can represent a growth rate.

Continuous System Models

- Models developed from continuous systems.
- The continuous system is modeled using the differential equations.
- When such systems are modeled mathematically, the attributes of the system are controlled by a continuous functions.
- In a continuous system, the relationships describe the rate at which system attributes change. So the model consists of differential equation.

Analog Computers

- An analog computer is a type of computer that uses the **continuously changeable aspects of physical phenomena** such as electrical, mechanical, or hydraulic quantities to model the problem being solved.
- Analog computers are those computers that are unified with devices like **adder and integral so as to simulate the continuous mathematical model of the system, which generates continuous outputs.**
- The most widely used form of analog computer is the electronic analog computer, based on the **use of high gain dc(direct current) amplifiers, called operational amplifiers.**
- In such analog computer, **voltages are equated to mathematical variables** and **the op amps can add and integrate the voltages.**
- With appropriate circuits, an amplifier can be <u>made to add several input voltages, each representing a variable of the model</u>, to produce a voltage representing the sum of the input variables.
- Different scale factors can be used on the inputs or input values to represent coefficients of the model equations.

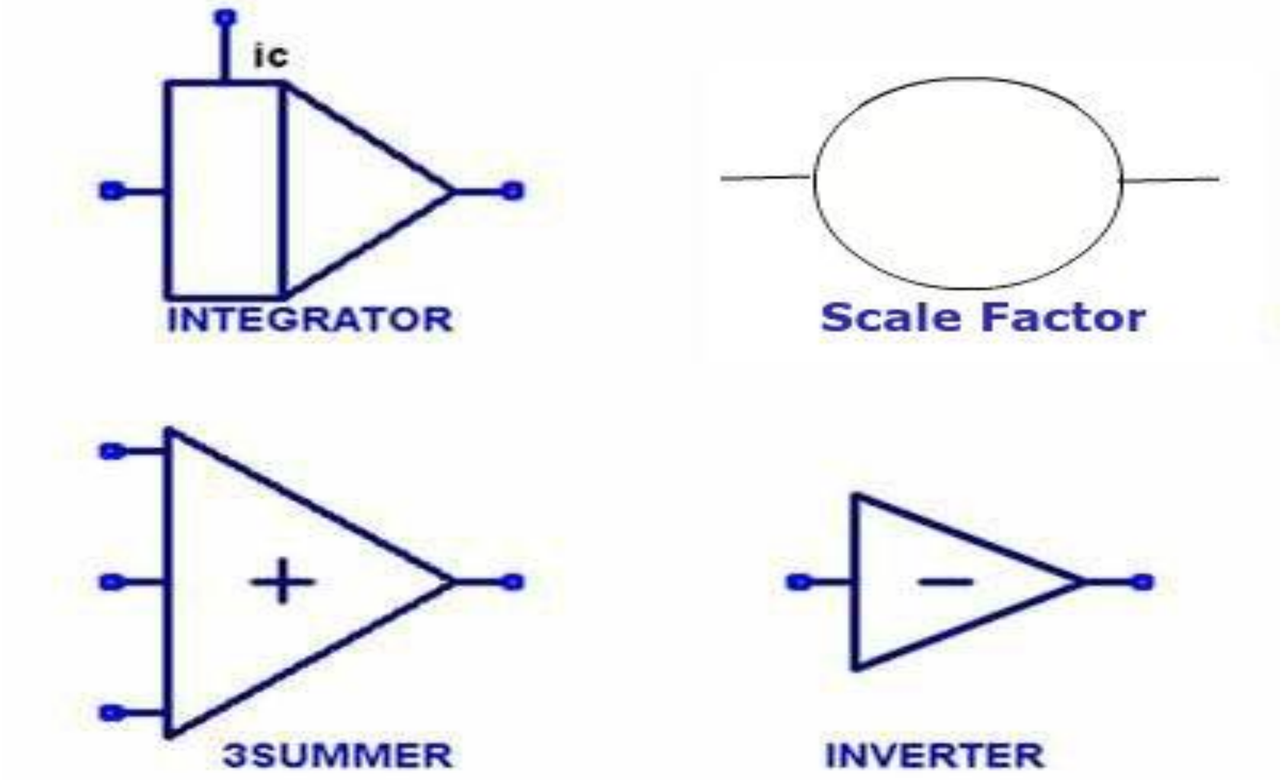
- Another circuit arrangement produces an integrator for which the output is the integral with respect to time of a single input voltage or the sum of several input voltages.
- All voltages can be positive or negative to correspond to the sign of the variable represented.
- Sign inverter can be used to reverse the sign of the input as per the requirement of the model equation.
- Electronic analog computers are limited in accuracy for several reasons. It is difficult to carry the accuracy of measuring a voltage beyond a certain point.
- A number of assumptions are made in deriving the relationships for operational amplifiers, none of which is strictly true. So, amplifiers do not solve the mathematical model with complete accuracy.
- Another type of difficulty is presented by the fact that the operational amplifiers have a limited dynamic range of output, so that scale factors must be introduced to keep within the range.
- As a consequence, it is difficult to maintain an accuracy better than 0.1% in an electronic analog computer.

- A digital computer is not subject to the same type of inaccuracies.
- Virtually any degree of accuracy can be programmed and, with the use of floating-point representation of numbers, an extremely wide range of variations can be tolerated.
- A digital computer also has the advantage of being easily used for many different problems.
- An analog computer must usually be dedicated to one application at a time, although time-sharing sections of an analog computer has become possible.
- In spite of the widespread availability of digital computers, many users prefer to use analog computers. There are several considerations involved.
- The analog representation of a system is often more natural in the sense that it directly reflects the structure of the system; thus simplifying both the setting up of a simulation and the interpretation of the results.
- Under certain circumstances, an analog computer is faster than a digital computer, principally because it can solve many equations in a truly simultaneous manner.

- Whereas a digital computer can be working only on one equation at a time, giving the appearance of simultaneity by interfacing the equations.
- On the other hand, the possible disadvantages of analog computers, such as limited accuracy and the need to dedicate the computer to one problem, may not be significant.

Analog Methods

- The general method by which analog computers are applied can be demonstrated using the second-order differential equation.
- The general method to apply analog computers for the simulation of continuous system models involves following components:



- The equation representing the car wheel system(automobile suspension problem) is

$$M\ddot{x} + D\dot{x} + Kx = KF(t)$$

$$\text{or, } M\ddot{x} = KF(t) - D\dot{x} - Kx$$

Suppose a variable representing the input F(t) is supplied, and assume for the time being that there exist variables representing x and $\frac{dx}{dy}$ i.e. $\dot{x}$.

- These three variables can be scaled and added with a summer to produce a voltage representing $M\ddot{x}$.
- This variable($M\ddot{x}$) is first scaled by a scaling factor $\frac{1}{M}$ and the result is supplied to an integrator which produces $\frac{dx}{dy}$ i.e. $\dot{x}$.
- Also an inverter is used which changes the sign of the variable and hence produces

$-\dot{x}$. Again this variable is fed to an integrator which produces –x.

 For convenience, a further sign inverter is included to produce $+x$ as an output.

 Block Diagram to solve the automobile suspension problem is shown below:

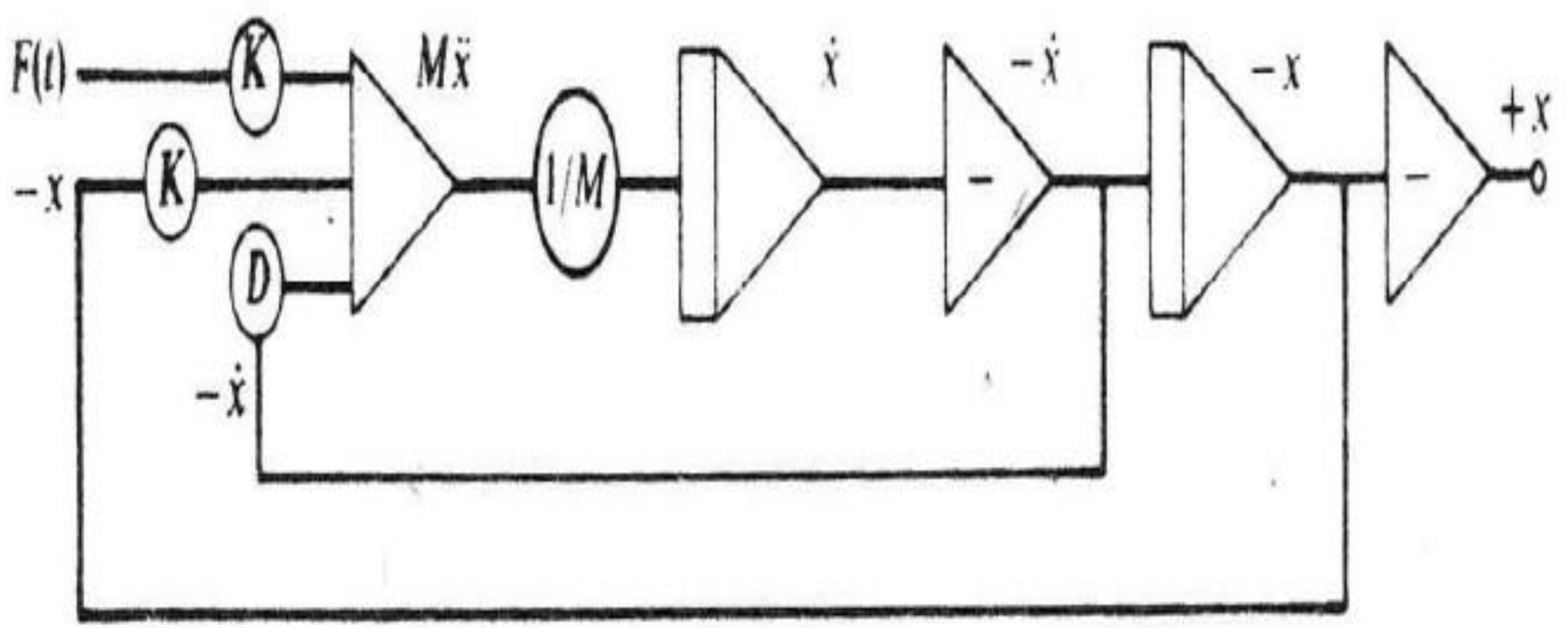


Figure: Diagram for automobile suspension problem

- The addition on the left, with its associated scaling factors, corresponds to the addition of the variables representing the three forces on the wheel, producing a variable representing $M\ddot{x}$.
- The scale is changed to produce $\ddot{x}$ from $M\ddot{x}$ and the result is integrated twice to produce $\dot{x}$ and x.
- Sign changers are introduced so that variables of the correct sign can be fed back to the adder, and the output can be given in convenient form.
- With an electronic analog computer, the variables that have been described would be voltages, and the symbols would represent operational amplifiers arranged as adders, integrators, and sign changers.
- The above figure would then represent how the amplifiers are interconnected to solve the equation.
- There can be several ways of drawing a diagram for a particular problem, depending upon which variables are of interest, and on the size of the scale factors.

 When a model has more than one independent variable, a separate block diagram is drawn for each independent variable and where necessary, interconnections are made between the diagrams.

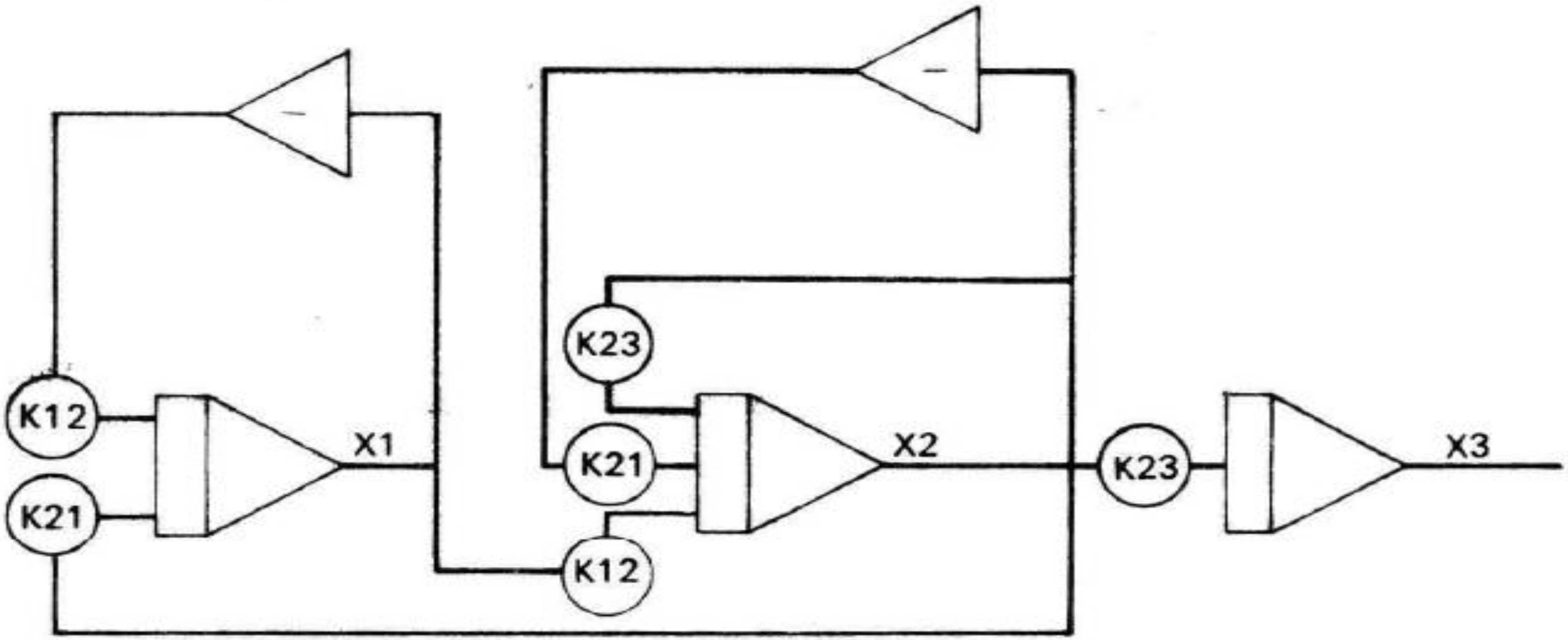


Figure: Analog Computer Model of the Liver

- There are three integrators, shown at the bottom of the figure. Reading from left to right, they solve the equations for x_1, x_2 and x_3.
- Interconnections between the three integrators, with sign changers where necessary, provide inputs that define the differential coefficients of the three variables.
- The first integrator, for example, is solving the equation

$$\dot{x}_1 = -k_{12}x_1 + k_{21}x_2$$

- The second integrator is solving the equation

$$\dot{x}_2 = -k_{21}x_2 + k_{12}x_1 + k_{23}x_2$$

$$\text{or, } \dot{x}_2 = k_{12}x_1 - (k_{21} - k_{23})x_2$$

- Similarly the third integrator is solving the equation

$$\dot{x}_3 = k_{23}x_2$$

Hybrid Simulation

- In case of hybrid simulation, the system is of neither a pure continuous nor a pure discrete in nature.
- For simulating such system, the combination of analog and digital computers are used. Such setup is known as hybrid computers.
- Hybrid computers are computers that exhibit features of analog computers and digital computers.
- The simulation provided by the hybrid computers is known as hybrid simulation.
- The term hybrid is reserved for the case in which functionally distinct analog and digital computers are linked together for the purpose of simulation.
- Hybrid computers can be used to simulate systems that are mainly continuous, but have some digital elements.
- One computer may be simulating the system being studied, while the other is providing the simulation of the environment where the system is to operate.

- The major difficulty in use of hybrid simulation is that it requires **high speed converters** to transform signals from analog to digital form and vice versa.
- High speed converters are required to transform signals from one form of representation to the other form.
- The availability of mini-computers has made hybrid simulation easier by lowering costs and allowing computers to be dedicated to an application.
- For Example: An artificial satellite for which both the continuous equations of motion and the digital signals need to be simulated.

Digital-Analog Simulators

- Digital Analog simulators indicates the use of programming languages in digital computer to simulate the continuous system.
- They allow a continuous model to be programmed on a digital computer.
- The language is composed of macro-instructions which are able to act as **adder**, **integrator** and **sign-changer**.
- A program is written to link these macro-instruction essentially in the same manner as operational amplifiers are connected in analog computers.
- More powerful techniques of applying digital computers to the simulation of continuous system have been developed.

Continuous System Simulation Languages(CSSLs)

- Continuous system simulation languages are high level programming languages which facilitate modelling and simulation of systems characterized by ordinary and partial differential equations.
- CSSLs help to model and study basically continuous systems formulated as block diagrams or in Ordinary Differential Equations(ODE).
- They allow a problem to be programmed directly from the equations of mathematical model rather than breaking those equations into functional elements.
- CSSLs can easily include macros and sub-routines that perform the function of specific analog elements.
- CSSLs include a variety of algebraic and logical expressions to describe the relations between variables.
- They, therefore, remove the orientation towards linear differential equations which characterizes analog methods.

Application Areas of CSSLs

1. **Vehicle Development:** Used in the fields of hydraulic systems(injection pumps, breaks), vehicle-ground interactions, dynamics of accident, etc.
2. **Missiles:** Can be used in autopilot mechanisms, flight systems, all kinds of control loops, etc.
3. **Peripheral System for Computers:** Can be used in electrical-mechanical interaction, diskdrives, pendrives, printers, etc.
4. **Environmental Analysis:** Can be used in the field of environmental analysis growth of plants, spread of harmful substances, etc.
5. **Chemical Processes:** Can be used in the study of diffusion process, in heat-exchangers, chemical-plants, etc.
6. **Electrical Supply:** Can be used in power plants, pumps, power distribution plants, control loops, etc.

CSMP III(Continuous System Modeling Program Version III)

- It is a program used for modeling continuous systems.
- A CSMP III program is constructed from three general types of statements:

a. Structure Statement

- Structure Statement are used to define the model.
- They consist of FORTRAN like statements, and functional blocks designed for operations that frequently occur in model definition.

b. Data Statements

- These statements are used to assign values to parameters, constants and initial conditions.

c. Control Statements

- These statements are used to specify options in the execution of the program and the choice of output.

- Structural Elements can make use of the operations of addition, subtraction, multiplication, division, exponential operations using the same notations and rules that are used in FORTRAN.
- For example, if the model includes the equation $X = \frac{6Y}{W} + (Z - 2)^2$ then following statement will le used:

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

- Note that real constants are specified in decimal notation. Exponent notation may also be used.
- Fixed value constants may also be declared.
- Also there are many functional blocks and functions that can be used such as exponential function, trigonometric functions and functions for taking maximum and minimum value.

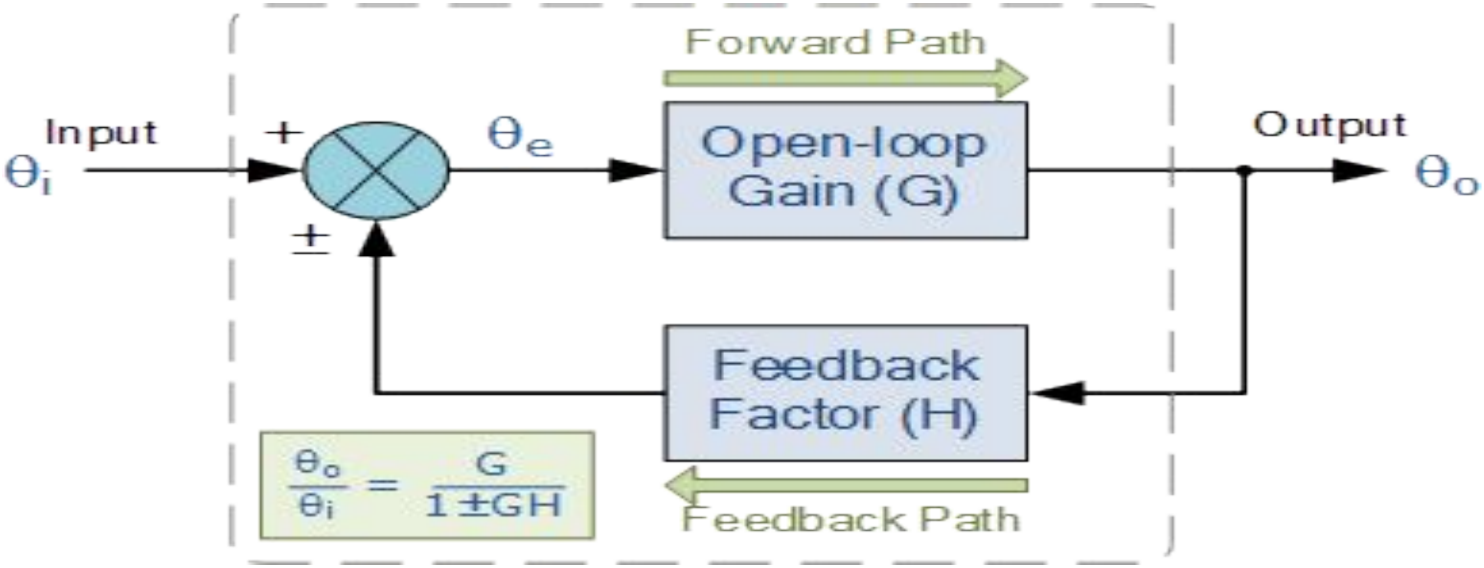
CSMP III Functional Blocks

General Form For CSMP III	Function
Y = INTGRL(IC,X)	
Y = LIMIT(P_1,P_2,X)	Used for finding limiting values. $Y = P_1$ for $X < P_1$ $Y = P_2$ for $X > P_2$ $Y = X$ for $P_1 \leq X \leq P_2$
Y = STEP(P)	Step Function $Y = 0$ for $t < P$ $Y = 1$ for $t \geq P$
Y = EXP(X)	
Y = ALOG(X)	For finding natural logarithm. $Y = \ln(X)$
Y = SIN(X)	Trigonometric Sine Function $Y = \sin(X)$

General Form For CSMP III	Function
Y = COS(X)	Trigonometric COSINE Function Y = cos (X)
Y = SQRT(X)	
Y = ABS(X)	For finding the absolute value Y = |X|
Y = AMAX1(X_1,X_2,…,X_n)	For finding the maximum value among the available values.
Y = AMIN1(X_1,X_2,…,X_n)	For finding the minimum value among the available values.

Feedback Systems

- Feedback system is one in which the output signal is sampled and then fed back to the input to form an error signal that drives the system.
- Feedback systems have a closed loop structure that bring results from past action of the system back to control future action.
- So feedback systems are influenced by their own past behavior.
- Feedback Systems are very useful and widely used in amplifier circuits, oscillators, process control systems as well as other types of electronic systems.

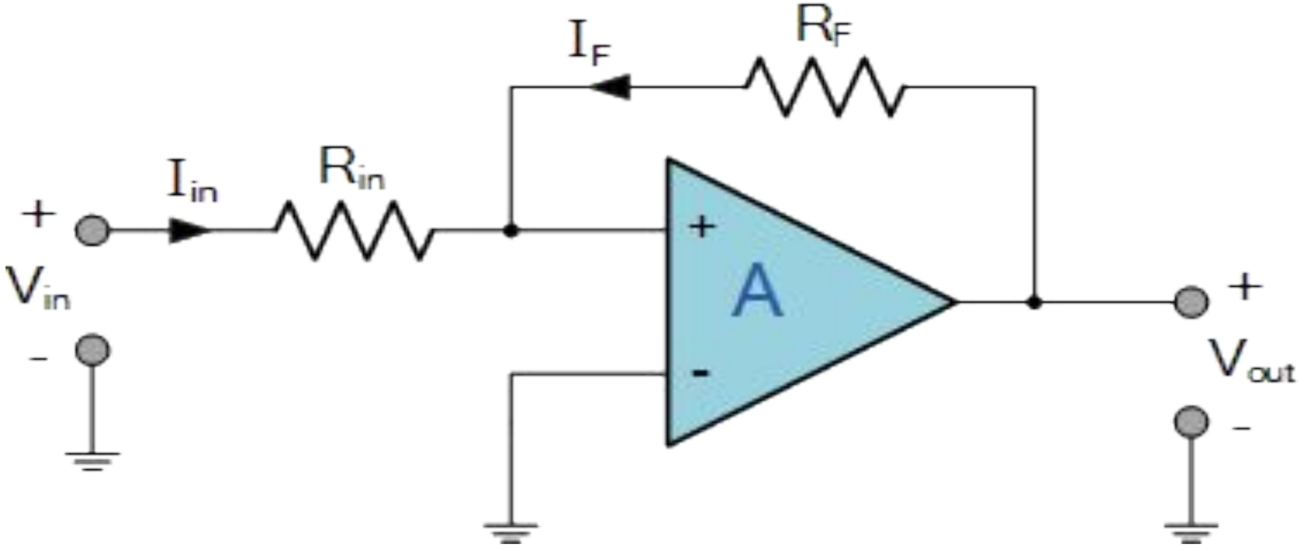


- A home heating system controlled by a thermostat is a simple example of a feedback system.
- The system has a furnace whose purpose is to heat a room and the output of the system can be measured as a room temperature.
- Depending upon whether the temperature is below or above the thermostat setting, the furnace will be turned on or off.
- There are two types of feedback systems:

1. Positive Feedback System
2. Negative Feedback System

1. Positive Feedback System

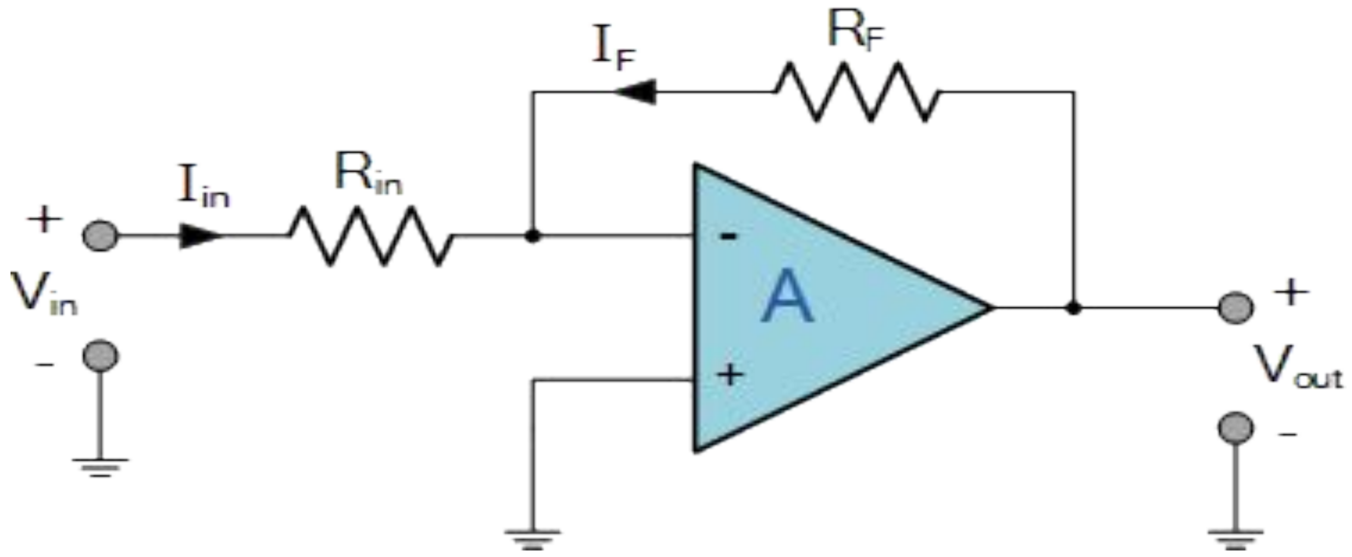
- In a positive feedback system the feedback is in-phase with the original input.
- The set point and output values are added together by the controller.
- The effect of positive (or regenerative) feedback is to "increase" the system gain, i.e, the overall gain with positive feedback applied will be greater than the gain without feedback.
- Positive feedback control of the op-amp is achieved by applying a small part of the output voltage signal at Vout back to the non-inverting (+) input terminal via the feedback resistor, R_F.



- Positive or regenerative feedback increases the gain and the possibility of instability in a system which may lead to self-oscillation and as such, positive feedback is widely used in oscillatory circuits such as Oscillators and Timing circuits.

2. Negative Feedback System

- In a negative feedback system the feedback is out-of-phase with the original input.
- The set point and output values are subtracted from each other by the controller.
- The effect of negative (or degenerative) feedback is to “reduce” the systems gain, i.e, the overall gain with negative feedback applied will be less than the gain without feedback.
- Negative feedback control of the op-amp is achieved by applying a small part of the output voltage signal at Vout back to the inverting (-) input terminal via the feedback resistor, R_F.



- The use of negative feedback in amplifier and process control systems is widespread because as a rule negative feedback systems are more stable than positive feedback systems.
- A negative feedback system is said to be stable if it does not oscillate by itself at any frequency except for a given circuit condition.
- Another advantage is that negative feedback also makes control systems more immune to random variations in component values and inputs.

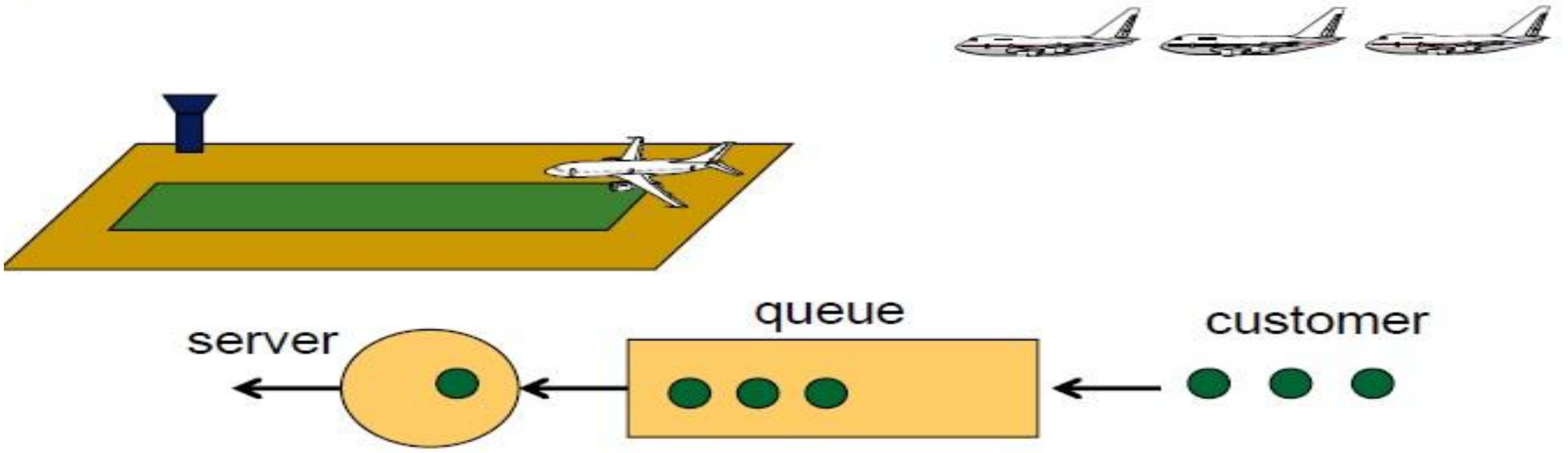
Positive vs Negative Feedback System

Positive Feedback System	Negative Feedback System
The feedback is in-phase with the original input.	The feedback is out-of-phase with the original input.
The set point and output values are added together by the controller.	The set point and output values are subtracted from each other by the controller.
The overall gain with positive feedback applied will be greater than the gain without feedback.	The overall gain with negative feedback applied will be less than the gain without feedback.
Positive feedback system are more oscillatory	Negative feedback systems are more stable than positive feedback systems.

Chapter 4
Queuing System

State Variables

- A state variable is one of the set of variables that are used to describe the state of a dynamical system.



- The state variables for above system can be:

1. InTheAir: Number of aircrafts either landing or waiting to land.
2. OnTheGround**:** Number of landed aircraft.
3. RunWayFree**:** Can be a Boolean value which is true if runway available.

Queuing System

- Waiting line queues are one of the most important areas, where the technique of simulation has been extensively employed.
- People at bank for service, 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 are a few examples of waiting lines.
- The problem with the queue is that if it is not managed properly, the customers should wait a long time.
- To prevent customer from being unsatisfied with the provided service, queuing system is managed. For eg: During cash withdraw in bank, you have to stay in queue and if it is not managed properly then you will surely be disappointed from the bank service even if that bank is one of the finest one in the city.

- The waiting situation arise because of any of the following reasons:

1. There is too much demand on the service facility so that the customers or entities have to wait for getting service.

2. There is too less demand, in which case the service facility have to wait for the entities.

- The main objective for the analysis of queuing situations is to balance the waiting time and idle time so as to balance the **waiting time** and **idle time**, so that the total cost will be minimized.

 Major elements in a Queuing System are:

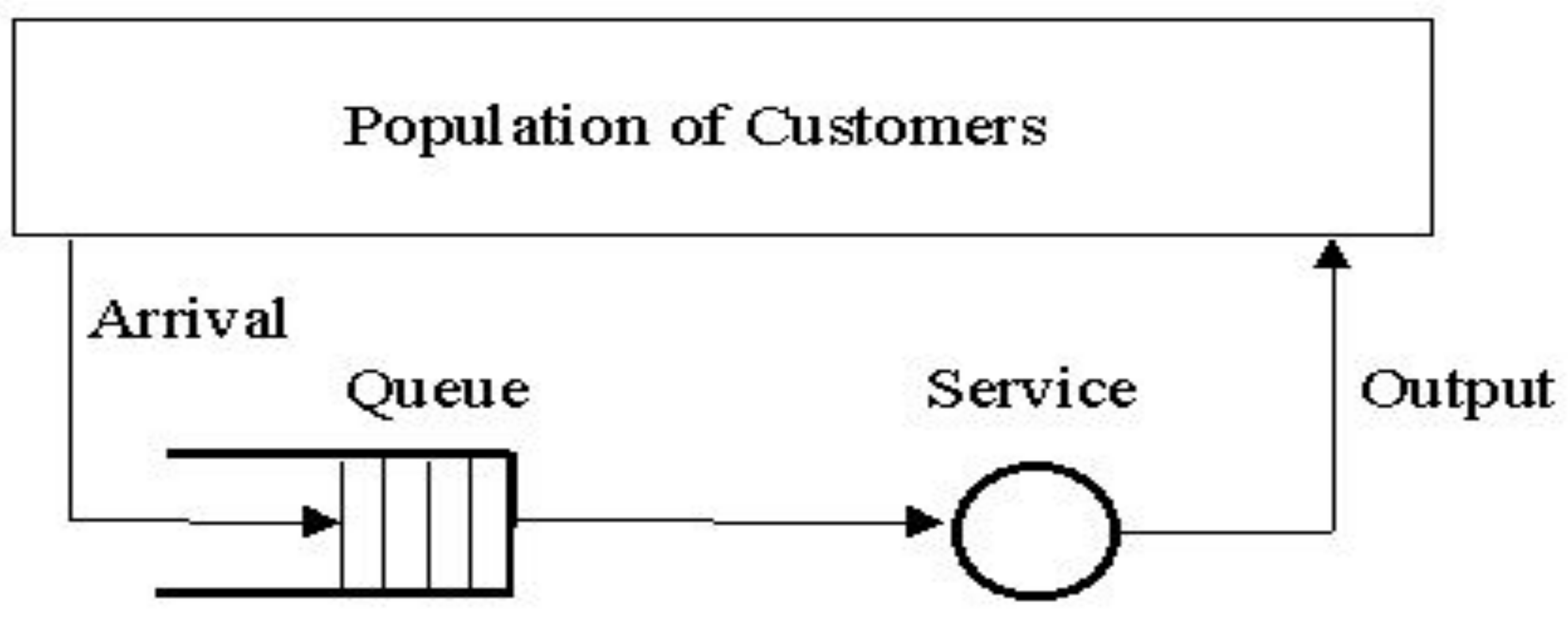


Figure 1

Elements of a Queuing System

1. Population of Customers or Calling Population

- The population of potential customers of the service is called calling population.
- Population of Customers or calling source can be considered either limited (**closed systems**) or unlimited(**open systems**).
- Unlimited population represents a theoretical model of systems with a large number of possible customers. The system of the restaurant or bank, a motorway petrol station and so on are considered to be open system i.e. with infinite calling population.
- Example of a limited population may be a number of processes to be run (served) by a computer or a certain processes to be run (served) by a computer, system of repairing certain number of machines by a service man, etc.
- The term customer must be taken in general which may be people, machines, computer processes, telephone calls,etc.

2. Arrival

- Arrival is defined as the way in which the customers arrives into the system.
- In most of the cases, the arrival of the customer is random.
- So the inter-arrival between two customers is described by a random distribution of interval known as arrival pattern.

3. Queue

- Queue represents the number of customers that have entered into the system and are waiting for the service.
- Maximum Queue Size (also called System capacity) is the maximum number of customers that may wait in the queue.
- The two main properties of queue are as follows:

  a) Maximum Size:

  - Queue, in practice, is always limited.
  - Maximum size represents the maximum number of customers that can accommodate in the queue.

b) Queue Discipline:

- Queue discipline represents the rules in which the customers are inserted or removed to or from the queue.
- It can be organized in various ways like FIFO, LIFO, Serve In Random Order(SIRO), Priority Queue, etc.

4. Service Time

- Service time represents the time needed to provide service to a customer by a server.
- Service time may be of constant duration or of random duration.

5. Number of servers

- Servers represent the entity that provides service to the customer.
- A system may consist of single server or multiple servers.
- A system with multiple servers is able to provide parallel services to the customers.

6. Output

- Output represents the way customers leave the system.
- Output is mostly ignored by theoretical models, but sometimes the customers leaving the server enter the queue again.

Applications of Queuing Theory

1. Telecommunication
2. Traffic control
3. Determining the sequence of computer operations
4. Predicting computer performance
5. Health services (e.g. control of hospital bed)
6. Airport traffic
7. Airline ticket sales
8. Layout of manufacturing systems.

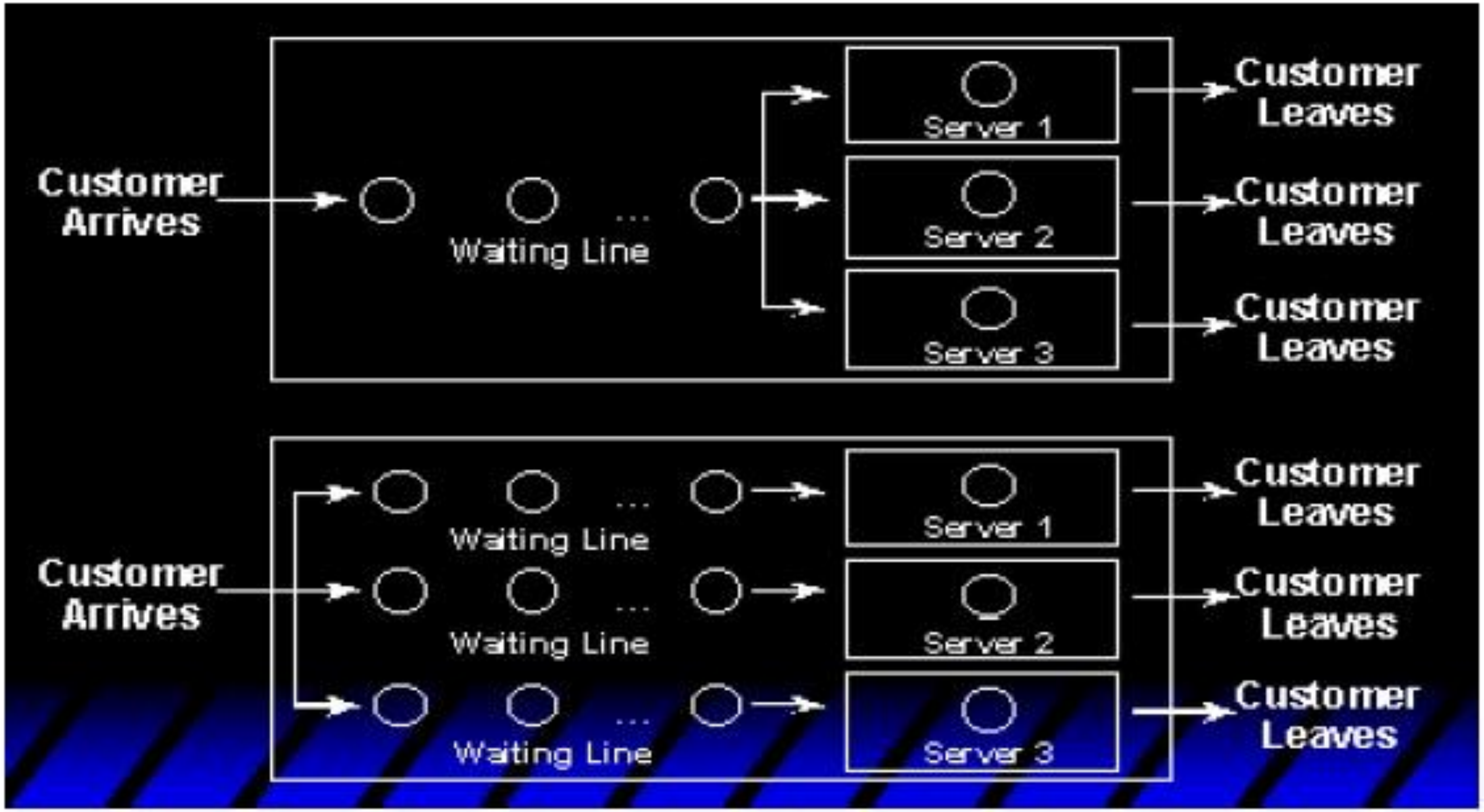


Figure: Example of Application of Queuing Theory

Examples of Some Real World Queuing System

1. **Commercial Queuing Systems:** Commercial organizations serving external customers. For example Queuing Systems in Dental Service, Bank, Garage, Gas stations, etc.
2. **Transportation Queuing Systems:** Queuing System for vehicles waiting at toll stations and traffic lights, trucks or ships waiting to be loaded, taxi cabs, etc.
3. **Business-Internal Service Systems:** Queuing systems at Inspection Stations, conveyor belts, computer support, etc.
4. **Social Service Systems:** Queuing systems at Judicial process, hospitals, waiting list for organ transplant, etc.

Characteristics of Queuing System

1. **Arrival Process:** It is a distribution that determines how the tasks arrive in a system.
2. **Service Process:** It is a distribution that determines the task processing time.
3. **Number of servers:** It is the total number of servers available to process the tasks.
4. **Queuing Discipline:**

- It is the discipline that represents the way the queue is organized.
- Queuing Discipline is the rule for inserting or removing customers to or from the queue.
- There are various discipline for inserting and removing customers to and from queue.

a. FIFO(First In First Out)

- Also called as First Come First Serve(FCFS).
- The customer that enters the queue first will be served first.

b. LIFO(Last In First Out)

- Also called as Last Come First Serve(LCFS).
- The customer that enters the queue last will be served first.

c. SIRO(Serve In Random Order)

- The customer are served in random fashion.

d. Priority Queue

- It can be viewed as a queue with various priority.
- The customer are served as per the priorities.

e. Many other more complex queuing methods that typically change the customer's position in the queue according to the time spent already in the queue, expected service duration, and/or priority.

5. Number of customers: It is the number of customers waiting to be served.

Continue of Queuing System

- Most quantitative parameters (like average queue length, average time spent in the system) do not depend on the queuing discipline.
- That's why most models either do not take the queuing discipline into account at all or assume the normal FIFO queue.
- In fact the only parameter that depends on the queuing discipline is the variance (or standard deviation) of the waiting time. There is this important rule since it is used to verify results of a simulation experiment.
- The two extreme values of the waiting time variance are for the FIFO queue (minimum) and the LIFO queue (maximum).
- Theoretical models (without priorities) assume only one queue. This is not considered as a limiting factor because practical systems with more queues (bank with several tellers with separate queues) may be viewed as a system with one queue, because the customers always select the shortest queue.
- Of course, it is assumed that the customers leave after being served.

- Systems with more queues (and more servers) where the customers may be served more times are called **Queuing Networks**.
- **Service:** Service represents some activity that takes time and that the customers are waiting for. It may be a real service carried on persons or machines, but it may be a CPU time slice, connection created for a telephone call, being shot down for an enemy plane, etc. Typically a service takes random time.
- **Service Pattern:** Theoretical models are based on random distribution of service duration also called Service Pattern.
- Another important parameter is the number of servers. Systems with one server only are called **Single Channel Systems** whereas systems with more servers are called **Multi Channel Systems.**

Queuing Theory

- Queuing Theory is a collection of mathematical models of various queuing systems that take inputs parameters and provide quantitative parameters describing the system performance.
- It is the mathematical study of waiting lines or queues.
- Queuing Theory refers to the mathematical models used to simulate these queues.

- Many systems (especially queuing networks) are not soluble at all, so the only technique that may be applied is simulation.

-Nevertheless queuing systems are practically very important because of the typical trade-off between the various costs of providing service and the costs associated with waiting for the service (or leaving the system without being served).

-High quality fast service is expensive, but costs caused by customers waiting in the queue are minimum.

-On the other hand long queues may cost a lot because customers (machines e.g.) do not work while waiting in the queue or customers leave because of long queues.

-So a typical problem is to find an optimum system configuration (e.g. the optimum number of servers).

-The solution may be found by applying queuing theory or by simulation .

Types of Queuing System

1. Single Line with Single Server Queuing System

- There is a single line of customers to be served which is served by a single server.

2. Single Line with Multiple Server Queuing System

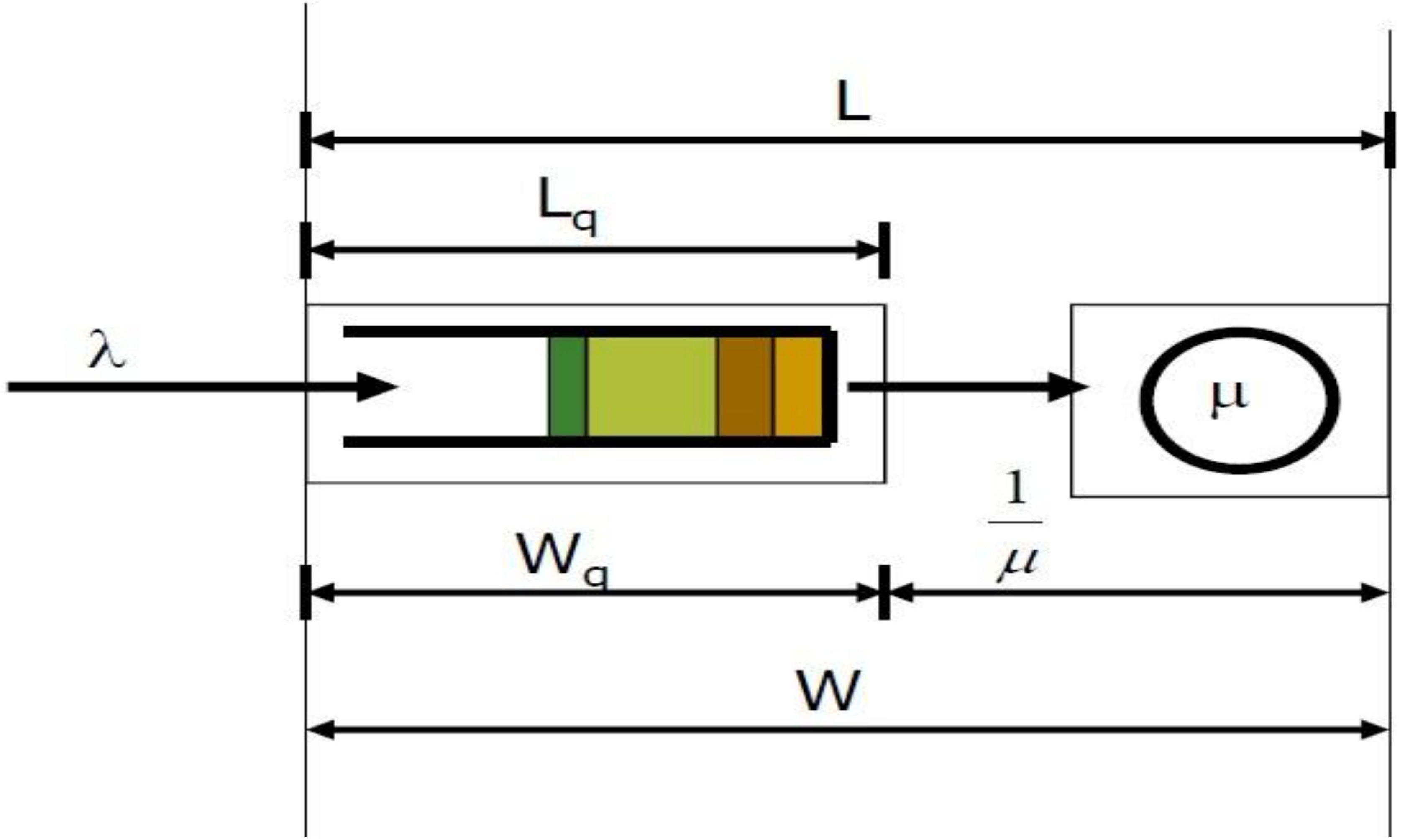
- There is a single line of customers to be served which is served by multiple or more than one server.

3. Multiple Line with Multiple Server Queuing System

- There are multiple or more than one line of customers to be served which is served by There is a single line of customers to be served which is served by a single server.

Queuing Model

M/M/1 Queuing Model



 In the above figure:

1. λ represents arrival rate of jobs
2. μ represents service rate of server
3. L represents length of the queuing system
4. Lq represents length of queue
5. W represents average waiting time in the whole system
6. Wq represents average waiting time in the queue

- M/M/1 Queue is the most widely used queue.
- This queue is used to model single processor systems or individual systems in a computer system.
- In this queue model following assumptions are made:

  -Inter-arrival rate is exponentially distributed

  - Service rate of server is exponentially distributed

  - Contains a single server

  -Follows FCFS Discipline

  - Unlimited queue length is allowed

  - Infinite number of customers

Kendall Notation

- Also called as Queuing Notation
- Six parameters are used.
- The basic format of this notation is of form: A / B / c / D / N / K

  - A represents the inter-arrival time distribution.
  - B represents the service time distribution.
  - c represents the number of parallel servers.
  - D represents the queue or service discipline.
  - N represents the maximum size of queue.
  - K represents the size of the calling population or the population size .

- The symbols used for A and B are :

  a. M is the Poisson (Markovian) process

  - If arrival time is Poisson Distribution, then the inter-arrival time is exponential distribution

Poisson Distribution	Exponential Distribution
Number of events in a time interval	Time between two events.
Discrete	Continuous on an interval

  - So M here denotes Exponential Inter-Arrival Time Distribution

  b. D is the symbol for deterministic (known) arrivals and constant service duration/ Deterministic Service Time Distribution

  c. E_k represents Erlang distribution of intervals or service duration

  d. G is Arbitrary or general distribution

  e. GI is a general (any) distribution with independent random values

  f. PH (Phase type)

- If arrival time is Poisson Distribution, then the inter-arrival time is exponential distribution.

- The Kendall classification of queuing systems exists in several modifications.
- Another form is 1/2/3(/4/5/6) where:
  - 1 indicates the inter-arrival time distribution
  - 2 indicates the service time distribution
  - 3 indicates the number of servers
  - 4 indicates the maximum size of queue.
  - 5 indicates the size of the calling population or the population size
  - 6 indicates the queue or service discipline

Kendall Classification of Queuing System Examples:

1. D/M/1

- The provided notation indicates that the system has a single server with Deterministic inter-arrival time distribution and Exponential service time distribution.
- The system has unlimited population and unspecified queuing discipline.

2. M/G/3/20

- The notation indicates that the system has 3 servers with Exponential inter-arrival time distribution and has General service distribution.
- It has a maximum queue size of 20 customers and unlimited customer population can be served.

3. D/M/1/LIFO/20/510

- The notation indicates that the system has a single server with Deterministic inter-arrival time distribution and Exponential service time distribution.
- It has a maximum queue size of 20 customers and the queue follows LIFO discipline. Total of 510 customers can be served.

4. M/M/3/20/1500/FCFS

- The notation indicates that the system has a 3 servers with Exponential inter-arrival time distribution and Exponential service time distribution.
- It has a maximum queue size of 20 customers and the queue follows FCFS(FIFO) discipline. Total of 1500 customers are served.

Network of Queues

- Queuing Network are the systems in which single queues are connected by routing network.
- Network of queues are used to model queuing when a set of resources is shared.
- Such a network can be modeled by a set of service centers where each service center may contain one or more servers.
- In the study of queue networks, one queue typically tries to obtain the distribution of the network.
- In network of queues, when a customer is connected to one node it can join another node or queue for service or can leave the network.
- For a network of m-nodes, the states of the system can be described by m-dimensional vector.
- Network of queues is widely used in the field of telecommunication, computer network, etc.

Measurement of System Performance

- It is the analysis and measurement of how well the queuing system performs.
- The various parameters used for measuring the system performance are:

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

- 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.

2. Number of servers

3. Average waiting time of the customers in the queue or in the system.

- The knowledge of average waiting time in the queue is necessary for determining the cost of waiting in the queue.

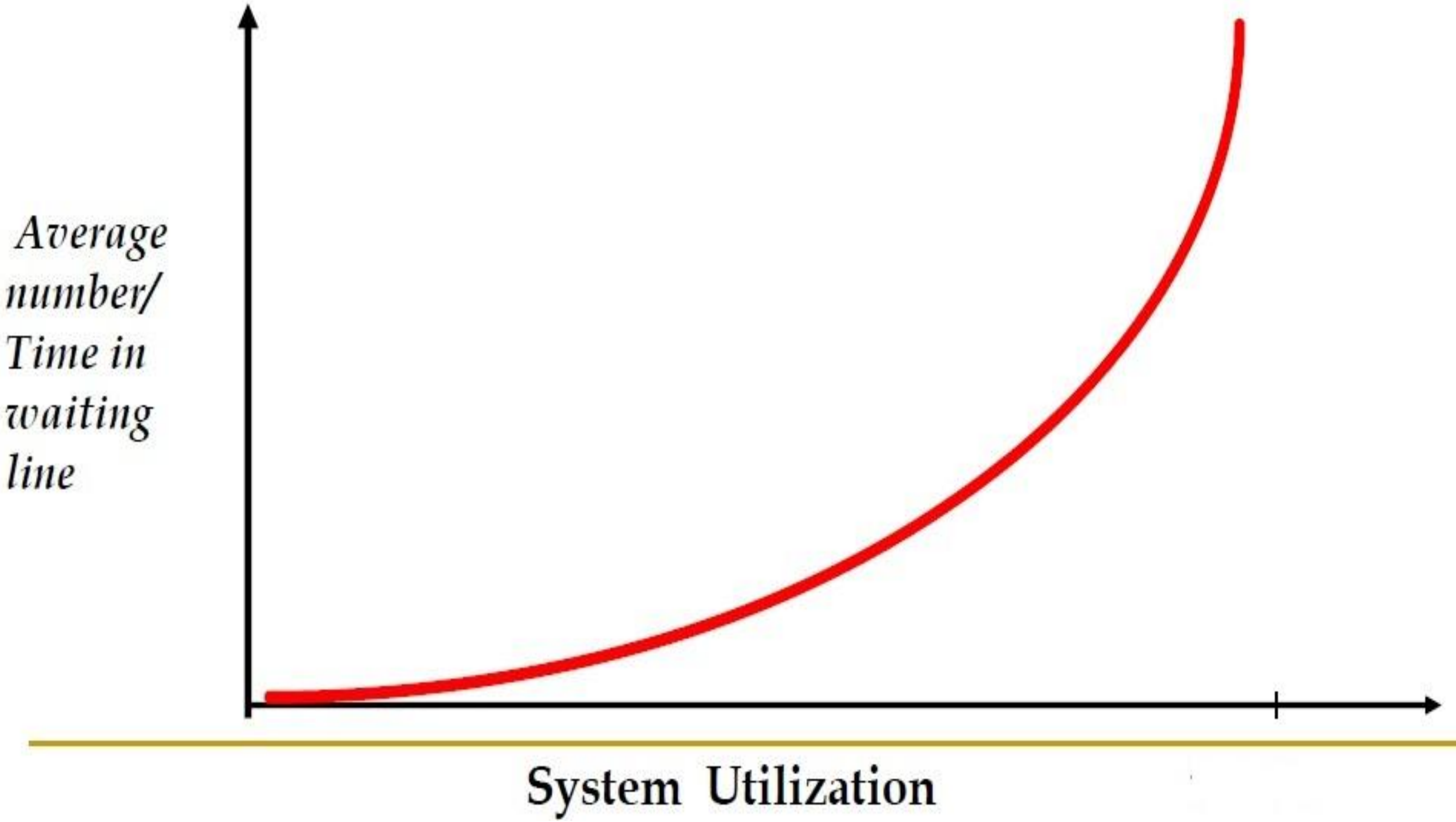
4. Length/Size of queue
5. The cost of waiting/idle time
6. System/Server Utilization

System/Server Utilization of Queuing System

- System/Server Utilization is defined as the extent to which a system/server is busy rather than idle.
- It is defined as the percentage of time during which system/server is busy processing jobs during system.
- System utilization factor denoted by S is the ratio of average arrival rate to the average service rate (μ).

$$S = \frac{\lambda}{\mu}$$

- In the case of n-server model, $S = \frac{\lambda}{n\mu}$
- Under the normal situations, 100% system utilization is not a realistic goal.
- 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 figure below.



Average number/ Time in waiting line
System Utilization

Time Oriented Simulation

A factory has large number of semi automatic machines. On 50% of the working days none of the machines fail. On 30%of the days one machines 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 remaining 5% in three days.

Simulate the system for 30 days duration and estimate the average length of queue, average waiting time and server loading that is the fraction of time for which server is busy.

- 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.
- According to the scenario given, arrival pattern of machine is:

Arrival pattern:

On 50%of the days arrival=0

On 30%of the days arrival=1

On 20%of the days arrival=2

- So the expected arrival rate can be calculated as:

Expected Arrival Rate = 0*0.5+0.3*1+0.2*2

= 0.7 per day

- Again according question, Service pattern is:

65% machines in 1 day

30% machines in 2 days

5% machines in 3 days

- So the expected service time = 0.65*1+0.3*2+0.05*3

  = 1.4 days

- Hence the expected service rate is = $\frac{1}{1.4} = 0.714$ 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=1day

0.65<r<=0.95 ST=2days

0.95<r<=1.0 ST=3 days

- 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 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.

- At start of the simulation, the system that is the maintenance facility can 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, the 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.

 The following statistics can be determined.

Machine failures(arrivals) during 30 days=21

Arrivals per day=21/30=0.7

Waiting time of customer=40 days

Waiting time per customer=40/21=1.9 days

Average length of the queue=1.9

Server idle time=4 days=4/30* 100=13.33 %

Server loading=(30-4)/30=0.87

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	70
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	.86	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	3 4
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				

Important Formula for Numerical

1. System/Server Utilization(S) $= \frac{average\ arrival\ rate\ (\lambda)}{average\ service\ rate\ (\mu)}$
2. Fraction time Busy = S
3. Fraction Time idle = 1-S
4. Average Waiting Time $= \frac{S}{\mu-\lambda}$
5. Average Number of Customers in System(N) $= \frac{Fraction\ Time\ Busy}{Fraction\ Time\ Idle} = \frac{S}{1-S}$
6. Average Time Customer Spends in time(T) $= \frac{N}{\lambda}$
7. Probability of Zero Customer(P_0) = 1 - S
8. Probability of n Customer(P_n) = $S^n P_0$ for n>0

Numerical 1

Consider a database system with an average service time of 450 msec. As database requests are initiated by large number of clients, a random arrival pattern may be assumed. Thus the arrival process is assumed to be Poisson. On the average, a new database query arrives every 500 msec. Service time are assumed to be exponentially distributed, the queuing discipline is assumed to follow FCFS pattern. Calculate:

1. System Utilization
2. Fraction time busy
3. Fraction time idle
4. Average Waiting time
5. Average Number of customers in system
6. Average time customers spends in the system

Numerical 2

In a petrol pump, Customer arrival time is given by Poisson Distribution with an arrival rate of 2 customer/hour and they get exponentially served at the rate of 3 customer/hour. Find:

1. System Utilization
2. Probability of Zero Customer
3. Probability of 1 Customer
4. Probability of 4 or more Customers
5. Average Waiting time
6. Average Number of customers in system
7. Average time customers spends in the system

Hint for No. 4

Probability of 4 or more customers = 1 – probability of zero customers – probability of one customer – probability of two customers – probability of three customers

i.e. $P_{cust>=4} = 1 – P_{cust=0} – P_{cust=1} – P_{cust=2} - P_{cust=3}$

Chapter 5
Markov Chains

Markov Process

- Markov process is a process whose future probabilities are determined by its most recent values.
- If the future states of a process are **independent of the past and depend only on the present** , the process is called a Markov process.
- Markov process is a simple **stochastic process** in which the distribution of future states depends only on the present state and not on how it arrived in the present state.
- Markov process models are useful in studying the evolution of systems over repeated trails or sequence time periods or stages.
- Examples of Markov Process:

1. Airplane at Airport
2. Rainfall
3. Behavior of Business or Economy
4. Flow of traffic

Markov Chain

- A Markov chain is a stochastic model describing a sequence of possible events in which the probability of next state depends only on the previous event.
- A discrete state Markov process is called a Markov chain.
- A Markov chain is a probabilistic model describing a system that changes from state to state, and in which the probability of the system being in a certain state at a certain time step depends only on the state of the preceding time step
- Since the system changes randomly , it is generally impossible to predict the exact state of the system in the future.
- However, the statistical properties of the system's future can be predicted..
- Markov chains is a mathematical tools for statistical modeling in modern applied

mathematics, information science.

- M/M/m queues can be modeled using Markov processes. The <u>time spent by the job in such a queue is Markov process</u> and <u>the number of jobs in the queue is a Markov chain</u>.
- Markov chains are used to compute the probabilities of events occurring by viewing them as states transitioning into other states, or transitioning into the same state as before.
- Markov chains are used to analyze trends and predict the future such as weather forecasting, stock market prediction, genetics, product success, etc.
- A Markov chain consists of states and transition probabilities. Each transition probability is the probability of moving from one state to another in one step.
- The probability that j is the next event of the chain given that the current state is i is called the transition probability from i to j.
- These transition probabilities are independent of the past and depend only on the two states involved.

- The Markov chain has network structure much like that of website, where each node in the network is called a state and to each link in the network a transition probability is attached, which denotes the probability of moving from the source state of the link to its destination state.
- If at any time the system is in state i, then with probability equal to the transition probability from state i to state j, it moves to state j.
- We will make an assumption, called Markov property, according to which the probability of moving from source page to a destination page doesn’t depend on the route taken to reach.

- If the transition probability does not depend on the time n, we have a stationary Markov chain, with transition probabilities

$$P_{ij} = P(X_{n+1} = j | X_n = i)$$

i.e. $P_{ij} = Pr(X_1 = j \mid X_0 = i)$

- The probability of going from state i to state j in n time steps is

$$P_{ij}^{(n)} = Pr(X_n = j \mid X_0 = i)$$

Transition Matrix

- The matrix of transition probabilities is called the transition matrix.
- It is a square matrix and is represented by P.
- The transition probability matrix is the matrix that shows probabilities of moving from one state to another state.
- So the transition matrix for whole Markov chain can be represented as:

$$\mathbf{P} = \begin{bmatrix} P_{11} & P_{12} & \dots & P_{1n} \\ P_{21} & P_{22} & \dots & P_{2n} \\ \vdots & \vdots & \ddots & \\ P_{n1} & P_{n2} & & P_{nn} \end{bmatrix}$$

where P_{11}, P_{12},….P_{nn} are the transition probabilities.

- All the entries of the matrix lie between 0 and 1. The sum of entries of any row is equal to 1.

	Nice	Rainy	Snowy
Nice	0.0	0.75	0.25
Rainy	0.25	0.25	0.5
Snowy	0.25	0.5	0.25

Key Features of Markov Chain

 A sequence of trials of an experiment is a Markov chain if:

1. The outcome of each experiment is one of a set of discrete states.
2. The outcome of an experiment depends only on the present state and not on any past states.
3. The transition probability remain constant from one transition to the next.

Markov Process

- Markov process is a process whose future probabilities are determined by its most recent values.
- According to **Markov Property**, the state of the system at time t+1 depends only on the state of the system at time t.
- Also a stationary assumption is made according to which transition probabilities are independent of time(t). So,

$$Pr[X_{t+1} = b \mid X_t = a] = P_{ab}$$

Current Status Distribution Matrix

- Current status distribution matrix is a row matrix that provides the status of current state of all the discrete states of the Markov chain.
- It is denoted by Q_0.
- Each entry must be between 0 and 1 inclusive.
- The sum of entries of each row must be 1.

Scenario

Given that chance of a Honda bike user to buy Honda bike at next purchase is 70% and that his next purchase will be Yamaha is 30%. The chance of Yamaha bike user to buy Yamaha bike at next purchase is 80% and that his next purchase will be Honda is 20%. What is the probability to buy Yamaha bike after three purchase of a current Honda Bike user?

	Honda User	Yamaha User
Current Status Matrix Q_0 =	1	0

Steps on making predictions

1. Create current status distribution matrix Q_o.
2. Create probability distribution matrix or transition matrix P.
3. Calculate $Q_n = Q_o * P^n$, which represents probability vector after n repetitions of the experiment.

Application of Markov Chain

1. Internet Application

Link Analysis

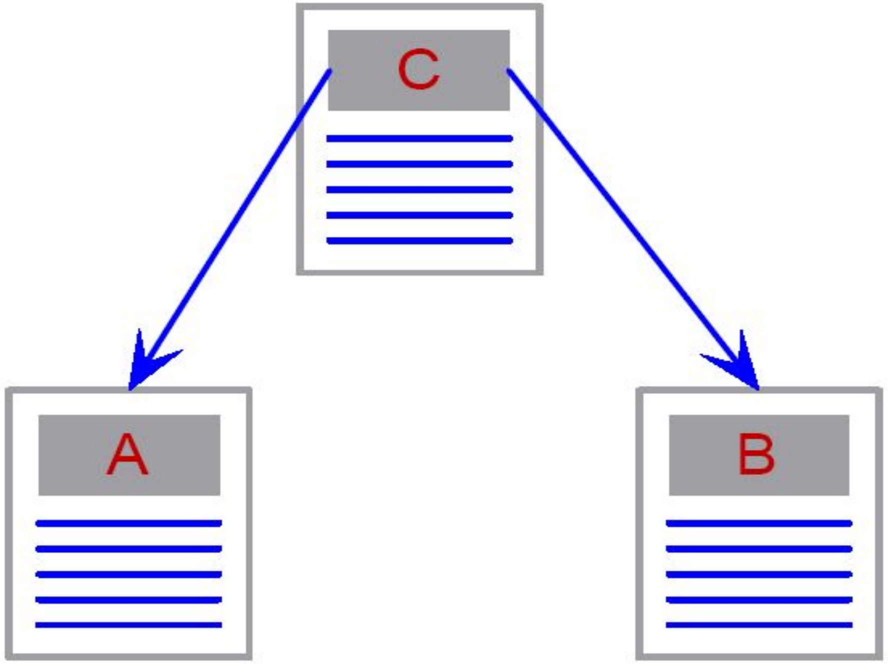
- Link analysis is a data-analysis technique used to evaluate relationships (connections) between nodes.
- Relationships may be identified among various types of nodes (objects), including organizations, people and transactions.
- A link from page A to page B is a vote of the author of A for B, or a recommendation of the page.
- The number of incoming links to a page is a measure of importance and authority of the page.
- A page is more important if the sources of its incoming links are important.

Why link analysis?

- The web is not just a collection of documents – its hyperlinks are important!
- A link from page A to page B may indicate:
  - A is related to B, or
  - A is recommending, citing, voting for or endorsing B
- Links are either
  - referential – click here and get back home, or
  - Informational – click here to get more detail
- Links effect the ranking of web pages and thus have commercial value.
- Link analysis has been used for:
  - investigation of criminal activity (fraud detection, counterterrorism, and intelligence)
  - computer security analysis
  - search engine optimization
  - market research
  - medical research

Citation Analysis – Impact Factor

- The **impact factor** of a journal = $\frac{A}{B}$
  - A is the number of current year citations to articles appearing in the journal during previous two years.
  - B is the number of articles published in the journal during previous two years.
- Co-citation: A and B are co-cited by C, implying that they are related or associated.
- The strength of co-citation between A and B is the number of times they are co-cited.



Page Rank

- The PageRank of a webpage as used by Google is defined by a Markov chain.
- Google's PageRank (PR) is method of ranking web pages for placement on a Search Engine Results Page (SERP).
- PageRank is a mathematical formula (algorithm) that Google uses to calculate the importance of a particular web page/URL based on incoming links.
- PageRank algorithm assigns each web page a relevancy score.
- It is used to measure the relative importance of a website within it's set of hyperlinked pages.
- If we rank better in organic search, then we should get more website traffic from search engines.

- Markov models can be used to make predictions regarding future navigation and to personalize the web page for an individual user.
- Markov models have also been used to analyze web navigation behavior of users.
- **Rank Sink:** A page or group of pages is a rank sink if they can receive rank propagation from their parent but can’t propagate rank to other pages. A rank sink occurs when a page does not link out.
- **Dangling pages**: Dangling pages are pages which do not have any out link or the page which not provide reference to other pages

The Page Rank Algorithm

 The original Page Rank algorithm was described by Lawrence Page and Sergey Brin in several publications. It is given by

PR(A) = (1-d) + d (PR(T1)/C(T1) + ... + PR(Tn)/C(Tn))

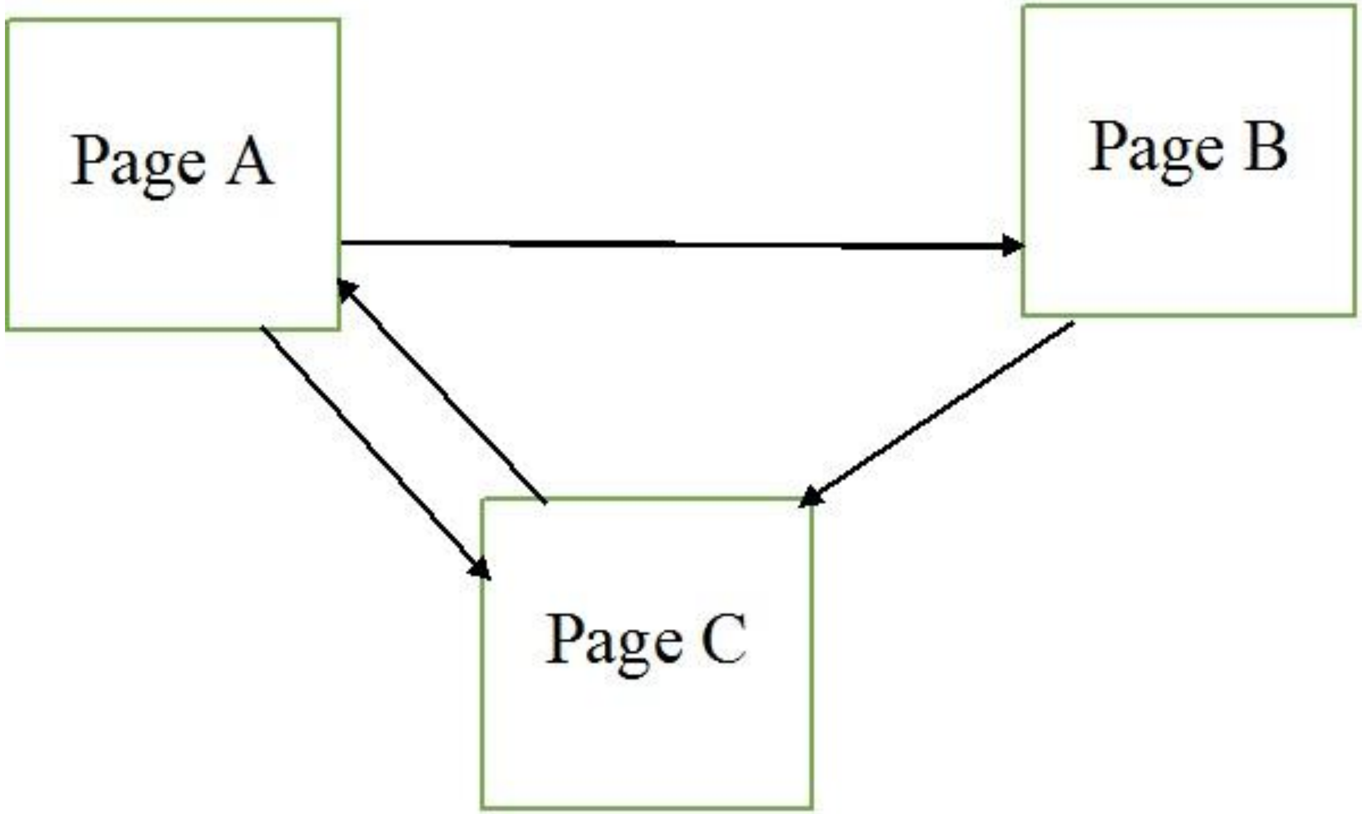
Where

- PR(A) is the Page Rank of page A,
- PR(Ti) is the Page Rank of pages Ti which link to page A,
- C(Ti) is the number of outbound links on page Ti and
- d is a damping factor which can be set between 0 and 1.

Page Rank Computation

We regard a small web consisting of three pages A, B and C, whereby page A links to the pages B and C, page B links to page C and page C links to page A. According to Page and Brin, the damping factor d is usually set to 0.85, but to keep the calculation simple we set it to 0.5. The exact value of the damping factor d admittedly has effects on Page Rank, but it does not influence the fundamental principles of Page Rank.

Initially let the page rank of each page be 1. Calculate iteratively and conclude which page has the highest score.


Page A
Page B
Page C

- Using formula for page rank

  PR(A) = 0.5 + 0.5 PR(C)/1
  PR(B) = 0.5 + 0.5 (PR(A) / 2)
  PR(C) = 0.5 + 0.5 (PR(A) / 2 + PR(B))

- These equations can easily be solved. We get the following Page Rank values for the single pages:

PR(A) = 14/13 = 1.07692308
PR(B) = 10/13 = 0.76923077
PR(C) = 15/13 = 1.15384615

- It is obvious that the sum of all page's Page Ranks is 3 and thus equals the total number of web pages.

2. **Market Research and Market Trend Prediction:** Markov chains and their respective diagrams can be used to model the probabilities of certain financial market climates and thus predicting the likelihood of future market conditions.

3. **Asset pricing and other financial predictions:** Markov chain and Markov process can be used to predict the price and financial factors of certain assets.

4. **Markov text generator:** Can be used in automatic text generation. A Markov chain algorithm basically determines the next most probable suffix word for a given prefix.

5. **Population Genetics:** Markov chain models have been the most widely used ones in the study of random fluctuations in the genetic compositions of populations over generations.

Numericals

Spring time has 3 possible conditions nice, rainy and snowy. If its nice today then tomorrow it will be:

a. rainy 75% of the time
b. snowy 25% of the time

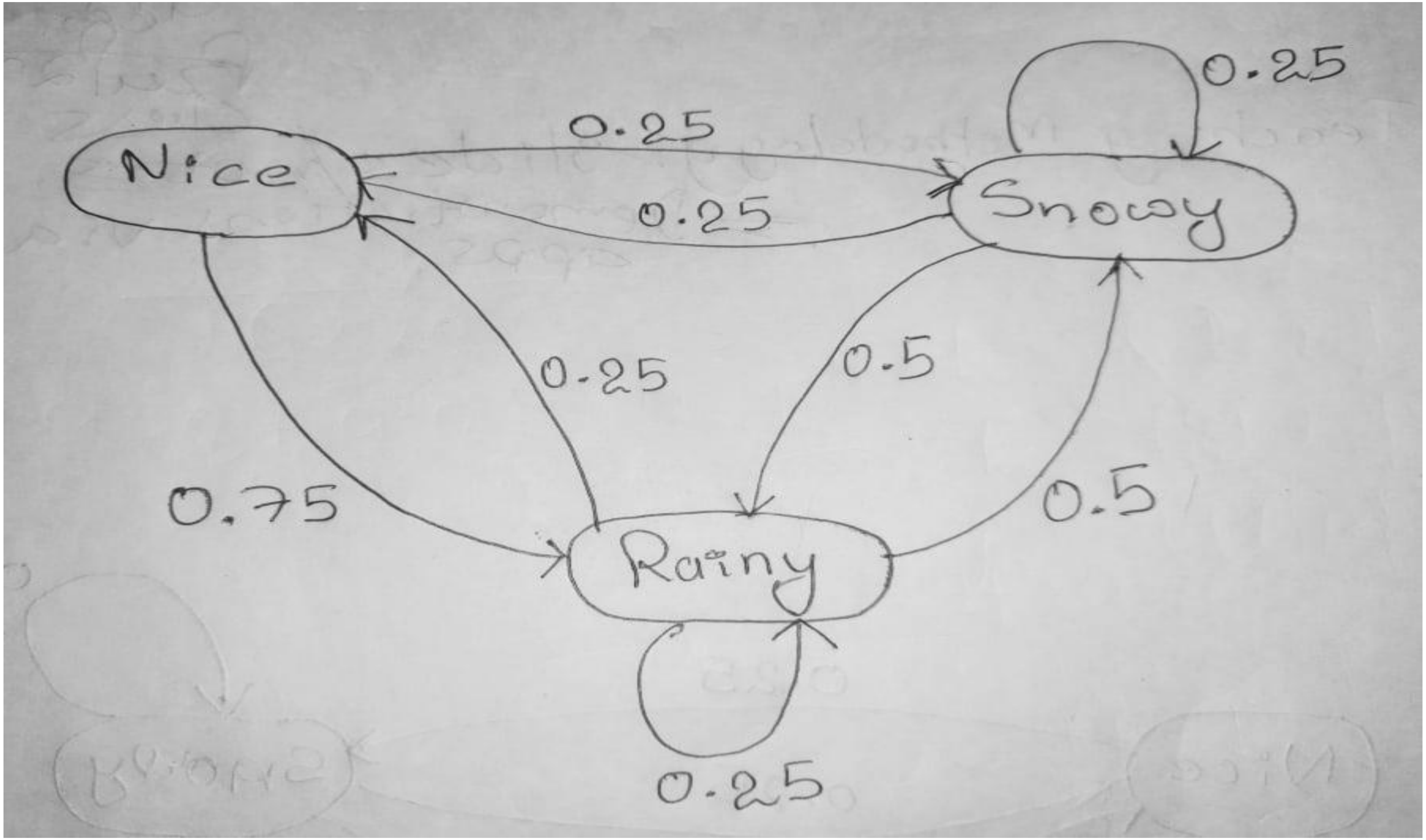
If its rainy today then tomorrow it will be:

a. rainy 25% of the time
b. nice 25% of the time
c. snowy 50%of the time

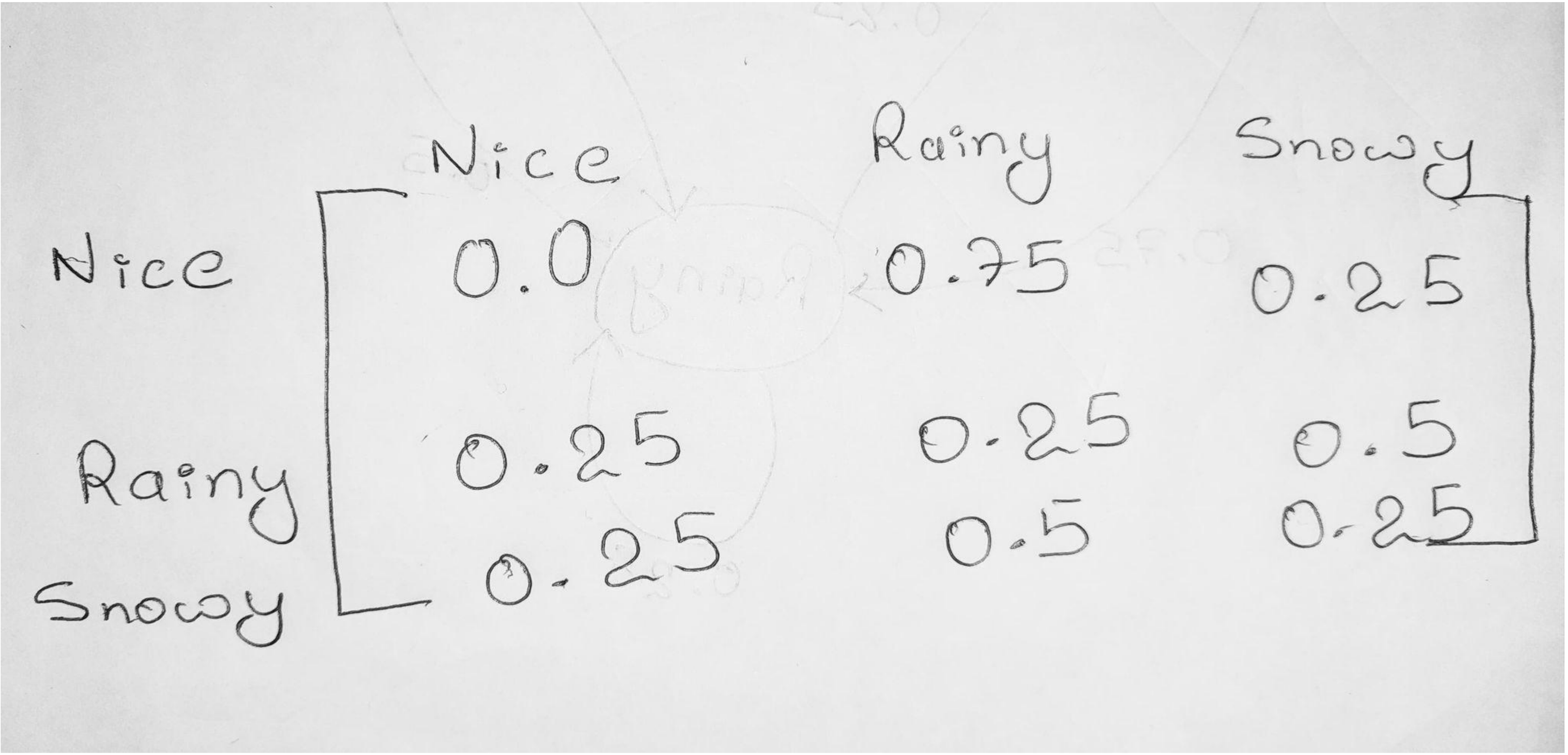
If its snowy today then tomorrow it will be:

a. rainy 50% of the time
b. nice 25% of the time
c. snowy 25% of the time

Make graph or stochastic FSM of above and construct Transition matrix.



Stochastic FSM



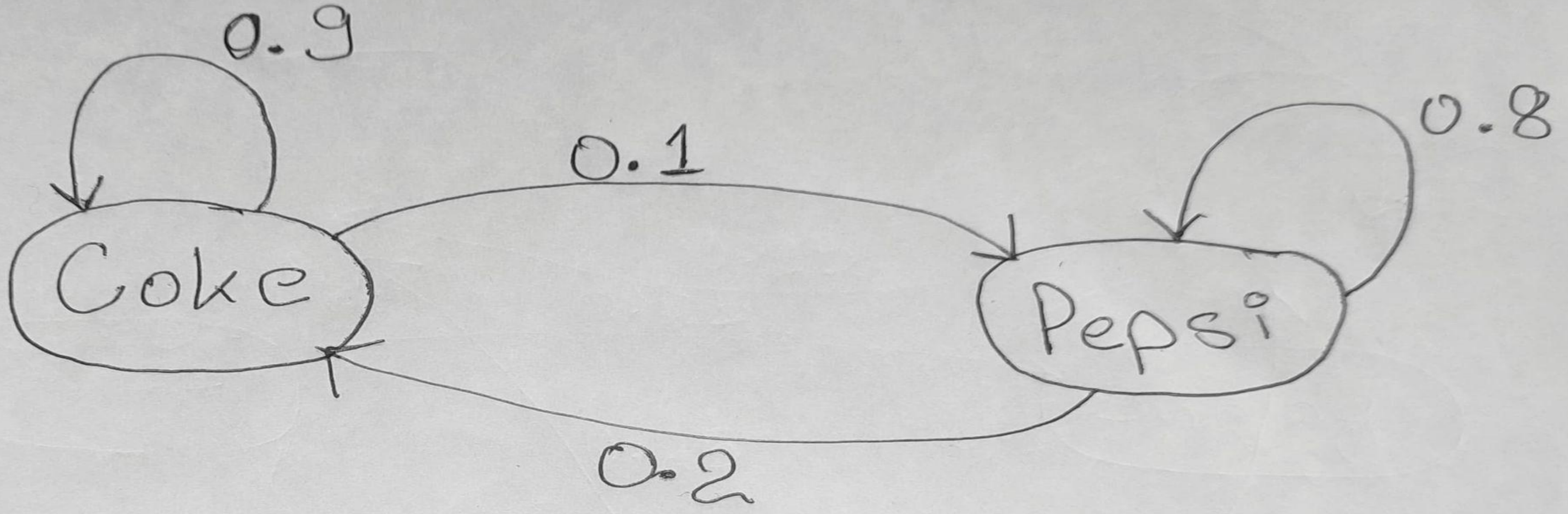
Transition Matrix

Numerical

Given that a person's last coca-cola purchase was coke, there is a 90% chance that his next cola purchase will also be coke. If a person's last cola purchase was pepsi, there is an 80% chance that his next cola purchase will also be pepsi.

a. Make graph of above problem and construct a transition matrix.
b. Given that a person is currently Pepsi purchaser, what is the probability that he will purchase Coke two purchases from now?
c. Given that a person is currently a Coke purchaser, what is the probability that he will purchase pepsi three purchase from now?
d. Assume each person makes one cola purchase per week. Suppose 60% of all people now drink coke and 40% drink pepsi, what fraction of people will be drinking coke three weeks from now?

a.



Transition Matrise

	Coke	Pepsi
Coke	0.9	0.1
Pepsi	0.2	0.8

b.

Method 1

Pepsi → [ ] → Coke

⇓

Either Coke/Pepsi

Possibilities

| Pepsi → Coke → Coke | Pepsi → Pepsi → Coke |

Pr[Pepsi → Coke → Coke] + Pr[Pepsi → Pepsi → Coke]

= Pr[Pepsi → Coke and Coke → Coke] +

Pr[Pepsi → Pepsi and Pepsi → Coke]

$= 0.2 * 0.9 + 0.8 * 0.2$

$= 0.34$

b.

Method 2 (Using Current Status Distribution Matrix)

$$P = \begin{array}{c} \\ \text{Coke} \\ \text{Pepsi} \end{array} \begin{array}{c} \begin{array}{cc} \text{Coke} & \text{Pepsi} \end{array} \\ \begin{bmatrix} 0.9 & 0.1 \\ 0.2 & 0.8 \end{bmatrix} \end{array}$$

Current Status Distribution Matrix (Q_n)

$$Q_0 = \begin{array}{c} \begin{array}{cc} \text{Coke} & \text{Pepsi} \end{array} \\ \begin{bmatrix} 0 & 1 \end{bmatrix} \end{array}$$

Since question is asking for two purchases from now,

$$\boxed{Q_2 = Q_0 * P^2}$$

$$= \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} 0.9 & 0.1 \\ 0.2 & 0.8 \end{bmatrix}^2$$

$$= \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} 0.83 & 0.17 \\ 0.34 & 0.66 \end{bmatrix}$$

$$= \begin{bmatrix} 0.34 & 0.66 \end{bmatrix}$$

Coke   Pepsi

∴ Probability is 0.34

c.

Currently Coke Purchaser

So, Current status distribution matrix (Q_0) =

$$\begin{bmatrix} Coke & Pepsi \\ 1 & 0 \end{bmatrix}$$

Probability he will buy (pepsi) three purchase from now $Q_3 = Q_0 * P^3$

$$= [1 \quad 0] \begin{bmatrix} 0.9 & 0.1 \\ 0.2 & 0.8 \end{bmatrix}^3$$

$$= [1 \quad 0] \begin{bmatrix} 0.781 & 0.219 \\ 0.438 & 0.562 \end{bmatrix}$$

$$= \begin{bmatrix} \underbrace{0.781}_{Coke} & \underbrace{0.219}_{Pepsi} \end{bmatrix}$$

∴ Probability he will buy Pepsi three purchases from now is 0.219.

d.

Current Status Distribution

Matrix (Q_0) = $\begin{bmatrix} \text{Coke} & \text{Pepsi} \\ 0.6 & 0.4 \end{bmatrix}$

$Q_3 = Q_0 * P^3$

$= \begin{bmatrix} 0.6 & 0.4 \end{bmatrix} \begin{bmatrix} 0.9 & 0.1 \\ 0.2 & 0.8 \end{bmatrix}^3$

$= \begin{bmatrix} \underbrace{0.6438}_{\text{Coke}} & \underbrace{0.3562}_{\text{Pepsi}} \end{bmatrix}$

$\therefore$ Probability = 0.6438

Numerical

Given that a chance of Ford car user to buy a Ford car in next purchase is 70% and that his next purchase is will be a Scorpio is 30% and chance of Scorpio car user to buy Scorpio car at the next purchase is 80% and chance that his next purchase will be Ford car is 20%. What is the probability to buy a Scorpio car after three purchase of a current Ford user? If 70% user use Ford car today, what percentage of user will use Scorpio after 3 purchase?

Numerical

Given that chance of a Honda bike user to buy Honda bike at next purchase is 70% and that his next purchase will be Yamaha is 30%. The chance of Yamaha bike user to buy Yamaha bike at next purchase is 80% and that his next purchase will be Honda is 20%. What is the probability to buy Yamaha bike after three purchase of a current Honda Bike user?

- Ans: 0.525

Chapter 6
Random Number

Random Numbers

- A number chosen from some specified distribution randomly.
- Random numbers are samples drawn from a uniformly distributed random variable between some satisfied intervals, they have equal probability of occurrence.
- A number chosen from some specified distribution randomly such that selection of large set of these numbers reproduces the underlying distribution is called random number.
- Every number is equally likely to occur and there is no pattern, and thus no way of predicting what number will be next in sequence.
- Most simulations are random number driven.

General Properties of Random Number

1. Uniformity
    - The random numbers generated should be uniform. That means a sequence of random numbers should be equally probable every where.
    - If we divide all the set of random numbers into several numbers of class interval then number of samples in each class should be same.
    - If 'N' number of random numbers are divided into 'K' class interval, then expected number of samples in each class should be equal to ei = N / K.
2. Independent
    - Each random number R_t is an independent sample drawn from a continuous uniform distribution between 0 and 1.
    - The probability density function(pdf) is given by:

$$\text{pdf} : f(x) = \begin{cases} 1 \; for \; 0 \le x \le 1 \\ 0 \; otherwise \end{cases}$$

3. Maximum Density: The large samples of random number should be generated in a given range.
4. Maximum Cycle: It states that the repetition of numbers should be allowed only after a large interval of time.

Pseudo Random Numbers

- Here pseudo means false.
- Pseudo implies that the random numbers are generated by <u>using some known arithmetic operation.</u>
- Since, the arithmetic operation is known and the <u>sequence of random numbers can be repeatedly obtained</u>, the numbers <u>cannot be called truly random.</u>
- However, the pseudo random numbers generated by many computer routines, very closely fulfill the requirement of desired randomness.

Pseudo Random Numbers

- If the method for generating random number or the random number generator is defective then generated pseudo random numbers may have following departures from idle randomness:

1. The generated numbers may not be uniformly distributed
2. The generated numbers may not be continuous
3. The mean of the generated numbers may be too high or too low
4. The variance may be too high or too low.

5. There may be cyclic patterns in the generated 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 group continuously below of mean.

Thus, before employing a pseudo random number generator, it should be properly validated, by testing the generated random numbers for randomness.

Generation of random number

- In computer simulation, where a very large number of random numbers is generally required, the random numbers can be obtained by the following methods:

1. Random numbers may be drawn from the random number tables stored in the memory of the computer.
2. Using electronics devices-Very expensive
3. Using arithmetic operation

Requirements of a good pseudo random generator

1. The sequence of generated random numbers must follow uniform distribution.
2. The sequence of random numbers generated must be statistically independent.
3. The sequence must be non-repeating for any desired length. Although theoretically not possible, a long repeatability cycle is adequate for practical purposes.
4. Generation of random numbers must be fast because in simulation studies, a large number of random numbers are required. A slow generator will greatly increase the time and cost fro simulation studies.
5. The generator must require less computer memory as will as computational resources.

Algorithm for generating Random Numbers

1. Linear Congruential Method

- A sequence of integers X1, X2, X3, ……….. are produced between zero and m-1 by using the recursive relation as follows:

$$X(i+1) = (a\,X(i) + c) \bmod m, \text{ for } i = 0, 1, 2, 3, 4, \ldots\ldots\ldots..$$

- The initial random integer X(0) is known as seed, a is called multiplier, c is increment and m is the modulus.

a. If a = 1 in above expression, the expression reduces to additive congruential method

i.e. $X(i+1) = (X(i) + c) \bmod m$

b. If c = 0 in above equation, the expression reduces to multiplicative congruential method,

i.e. $X(i+1) = aX(i) \bmod m$

c. If a>1 and c>0 in above expression, then it represents mixed type congruential method. For this type we use

$$X(i+1) = (a\,X(i) + c) \bmod m, \text{ for } i = 0, 1, 2, 3, 4, \ldots\ldots\ldots\ldots$$

2. Combined Linear Congruential Method: Combined linear congruential method uses the combination of two or more multiplicative congruential generators so as to provide good statistical properties and a longer period.

Note

- If question asks you to generate random numbers using Linear Congruential Method and provides you with multiplier, increment, modulus and seed values then always use the original formula.

$$\text{i.e. } X(i+1) = (a\,X(i) + c) \bmod m, \text{ for } i = 0, 1, 2, 3, 4, \ldots\ldots\ldots\ldots$$

Numerical

Let multiplier = 13, increment = 1 and modulus value = 19. Use congruential method to generate random numbers taking seed value = 1.

sol^n: Given a = 13, c = 1, m = 19 and X(0) = 1

We have, X(i+1) = (aX(i) + c) mod m

For i = 0, X(1) = (aX(0) + c) mod m

= (13 * 1 + 1) mod 19

= 14 mod 19 = 14

For i = 1, X(2) = (aX(1) + c) mod 19

= (13*14+1) mod 19

= 12

And so on.

Condition to stop Iteration

1. If question provides condition, do accordingly.
2. If condition not provided:
   a. Stop if same number repeats
   b. Else go and find all random numbers

Numerical

Use Linear Congruential Method to generate a sequence of three two digit random integers.

Given seed value = 32, multiplier = 8, increment = 47, modulus value = 100.

Numerical

Use Linear Congruential Method to generate a sequence of three two digit random integers.

Given seed value = 32, multiplier = 8, increment = 47, modulus value = 100.

Sol^n Given X(0) = 32, a = 8, c = 47, m = 100

X(1) = (8*32+47) mod 100 = 3 (Not OK)

X(2) = (8*3+47) mod 100 = 71 (OK)

X(3) = (8*71+47) mod 100 = 15 (OK)

X(4) = (8*15+47) mod 100 = 67 (OK)

Numerical

Use Multiplicative Congruential Method to generate a sequence of four three digit random integers.

Given seed value = 117, multiplier = 8, increment = 47, modulus value = 1000.

Test For Random Numbers

1. **Frequency test**: Uses the Kolmogorov-Smirnov(KS) or the chi-square test to compare the distribution of the set of numbers generated to a uniform distribution.
2. **Runs test**: Tests the runs up and down or the runs above and below the mean by comparing the actual values to expected values. The statistic for comparison is the chi-square.
3. **Autocorrelation test**: Tests the correlation between numbers and compares the sample correlation to the expected correlation of zero.
4. **Gap test**. Counts the number of digits that appear between repetitions of a particular digit and then uses the Kolmogorov-Smirnov(KS) test to compare with the expected number of gaps.
5. **Poker test**. Treats numbers grouped together as a poker hand. Then the hands obtained are compared to what is expected using the chi-square test.

Kolmogorov-Smirnov(KS) Test

- It is a test for random number developed by A.N. Kolmogorov and N.V. Smirnov.
- It is used to test the uniformity of random numbers i.e. whether random numbers are uniformly generated or not.
- This test is designed for continuous distributions where the Observed Cumulative Distribution Function(CDF) is compared with empirical CDF.

KS Test Algorithm

1. Define the hypothesis.
2. For provided data, rank data from smallest to greatest. Let $R(i)$ denote the i^{th} smallest observation among the N observations.

   i.e. $R(1) \leq R(2) \leq R(3) \leq \ldots \leq R(N)$

3. Compute D^+ and D^- as

$$D^+ = \max \left\{ \frac{i}{N} - R(i) \right\}$$

$$D^- = \max \left\{ R(i) - \frac{(i-1)}{N} \right\}$$

for $1 \leq i \leq N$

KS Test Algorithm

4. Compute $D = \max\{D^+, D^-\}$

5. Determine the critical value D_α from the table for specific value of α and sample size N.

6. If $D > D_\alpha$, the null hypothesis is rejected. This means the random numbers are **not uniform.**

   If $D \leq D_\alpha$, the null hypothesis is not rejected. This means the random numbers **are uniform**

n	α 0.01	α 0.05	α 0.1	α 0.15	α 0.2
1	0.995	0.975	0.950	0.925	0.900
2	0.929	0.842	0.776	0.726	0.684
3	0.828	0.708	0.642	0.597	0.565
4	0.733	0.624	0.564	0.525	0.494
5	0.669	0.565	0.510	0.474	0.446
6	0.618	0.521	0.470	0.436	0.410
7	0.577	0.486	0.438	0.405	0.381
8	0.543	0.457	0.411	0.381	0.358
9	0.514	0.432	0.388	0.360	0.339
10	0.490	0.410	0.368	0.342	0.322
11	0.468	0.391	0.352	0.326	0.307
12	0.450	0.375	0.338	0.313	0.295
13	0.433	0.361	0.325	0.302	0.284
14	0.418	0.349	0.314	0.292	0.274
15	0.404	0.338	0.304	0.283	0.266
16	0.392	0.328	0.295	0.274	0.258
17	0.381	0.318	0.286	0.266	0.250
18	0.371	0.309	0.278	0.259	0.244
19	0.363	0.301	0.272	0.252	0.237
20	0.356	0.294	0.264	0.246	0.231
25	0.320	0.270	0.240	0.220	0.210
30	0.290	0.240	0.220	0.200	0.190
35	0.270	0.230	0.210	0.190	0.180
40	0.250	0.210	0.190	0.180	0.170
45	0.240	0.200	0.180	0.170	0.160
50	0.230	0.190	0.170	0.160	0.150
OVER 50	$\frac{1.63}{\sqrt{n}}$	$\frac{1.36}{\sqrt{n}}$	$\frac{1.22}{\sqrt{n}}$	$\frac{1.14}{\sqrt{n}}$	$\frac{1.07}{\sqrt{n}}$

Numerical –KS Test

Perform uniformity test using KS test with a level of significance $\alpha = 0.05$ on the following five generated numbers.

0.44, 0.81, 0.14, 0.05, 0.93

Soln:

Let H_0 represent the null hypothesis.

H_0: the generated numbers are uniform

Arranging given data in ascending order

i	1	2	3	4	5
R(i)	0.05	0.14	0.44	0.81	0.93

Calculating D^+ and D^-

(i)	$R(i)$	$\frac{i}{N}$	$\frac{(i-1)}{N}$	$\frac{i}{N} - R(i)$	$R(i) - \frac{(i-1)}{N}$
1	0.05	0.2	0	0.15	0.05
2	0.14	0.4	0.2	0.26	-0.06
3	0.44	0.6	0.4	0.16	0.04
4	0.81	0.8	0.6	-0.01	0.21
5	0.93	1	0.8	0.07	0.13

Note: While determining D^+ and D^-, no need to ~~calculate~~ consider negative values.

So, we have $D^+ = \max\left\{\frac{i}{N} - R_{(i)}\right\}$

$= 0.26$

and $D^- = \max\left\{R_{(i)} - \frac{(i-1)}{N}\right\}$

$= 0.21$

Calculation of D

$D = \max\{D^+, D^-\}$

$= 0.26$

The critical value D_α for $\alpha = 0.05$ and $N = 5$ from table is 0.565

Since $D < D_\alpha$, the null hypothesis is not rejected. So we can conclude that the random numbers are uniform.

Numerical

K-S test is to be performed to test the uniformity of following random numbers with a level of significance of $\alpha = 0.05$.

0.24, 0.89, 0.11, 0.61, 0.23, 0.86, 0.41, 0.64, 0.50, 0.65

i	R(i)	i/n	(i-1)/n	(i/n)-R(i)	R(i)-((i-1)/n)
1	0.11	0.1	0	-0.01	0.11
2	0.23	0.2	0.1	-0.03	0.13
3	0.24	0.3	0.2	0.06	0.04
4	0.41	0.4	0.3	-0.01	0.11
5	0.50	0.5	0.4	0	0.10
6	0.61	0.6	0.5	-0.01	0.11
7	0.64	0.7	0.6	0.06	0.04
8	0.65	0.8	0.7	0.15	-0.05
9	0.86	0.9	0.8	0.04	0.06
10	0.89	1	0.9	0.11	-0.01

Chi-Square Test

- It is a type of frequency test
- It is a test used to check the randomness of a distribution
- This statistical test is used to determine how often certain observed data fit the theoretically expected data.
- This method compares the observed frequency with the theoretical. So it determines how often certain observed data fit the theoretically expected data.

Chi-Square Test

- The Chi-Square test uses the following sample statistics
- The total N observations are divided into n number of classes.

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

where n = number of classes

O_i = Observed number of occurrence(frequency) in each class

E_i = Expected number of occurrence(frequency) in each class

N = total number of observations

Chi-Square Test

Note:

1. For chi square test, degree of freedom = n-1
   - where n represents number of classes
2. Chi Square test is usually recommended when $E_i \geq 5$

   i.e number of expected occurrence/frequency in each class ≥ 5

Chi-Square Test - Procedure

1. Divide the sample range into n intervals. Note: 3 values in each interval at least.
2. Determine:
   a. Expected number of values E_i in each interval i under the null hypothesis.
   b. Observed number of values O_i in each interval.

3. Compute the statistic χ^2 value.
4. Find the standard value of χ^2 based on given confidence level/ level of significance(α) and degree of freedom. For simple chi-square test

   degree of freedom = n - 1

5. Draw conclusion accordingly. If $\chi^2_{calculated} < \chi^2_{standard}$ the null hypothesis is not rejected. Else the null hypothesis is rejected.

Chi-Square Test (Example)

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, 22

sol^n : Let H_0 represents null hypothesis where H_0: the numbers are acceptable for given confidence level

Here, Total number of samples (N) = 100

Let us divide these data into 10 classes i.e. n =10

E_i=N/n=100/10=10

`

Classes	Observed Frequency(Oi)	$(O_i - E_i)$	$(O_i - E_i)^2$	
0 < r ≤ 10	13	3	9	0.9
10 < r ≤ 20	7	-3	9	0.9
20 < r ≤ 30	12	2	4	0.4
30 < r ≤ 40	13	3	9	0.9
40 < r ≤ 50	7	-3	9	0.9
50 < r ≤ 60	13	3	9	0.9
60 < r ≤ 70	5	-5	25	2.5
70 < r ≤ 80	10	0	0	0
80 < r ≤ 90	12	2	4	0.4
90 < r ≤ 100	8	-2	4	0.4

So $\chi^2 = 8.2$

i.e. $\chi^2_{calculated} = 8.2$

Given, confidence level = 95% = 0.95

So level of significance(α) = 1 – 0.95 = 0.05

Similarly degree of freedom in this case = n – 1

= 10 - 1 = 9

Now we use Chi-Square value from table for α = 0.05 and degree of freedom =9

Percentage Points of the Chi-Square Distribution

Degrees of Freedom	Probability of a larger value of x^2								
	0.99	0.95	0.90	0.75	0.50	0.25	0.10	0.05	0.01
1	0.000	0.004	0.016	0.102	0.455	1.32	2.71	3.84	6.63
2	0.020	0.103	0.211	0.575	1.386	2.77	4.61	5.99	9.21
3	0.115	0.352	0.584	1.212	2.366	4.11	6.25	7.81	11.34
4	0.297	0.711	1.064	1.923	3.357	5.39	7.78	9.49	13.28
5	0.554	1.145	1.610	2.675	4.351	6.63	9.24	11.07	15.09
6	0.872	1.635	2.204	3.455	5.348	7.84	10.64	12.59	16.81
7	1.239	2.167	2.833	4.255	6.346	9.04	12.02	14.07	18.48
8	1.647	2.733	3.490	5.071	7.344	10.22	13.36	15.51	20.09
9	2.088	3.325	4.168	5.899	8.343	11.39	14.68	16.92	21.67
10	2.558	3.940	4.865	6.737	9.342	12.55	15.99	18.31	23.21
11	3.053	4.575	5.578	7.584	10.341	13.70	17.28	19.68	24.72
12	3.571	5.226	6.304	8.438	11.340	14.85	18.55	21.03	26.22
13	4.107	5.892	7.042	9.299	12.340	15.98	19.81	22.36	27.69
14	4.660	6.571	7.790	10.165	13.339	17.12	21.06	23.68	29.14
15	5.229	7.261	8.547	11.037	14.339	18.25	22.31	25.00	30.58
16	5.812	7.962	9.312	11.912	15.338	19.37	23.54	26.30	32.00
17	6.408	8.672	10.085	12.792	16.338	20.49	24.77	27.59	33.41
18	7.015	9.390	10.865	13.675	17.338	21.60	25.99	28.87	34.80
19	7.633	10.117	11.651	14.562	18.338	22.72	27.20	30.14	36.19
20	8.260	10.851	12.443	15.452	19.337	23.83	28.41	31.41	37.57
22	9.542	12.338	14.041	17.240	21.337	26.04	30.81	33.92	40.29
24	10.856	13.848	15.659	19.037	23.337	28.24	33.20	36.42	42.98
26	12.198	15.379	17.292	20.843	25.336	30.43	35.56	38.89	45.64
28	13.565	16.928	18.939	22.657	27.336	32.62	37.92	41.34	48.28
30	14.953	18.493	20.599	24.478	29.336	34.80	40.26	43.77	50.89
40	22.164	26.509	29.051	33.660	39.335	45.62	51.80	55.76	63.69
50	27.707	34.764	37.689	42.942	49.335	56.33	63.17	67.50	76.15
60	37.485	43.188	46.459	52.294	59.335	66.98	74.40	79.08	88.38

From Chi-Square Table, $\chi^2_{standard} = 16.92$

Since $\chi^2_{calculated} < \chi^2_{standard}$ the null hypothesis is not rejected. So we conclude that the numbers are acceptable for 95% confidence level.

K-S test vs Chi-Square Test

K-S test	Chi-Square Test
Done for smaller samples.	Done for larger samples.
Difference between observed and expected CDFs(Cumulative Distribution Function)	Difference between observed and expected PDFs(Probability Density Function)
Uses each observed sample without grouping	Group observations

Numerical – 2074 Bhadra

7. What are the properties of Random number? Using Chi-Square test the uniformity at 90% for the given random numbers. Degree of freedom for 6 = 10.645, 7 = 12.017, 8 = 13.362 9 = 14.684, 10 = 15.987.

20	34	43	42	14	10	33	17	6	11
15	16	4	1	35	22	9	46	37	57
51	49	40	27	59	5	44	19	41	55
53	29	3	31	48	8	56	28	12	7

Classes	Observed Frequency(Oi)	$(O_i - E_i)$	$(O_i - E_i)^2$	
0 < r ≤ 6	5	1	1	0.25
6 < r ≤12	6	2	4	1
12 < r ≤ 18	4	0	0	0
18 < r ≤ 24	3	-1	1	0.25
24 < r ≤ 30	3	-1	1	0.25
30 < r ≤ 36	4	0	0	0
36 < r ≤ 42	4	0	0	0
42 < r ≤ 48	4	0	0	0
48 < r ≤ 54	4	0	0	0
54 < r ≤ 60	3	-1	1	0.25
				∑=2

Note: E_i=N/n

- Here degree of freedom = n-1 = 10-1=9

Here $\chi^2_{standard} = 14.684$

Since $\chi^2_{calculated} < \chi^2_{standard}$ the null hypothesis is not rejected.

Gap Test

- The gap test is used to determine the significance of the interval between recurrence of the same digit.
- The Gap Test measures the number of digits between successive occurrences of the same digit.
- A gap of length x occurs between the recurrence of some digit.
- If we are only concerned with digits between 0 and 9 then,

$$P(\text{gap of } n) = 0.9^n * 0.1$$

- The theoretical frequency distribution of randomly ordered digits is given by

$$P(\text{gap} \leq x) = F(x) = 0.1\sum_{n=0}^{x} 0.9^n$$

$$= 1 - 0.9^{x+1}$$

Algorithm for Gap Test

Step 1

Specify the Cumulative Distribution Function(CDF) for the theoretical frequency distribution given by

$$P(gap \le x) = F(x) = 0.1\sum_{n=0}^{x} 0.9^{n}$$
$$= 1 - 0.9^{x+1}$$

where x = maximum gap length based on the selected class interval width

Step 2

Arrange the observed sample of gaps in a cumulative distribution with these same classes.

Algorithm for Gap Test

- Step 3

Find D, the maximum deviation between F(x) and $S_N(x)$ as

$$D = \max |F(x) - S_N(x)|$$

Where $S_N(X)$ is the observed frequency distribution.

$$S_N(x) = \frac{Number\ of\ gaps \le x}{Total\ number\ of\ gaps}$$

<u>This value is equal to the cumulative relative frequency of each gap class</u>

Step 4

Determine the critical value D_α, from Table(K-S critical value) for the specified value of α and the sample size N.

Step 5

If the calculated value of D is greater than the tabulated value of D_α , the null hypothesis of independence is rejected.

Numerical- Gap Test

Based on the frequency with which gaps occur, analyze following 110 digits to test whether they are independent. Use $\alpha = 0.05$

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

sol^n : Let H_0 represents null hypothesis.

H_0 : The numbers are independent

Here digits are from 0 to 9. So total number of distinct digits = 10

So number of gaps(N) = Number of data values – Number of distinct digits

= 110 -10 = 100

Numerical- Gap Test

Gap Length	Frequency	Relative Frequency	Cumulative Relative frequency	$F(x) = 1 - 0.9^{x+1}$	$\lvert F(x) - S_N(x) \rvert$
0 – 3	35	0.35	0.35	0.3439	0.0061
4 – 7	22	0.22	0.57	0.5695	0.0005
8 – 11	17	0.17	0.74	0.7176	0.0224
12 – 15	9	0.09	0.83	0.8147	0.0153
16 – 19	5	0.05	0.88	0.8784	0.0016
20 – 23	6	0.06	0.94	0.9202	0.0198
24 – 27	3	0.03	0.97	0.9497	0.0223
28 – 31	0	0	0.97	0.9657	0.0043
32 – 35	0	0	0.97	0.9775	0.0075
36 – 39	2	0.02	0.99	0.9852	0.0043
40 – 43	0	0	0.99	0.9903	0.0003
44 – 47	1	0.01	1	0.9936	0.0064

n	α 0.01	α 0.05	α 0.1	α 0.15	α 0.2
1	0.995	0.975	0.950	0.925	0.900
2	0.929	0.842	0.776	0.726	0.684
3	0.828	0.708	0.642	0.597	0.565
4	0.733	0.624	0.564	0.525	0.494
5	0.669	0.565	0.510	0.474	0.446
6	0.618	0.521	0.470	0.436	0.410
7	0.577	0.486	0.438	0.405	0.381
8	0.543	0.457	0.411	0.381	0.358
9	0.514	0.432	0.388	0.360	0.339
10	0.490	0.410	0.368	0.342	0.322
11	0.468	0.391	0.352	0.326	0.307
12	0.450	0.375	0.338	0.313	0.295
13	0.433	0.361	0.325	0.302	0.284
14	0.418	0.349	0.314	0.292	0.274
15	0.404	0.338	0.304	0.283	0.266
16	0.392	0.328	0.295	0.274	0.258
17	0.381	0.318	0.286	0.266	0.250
18	0.371	0.309	0.278	0.259	0.244
19	0.363	0.301	0.272	0.252	0.237
20	0.356	0.294	0.264	0.246	0.231
25	0.320	0.270	0.240	0.220	0.210
30	0.290	0.240	0.220	0.200	0.190
35	0.270	0.230	0.210	0.190	0.180
40	0.250	0.210	0.190	0.180	0.170
45	0.240	0.200	0.180	0.170	0.160
50	0.230	0.190	0.170	0.160	0.150
OVER 50	$\frac{1.63}{\sqrt{n}}$	$\frac{1.36}{\sqrt{n}}$	$\frac{1.22}{\sqrt{n}}$	$\frac{1.14}{\sqrt{n}}$	$\frac{1.07}{\sqrt{n}}$

Numerical- Gap Test

So the calculated value of D is

$$D = \max| F(x) - S_N(x) |$$
$$= 0.0224$$

The critical value of D for $\alpha = 0.05$ is $D_\alpha = \frac{1.36}{\sqrt{N}} = \frac{1.36}{\sqrt{100}}$

$$= 0.136$$

Since $D < D_\alpha$, the null the null hypothesis is not rejected. So the numbers are independent.

Gap Test Example For Exam

Explain the algorithm for gap test with an example.

Let us assume 110 random numbers between 0 to 9 with varying gap length. Let the maximum gap length be 34.

Let H_0 represents null hypothesis.

H_0 : The numbers are independent

Here digits are from 0 to 9. So total number of distinct digits = 10

So number of gaps(N) = Number of data values – Number of distinct digits

= 110 -10 = 100

Numerical- Gap Test

Gap Length	Frequency	Relative Frequency	Cumulative Relative frequency	$F(x) = 1 - 0.9^{x+1}$	$\lvert F(x) - S_N(x) \rvert$
0 – 5	45	0.45	0.45	0.4685	0.0185
6 – 11	15	0.15	0.6	0.7175	0.1175
12 – 17	12	0.12	0.72	0.8499	0.1299
18 – 23	8	0.08	0.8	0.920	0.12
24 – 29	13	0.13	0.93	0.9576	0.0276
30 – 35	7	0.07	1	0.9774	0.0226

Numerical- Gap Test

So the calculated value of D is

$$D = \max| F(x) - S_N(x) |$$
$$= 0.1299$$

The critical value of D for $\alpha = 0.05$ is $D_\alpha = \frac{1.36}{\sqrt{N}} = \frac{1.36}{\sqrt{100}}$

$$= 0.136$$

Since $D < D_\alpha$, the null the null hypothesis is not rejected. So the numbers are independent.

Poker Test

- This test gets its name from a game of cards called poker.
- Poker test for independence is based on the frequency with which certain digits are repeated.
- Poker test not only tests for randomness of the sequence of numbers, but also the digits comprising of each number.
- Poker test treats numbers grouped together as a poker hand. Then the hands obtained are compared to what is expected using the chi-square test.

Poker test for 3 digit random number

 Possibilities for 3 digit number:

a. Three different digits
b. Three like digits
c. Exactly one pair

Calculating Probabilities of each possibility

1. Probability(Three different Digits) $= \frac{10}{10} * \frac{9}{10} * \frac{8}{10}$

   $= 0.72$

2. Probability(Three like digits) $= \frac{10}{10} * \frac{1}{10} * \frac{1}{10}$

   $= 0.01$

3. Probability(Exactly One Pair) $= 3C2 * \frac{10}{10} * \frac{1}{10} * \frac{9}{10}$

   $= \frac{3!}{(3-2)! * 2!} * \frac{10}{10} * \frac{1}{10} * \frac{9}{10}$

   $= 0.27$

OR last probability value can be calculated by subtracting other probabilities from 1

$= 1 - 0.72 - 0.01$

$= 0.27$

Numerical

A sequence of 1000 three-digit numbers has been generated and an analysis indicates that 680 have three different digits, 289 contain exactly one pair of like digits, and 31 contain three like digits. Based on the poker test, are these numbers independent? Take $\alpha = 0.05$.

soln: Let H_0 be null hypothesis.

H_0 : The numbers are independent

Total number of three-digit numbers(N) = 1000

Numerical

- Here,

1. Probability(Three different Digits) $= \frac{10}{10} * \frac{9}{10} * \frac{8}{10}$

   $= 0.72$

2. Probability(Three like digits) $= \frac{10}{10} * \frac{1}{10} * \frac{1}{10}$

   $= 0.01$

3. Probability(Exactly One Pair) $= 3C2 * \frac{10}{10} * \frac{1}{10} * \frac{9}{10}$

   $= \frac{3!}{(3-2)! * 2!} * \frac{10}{10} * \frac{1}{10} * \frac{9}{10}$

   $= 0.27$

Combination (i)	Expected Frequency (E_i) = Probability(i)*N	Observed Frequency (O_i)	
Three Different Digits	0.72*1000 = 720	680	2.22
Three like digits	0.01*1000 = 10	31	44.10
Exactly one pair	0.27*1000 = 270	289	1.33
	1000	1000	

-

So $\chi^2_{calculated} = 47.65$

Here degree of freedom = n-1 = 3-1 = 2

So $\chi^2_{standard} = 5.99$

Since $\chi^2_{calculated} > \chi^2_{standard}$ the null hypothesis is rejected. So given numbers are not independent.

Numerical

A sequence of 1000 three-digit numbers has been generated and an analysis indicates that 695 have three different digits, 293 contain exactly one pair of like digits, and 12 contain three like digits. Based on the poker test, are these numbers independent? Take $\alpha = 0.05$.

Poker test for 4 digit random number

 Possibilities for 4 digit number:

a. Four different digits
b. Exactly one pair
c. Two pairs
d. Three of a kind
e. All four like digits

Calculating Probabilities of each possibility

1. Probability(Four different Digits) = $\frac{10}{10} * \frac{9}{10} * \frac{8}{10} * \frac{7}{10}$
$= 0.504$

2. Probability(Exactly one pair) =4C2* $\frac{10}{10} * \frac{1}{10} * \frac{9}{10} * \frac{8}{10}$
$= 0.432$

3. Probability(Two Pairs) = 4C2 * $\frac{10}{10} * \frac{1}{10} * \frac{2C2}{2!} * \frac{9}{10} * \frac{1}{10}$
$= 0.027$

4. Probability(Three of a kind) = 4C3 * $\frac{10}{10} * \frac{1}{10} * \frac{1}{10} * \frac{9}{10}$
$= 0.036$

Calculating Probabilities of each possibility

5. Probability(All four like digits) $= \frac{10}{10} * \frac{1}{10} * \frac{1}{10} * \frac{1}{10}$
$= 0.001$

Numerical

A sequence of 1000 four-digit numbers has been generated and an analysis indicates:

Combinations	Observed Frequency
Four Different Digits	540
One pair	320
Two pairs	70
Three like digits	50
Four like digits	20
	1000

Based on poker test, test these numbers are independent for $\alpha = 0.05$

soln: Let H_0 be null hypothesis.

H_0 : The numbers are independent

Total number of four-digit numbers(N) = 1000

Here,

Probability(Four different Digits) = $\frac{10}{10} * \frac{9}{10} * \frac{8}{10} * \frac{7}{10}$

$= 0.504$

Probability(Exactly one pair) =4C2* $\frac{10}{10} * \frac{1}{10} * \frac{9}{10} * \frac{8}{10}$

$= 0.432$

Probability(Two Pairs) = 4C2 * $\frac{10}{10} * \frac{1}{10} * \frac{2C2}{2!} * \frac{9}{10} * \frac{1}{10}$

$= 0.027$

Probability(Three like digits) = 4C3 * $\frac{10}{10} * \frac{1}{10} * \frac{1}{10} * \frac{9}{10}$

$= 0.036$

Probability(Four like digits) = $\frac{10}{10} * \frac{1}{10} * \frac{1}{10} * \frac{1}{10}$

$= 0.001$

Combination (i)	Expected Frequency (E_i) = Probability(i)*N	Observed Frequency (O_i)	
Four different Digits	0.504*1000 = 504	540	2.5714
Exactly one pair	0.432*1000 = 432	320	29.037
Two Pairs	0.027*1000 = 27	70	68.4814
Three like digits	0.036*1000= 36	50	5.444
Four like digits	0.001*1000 = 1	20	361
	1000	1000	

So $\chi^2_{calculated} = 466.5298$

Here degree of freedom = n-1 = 5-1 = 4

So $\chi^2_{standard} = 9.49$

Since $\chi^2_{calculated} > \chi^2_{standard}$ the null hypothesis is rejected. So given numbers are not independent.

Numerical

A set of 10,000 4-digit random values have been generated. An observation shows than 5065 values have all different digits, 2000 have 2 of a kind digits, 760 have 3 of a kind, 1500 have 2 pairs and 675 have all same digits. Test these values for randomness using Poker test (Use α :0.05).

Numerical

Write an algorithm for gap test. Formulate 4-digit poker test with suitable data with example.

2071 Bhadra

Poker test for 5 digit random number

 Possibilities for 5 digit number:

a. All different digits
b. Exactly one pair
c. Two pairs
d. Three of a kind
e. Full House/Three of a kind + Two of a kind
f. Four of a kind
g. Five of a kind

Calculating Probabilities of each possibility

1. Probability(All different Digits) $= \frac{10}{10} * \frac{9}{10} * \frac{8}{10} * \frac{7}{10} * \frac{6}{10}$

$= 0.3024$

2. Probability(Exactly one pair) $= 5C2 * \frac{10}{10} * \frac{1}{10} * \frac{9}{10} * \frac{8}{10} * \frac{7}{10}$

$= 0.504$

3. Probability(Two Pairs) $= 5C2 * \frac{10}{10} * \frac{1}{10} * \frac{3C2}{2!} * \frac{9}{10} * \frac{1}{10} * \frac{8}{10}$

$= 0.108$

4. Probability(Three of a kind) $= 5C3 * \frac{10}{10} * \frac{1}{10} * \frac{1}{10} * \frac{9}{10} * \frac{8}{10}$

$= 0.072$

Calculating Probabilities of each possibility

5. Probability(Full House/Three of a kind + Two of a kind)

$$= 5C3*\frac{10}{10}*\frac{1}{10}*\frac{1}{10}*2C2*\frac{9}{10}*\frac{1}{10}$$

$$= 0.009$$

6. Probability(Four of a kind) $= 5C4*\frac{10}{10}*\frac{1}{10}*\frac{1}{10}*\frac{1}{10}*\frac{9}{10}$

$$= 0.0045$$

7. Probability(Five of a kind) $= \frac{10}{10}*\frac{1}{10}*\frac{1}{10}*\frac{1}{10}*\frac{1}{10}$

$$=0.0001$$

Numerical

Write an algorithm for Gap Test. A sequence of 10,000 five digit numbers has been generated and analysis indicates the following combinations and frequencies. Based on Poker Test check whether the number are independent. Use $\chi^2 0.056,6 = 12.592$ [4+6]

Combinations	Observed Frequencies
All Different	3044
One pair	5020
Two pair	1090
Three of a kind	700
Full house	95
Four of a kind	40
Five of a kind	11
Total	**10,000**

soln: Let H_0 be null hypothesis.

H_0 : The numbers are independent

Total number of five-digit numbers(N) = 10000

Here, Probability(All different Digits) = $\frac{10}{10} * \frac{9}{10} * \frac{8}{10} * \frac{7}{10} * \frac{6}{10}$

= 0.3024

Probability(one pair) =5C2* $\frac{10}{10} * \frac{1}{10} * \frac{9}{10} * \frac{8}{10} * \frac{7}{10}$

= 0.504

Probability(Two Pairs) = 5C2 * $\frac{10}{10} * \frac{1}{10} * \frac{3C2}{2!} * \frac{9}{10} * \frac{1}{10} * \frac{8}{10}$

= 0.108

Probability(Three of a kind) = 5C3 * $\frac{10}{10} * \frac{1}{10} * \frac{1}{10} * \frac{9}{10} * \frac{8}{10}$

= 0.072

- Probability(Full House/Three of a kind + Two of a kind)

$$= 5C3 * \frac{10}{10} * \frac{1}{10} * \frac{1}{10} * 2C2 * \frac{9}{10} * \frac{1}{10}$$

$$= 0.009$$

Probability(Four of a kind) $= 5C4 * \frac{10}{10} * \frac{1}{10} * \frac{1}{10} * \frac{1}{10} * \frac{9}{10}$

$$= 0.0045$$

Probability(Five of a kind) $= \frac{10}{10} * \frac{1}{10} * \frac{1}{10} * \frac{1}{10} * \frac{1}{10}$

$$= 0.0001$$

Combination (i)	Expected Frequency (E_i) = Probability(i)*N	Observed Frequency (O_i)	
All different Digits	3024	3044	0.1322
one pair	5040	5020	0.0793
Two Pairs	1080	1090	0.0925
Three of a kind	720	700	0.5556
Full House	90	95	0.2778
Four of a kind	45	40	0.5556
Five of a kind	1	11	100
	10000	10000	

So $\chi^2_{calculated} = 101.693$

Here degree of freedom = n-1 = 7-1 = 6

So $\chi^2_{standard} = 12.59$

Since $\chi^2_{calculated} > \chi^2_{standard}$ the null hypothesis is rejected. So given numbers are not independent.

Percentage Points of the Chi-Square Distribution

Degrees of Freedom	Probability of a larger value of x^2								
	0.99	0.95	0.90	0.75	0.50	0.25	0.10	0.05	0.01
1	0.000	0.004	0.016	0.102	0.455	1.32	2.71	3.84	6.63
2	0.020	0.103	0.211	0.575	1.386	2.77	4.61	5.99	9.21
3	0.115	0.352	0.584	1.212	2.366	4.11	6.25	7.81	11.34
4	0.297	0.711	1.064	1.923	3.357	5.39	7.78	9.49	13.28
5	0.554	1.145	1.610	2.675	4.351	6.63	9.24	11.07	15.09
6	0.872	1.635	2.204	3.455	5.348	7.84	10.64	12.59	16.81
7	1.239	2.167	2.833	4.255	6.346	9.04	12.02	14.07	18.48
8	1.647	2.733	3.490	5.071	7.344	10.22	13.36	15.51	20.09
9	2.088	3.325	4.168	5.899	8.343	11.39	14.68	16.92	21.67
10	2.558	3.940	4.865	6.737	9.342	12.55	15.99	18.31	23.21
11	3.053	4.575	5.578	7.584	10.341	13.70	17.28	19.68	24.72
12	3.571	5.226	6.304	8.438	11.340	14.85	18.55	21.03	26.22
13	4.107	5.892	7.042	9.299	12.340	15.98	19.81	22.36	27.69
14	4.660	6.571	7.790	10.165	13.339	17.12	21.06	23.68	29.14
15	5.229	7.261	8.547	11.037	14.339	18.25	22.31	25.00	30.58
16	5.812	7.962	9.312	11.912	15.338	19.37	23.54	26.30	32.00
17	6.408	8.672	10.085	12.792	16.338	20.49	24.77	27.59	33.41
18	7.015	9.390	10.865	13.675	17.338	21.60	25.99	28.87	34.80
19	7.633	10.117	11.651	14.562	18.338	22.72	27.20	30.14	36.19
20	8.260	10.851	12.443	15.452	19.337	23.83	28.41	31.41	37.57
22	9.542	12.338	14.041	17.240	21.337	26.04	30.81	33.92	40.29
24	10.856	13.848	15.659	19.037	23.337	28.24	33.20	36.42	42.98
26	12.198	15.379	17.292	20.843	25.336	30.43	35.56	38.89	45.64
28	13.565	16.928	18.939	22.657	27.336	32.62	37.92	41.34	48.28
30	14.953	18.493	20.599	24.478	29.336	34.80	40.26	43.77	50.89
40	22.164	26.509	29.051	33.660	39.335	45.62	51.80	55.76	63.69
50	27.707	34.764	37.689	42.942	49.335	56.33	63.17	67.50	76.15
60	37.485	43.188	46.459	52.294	59.335	66.98	74.40	79.08	88.38

Numerical

A sequence of 10,000 random numbers has been generated and an analysis shows following combinations and frequencies. For $\alpha = 0.05$ check whether generated numbers are independent or not.

Combination	Observed Frequency
All different Digits	3054
one pair	5020
Two Pairs	1073
Three of a kind	710
Full House	95
Four of a kind	44
Five of a kind	4

Runs Test

- The runs test can be used to decide if a dataset is from a random process.
- A run is defined as a series of increasing values or a series of decreasing values.
- A run is a succession of occurrences of certain type preceded and followed by occurrences of the alternate type or by no occurrences at all.
- If N is the total number of numbers in a sequence, the maximum number of runs is N-1 and minimum number of runs is 1.
- Test statistics is

$$Z_0 = \frac{x_r - \mu_r}{\sigma_r}$$

where x_r = observed number of runs

μ_r = expected number of runs

σ_r = standard deviate of number of runs

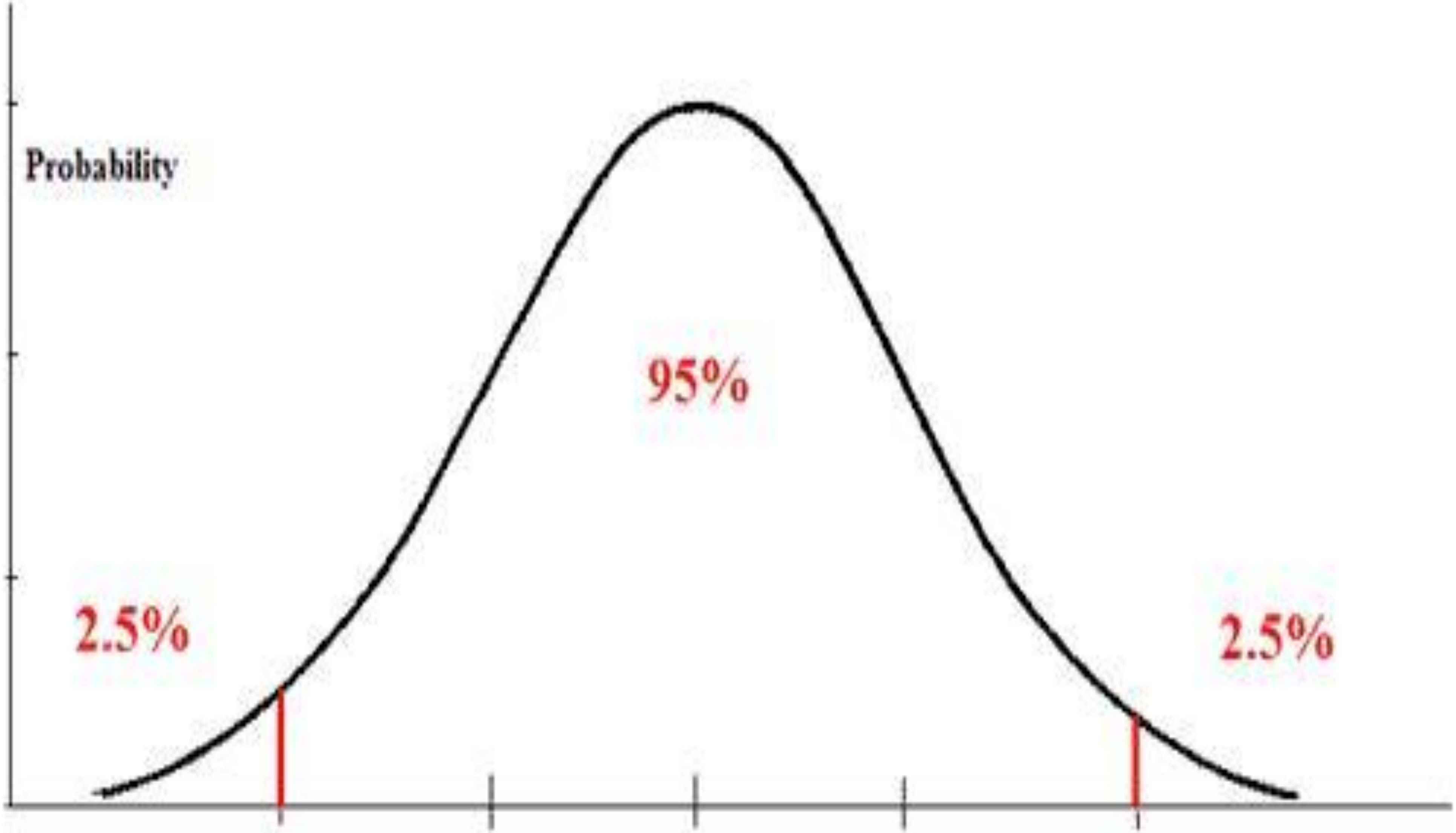
Runs Test

- Here $\mu_r = \frac{2n_1n_2}{n_1+n_2} + 1$

and $\sigma_r = \sqrt{\frac{2n_1n_2(2n_1n_2 - n_1 - n_2)}{(n_1+n_2)^2\,(n_1+n_2-1)}}$

where n1 and n2 represents number of occurrence of a type and number of occurrence of alternate type respectively

Acceptance region for acceptance of hypothesis is $-Z_{\alpha/2} \leq Z_0 \leq Z_{\alpha/2}$


Probability
95%
2.5%
2.5%

z	0	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09
-0	.50000	.49601	.49202	.48803	.48405	.48006	.47608	.47210	.46812	.46414
-0.1	.46017	.45620	.45224	.44828	.44433	.44034	.43640	.43251	.42858	.42465
-0.2	.42074	.41683	.41294	.40905	.40517	.40129	.39743	.39358	.38974	.38591
-0.3	.38209	.37828	.37448	.37070	.36693	.36317	.35942	.35569	.35197	.34827
-0.4	.34458	.34090	.33724	.33360	.32997	.32636	.32276	.31918	.31561	.31207
-0.5	.30854	.30503	.30153	.29806	.29460	.29116	.28774	.28434	.28096	.27760
-0.6	.27425	.27093	.26763	.26435	.26109	.25785	.25463	.25143	.24825	.24510
-0.7	.24196	.23885	.23576	.23270	.22965	.22663	.22363	.22065	.21770	.21476
-0.8	.21186	.20897	.20611	.20327	.20045	.19766	.19489	.19215	.18943	.18673
-0.9	.18406	.18141	.17879	.17619	.17361	.17106	.16853	.16602	.16354	.16109
-1	.15866	.15625	.15386	.15151	.14917	.14686	.14457	.14231	.14007	.13786
-1.1	.13567	.13350	.13136	.12924	.12714	.12507	.12302	.12100	.11900	.11702
-1.2	.11507	.11314	.11123	.10935	.10749	.10565	.10383	.10204	.10027	.09853
-1.3	.09680	.09510	.09342	.09176	.09012	.08851	.08692	.08534	.08379	.08226
-1.4	.08076	.07927	.07780	.07636	.07493	.07353	.07215	.07078	.06944	.06811
-1.5	.06681	.06552	.06426	.06301	.06178	.06057	.05938	.05821	.05705	.05592
-1.6	.05480	.05370	.05262	.05155	.05050	.04947	.04846	.04746	.04648	.04551
-1.7	.04457	.04363	.04272	.04182	.04093	.04006	.03920	.03836	.03754	.03673
-1.8	.03593	.03515	.03438	.03362	.03288	.03216	.03144	.03074	.03005	.02938
-1.9	.02872	.02807	.02743	.02680	.02619	.02559	.02500	.02442	.02385	.02330
-2	.02275	.02222	.02169	.02118	.02068	.02018	.01970	.01923	.01876	.01831
-2.1	.01786	.01743	.01700	.01659	.01618	.01578	.01539	.01500	.01463	.01426
-2.2	.01390	.01355	.01321	.01287	.01255	.01222	.01191	.01160	.01130	.01101
-2.3	.01072	.01044	.01017	.00990	.00964	.00939	.00914	.00889	.00866	.00842
-2.4	.00820	.00798	.00776	.00755	.00734	.00714	.00695	.00676	.00657	.00639
-2.5	.00621	.00604	.00587	.00570	.00554	.00539	.00523	.00508	.00494	.00480
-2.6	.00466	.00453	.00440	.00427	.00415	.00402	.00391	.00379	.00368	.00357
-2.7	.00347	.00336	.00326	.00317	.00307	.00298	.00289	.00280	.00272	.00264
-2.8	.00256	.00248	.00240	.00233	.00226	.00219	.00212	.00205	.00199	.00193
-2.9	.00187	.00181	.00175	.00169	.00164	.00159	.00154	.00149	.00144	.00139
-3	.00135	.00131	.00126	.00122	.00118	.00114	.00111	.00107	.00104	.00100
-3.1	.00097	.00094	.00090	.00087	.00084	.00082	.00079	.00076	.00074	.00071
-3.2	.00069	.00066	.00064	.00062	.00060	.00058	.00056	.00054	.00052	.00050
-3.3	.00048	.00047	.00045	.00043	.00042	.00040	.00039	.00038	.00036	.00035
-3.4	.00034	.00032	.00031	.00030	.00029	.00028	.00027	.00026	.00025	.00024
-3.5	.00023	.00022	.00022	.00021	.00020	.00019	.00019	.00018	.00017	.00017
-3.6	.00016	.00015	.00015	.00014	.00014	.00013	.00013	.00012	.00012	.00011
-3.7	.00011	.00010	.00010	.00010	.00009	.00009	.00008	.00008	.00008	.00008
-3.8	.00007	.00007	.00007	.00006	.00006	.00006	.00006	.00005	.00005	.00005
-3.9	.00005	.00005	.00004	.00004	.00004	.00004	.00004	.00004	.00003	.00003
-4	.00003	.00003	.00003	.00003	.00003	.00003	.00002	.00002	.00002	.00002

Numerical

Consider the following series representing 44 computer chips which may be either Defective(D) or Acceptable(A). Based on the runs up and down, determine the hypothesis of independence for $\alpha = 0.05$.

D A A A A A A A D D D D A A A A A A A A D D A A A A A A A A

D D D D A A A A A A A A A A

soln: Here observed number of runs $x_r = 8$

Let n_1 represents number of Defective(D) chips and n_2 represents Acceptable(A) chips

So $n_1 = 11$ and $n_2 = 33$

Expected Number of runs $\mu_r = \frac{2n_1n_2}{n_1+n_2} + 1$

$= 17.6$

Similarly $\sigma_r = \sqrt{\frac{2n_1n_2(2n_1n_2 - n_1 - n_2)}{(n_1+n_2)^2\,(n_1+n_2-1)}} = 2.4387$

So $Z_0 = \frac{X_r - \mu_r}{\sigma_r} = -3.895$

Confidence Level	Alpha	Alpha/2	z alpha/2
90%	10%	5.0%	1.645
95%	5%	2.5%	1.96
98%	2%	1.0%	2.326
99%	1%	0.5%	2.576

Here $-Z_{\alpha/2} = -1.96$

Since $Z0 < -Z_{\alpha/2}$ (i.e. $-Z_{\alpha/2} \le Z_0 \le Z_{\alpha/2}$ not valid) the hypothesis for independence is rejected.

Auto-Correlation Test

- Autocorrelation is a statistical test that determines whether a random number generator is producing independent random numbers in a sequence.
- The tests for auto-correlation are concerned with the dependence between numbers in a sequence.
- The test computes the autocorrelation between every m numbers (m is also known as the lag) starting with the i^{th} number (i is also known as the index).
- Important variables to remember:
  1. m - is the lag, the space between the numbers being tested
  2. i - is the index, or the number in the sequence that you start with
  3. N - the number of numbers generated in a sequence
  4. M – is the largest integer such that $i + (M + 1)m \leq N$

Auto-Correlation Test Algorithm

1. Define the hypothesis.
2. Find the value of ‘i’ and lag value ‘m’
3. Using the value of ‘i’, ‘m’ and ‘N’ calculate the value of M as $i + (M + 1)m \leq N$ where
   a. m - is the lag, the space between the numbers being tested
   b. i - is the index, or the number in the sequence that we start with
   c. N - the number of numbers generated in a sequence
   d. M – is the largest integer such that $i + (M + 1)m \leq N$

4. Compute the test statistics as:

$$Z_0 = \frac{\rho_{in}}{\sigma_{\rho in}}$$

where $\rho_{in} = \frac{1}{M+1}\left[\sum_{k=0}^{M}\left(R_{(i + km)} * R_{(i + (k+1)m)}\right)\right] - 0.25$

$$\sigma_{\rho in} = \frac{\sqrt{13M+7}}{12(M+1)}$$

5. If $-Z_{\alpha/2} \leq Z_0 \leq Z_{\alpha/2}$ then the null hypothesis is not rejected.

Numerical

Consider a sequence of 30 numbers generated by a random number generator. Test whether the 3rd, 8th and 13th numbers in the sequence are auto-correlated with $\alpha = 0.05$ and $Z_{0.025} = 1.96$.

0.12, 0.01, 0.23, 0.28, 0.89, 0.31, 0.64, 0.28, 0.83, 0.93, 0.99, 0.15, 0.33, 0.35, 0.91

0.41, 0.60, 0.27, 0.75, 0.88, 0.68, 0.49, 0.05, 0.43, 0.95, 0.58, 0.19, 0.36, 0.69, 0.87

Soln: Let H_0 represent null hypothesis where H_0 : Numbers in sequence are auto-correlated.

Here m = 5

We have, $i + (M + 1)m \leq N$

or, $3 + (M + 1)*5 \leq 30$

or, $M \leq 4.4 \sim 4$

- We have, $\rho_{in} = \frac{1}{M+1}\left[\sum_{k=0}^{M}\left(R_{(i+km)} * R_{(i+(k+1)m)}\right)\right] - 0.25$

$$= \frac{1}{4+1}\left[\sum_{k=0}^{4}\left(R_{(i+km)} * R_{(i+(k+1)m)}\right)\right] - 0.25$$

$$= \frac{1}{5}\left[R_3 * R_8 + R_8 * R_{13} + R_{13} * R_{18} + R_{18} * R_{23} + R_{23} * R_{28}\right] - 0.25$$

$$= -0.1945$$

$$\sigma_{\rho in} = \frac{\sqrt{13M+7}}{12(M+1)} = \frac{\sqrt{13*4+7}}{12(4+1)} = 0.1280$$

Now $Z_0 = \frac{\rho_{in}}{\sigma_{\rho in}} = -1.581$

For $\alpha = 0.05$, $Z_{\alpha/2} = 1.96$

Since $-Z_{\alpha/2} \leq Z_0 \leq Z_{\alpha/2}$ then the null hypothesis is not rejected.
Hence the generated numbers are auto-correlated.

Methods of generating non-uniform Variables: Generating discrete distributions

- A discrete distribution describes the probability of occurrence of each value of a discrete random variable.
- A discrete random variable is a random variable that has countable values such as list of non-negative integers.
- When the discrete distribution is uniform, the requirement is to pick one of N alternatives with equal probability given to each.
- Given a random number U($0 \leq U < 1$), the process of multiplying by N and taking the integral portion of the product, which is denoted mathematically by the expression [UN], gives N different outputs. The output are the numbers 0,1,2,….,(N-1).
- The result can be changed to the range of values C to N+C-1 by adding C.

Methods of generating non-uniform Variables: Generating discrete distributions

- A discrete distribution describes the probability of occurrence of each value of a discrete random variable.
- A discrete random variable is a random variable that has countable values such as list of non-negative integers.
- When the discrete distribution is uniform, the requirement is to pick one of N alternatives with equal probability given to each.
- Given a random number U($0 \leq U < 1$), the process of multiplying by N and taking the integral portion of the product, which is denoted mathematically by the expression [UN], gives N different outputs. The output are the numbers 0,1,2,….,(N-1).

Methods of generating non-uniform Variables: Generating discrete distributions

- The result can be changed to the range of values C to N+C-1 by adding C.
- Generally, <u>the requirement is for a discrete distribution that is not uniform</u>, so that <u>a different probability is associated with each output</u>.

Number of Items X_i	Number of Customers N_i	Probability Distribution $P(X_i)$	Cumulative Probability Distribution
1	25	0.10	0.10
2	128	0.51	0.61
3	47	0.19	0.8
4	38	0.15	0.95
5	12	0.05	1

Generating discrete distributions

- Suppose, for example , it is necessary to generate a random variable representing the number of items bought by a customer at store , where the probability function is the discrete distribution given in previous table.
- A table is formed to list the number of items X , and the cumulative probability Y, as shown below:

Number of Items X	Probability P(X)	Cumulative Probability(Y)
1	0.10	0.10
2	0.51	0.61
3	0.19	0.8
4	0.15	0.95
5	0.05	1

Generating discrete distributions

- Taking the output of a uniform random number generator, U, the value is compared with the values of Y.
- If the value falls in an interval $Y_i < U \leq Y_{i+1}$ (i=0,1,…….,4), the corresponding value of X_{i+1} is taken as desired output.
- It is not necessary that the intervals be in any particular order.
- A computer routine will usually search the table from the first entry .
- The amount of searching can be minimized by selecting the intervals in decreasing order of probability

Generating discrete distributions

- For computer routine above data can be arranged as:

Probability	Cumulative Probability	Number of Items
0.51	0.51	2
0.19	0.70	3
0.15	0.85	4
0.10	0.95	1
0.05	1	5

- With this arrangement , 51% of the searches will only need to go to the first entry, 70% to the first or second and so on.
- With the original ordering, only 10% are satisfied with the first entry and only 61% with the first two

Inversion, Rejection and Composition - Inversion

- In the simplest case of inversion, we have a continuous random variable X with a strictly increasing distribution function F.
- Then F has an inverse F^{-1} defined on the open interval (0,1): for $0<u<1$, $F^{-1}(u)$ is the unique real number x such that $F(x)=u$ i.e.

$$F(F^{-1}(u))=u, \text{ and } F^{-1}(F(x))=x$$

$$P(F^{-1}(u)\leq x) = P(u\leq F(x)) = F(x)$$

- Let u~unif(0,1) denote a uniform random variable on (0,1) Then <u>$F^{-1}(u)$ has distribution function F.</u>

Inversion, Rejection and Composition - Inversion

- To extend this result to a general distribution function F, the generalized inverse of F is:

$$F^{-}_{0<u<1}(u) = \inf\{x: F(x) \geq u\}$$

Where inf represents the Infimum value(greatest lower bound value)

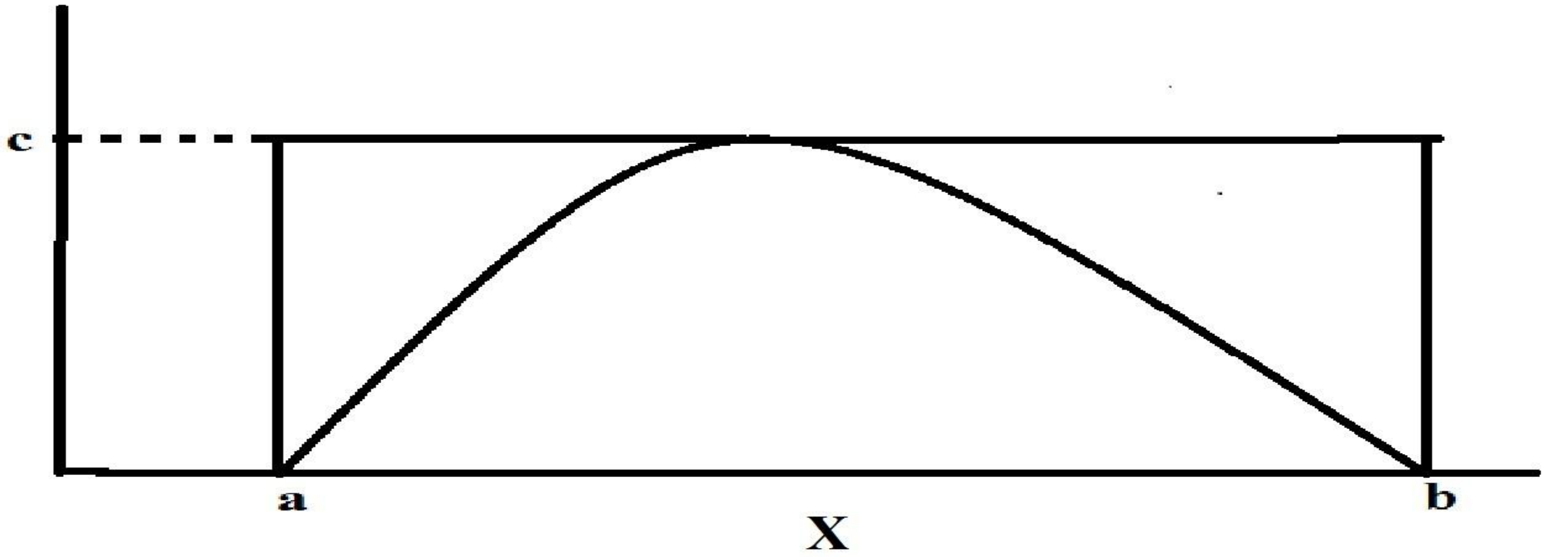
And $F^{-}(u)$ represents u- quartile

Inversion, Rejection and Composition - Rejection

- The rejection method is applied when the probability density function f(x), has a lower and upper limit to its range, lower bound a and b and an upper bound c respectively.
- The method can be specified as follows:
  - Compute the values of two independent uniformly distributed variates(a quantity having a numerical value for each member of group) U1 and U2.
  - Compute $X_0=a+U_1(b-a)$.
  - Compute $Y_0=cU_2$
  - Either accept X0 as the desired output otherwise repeat the process with two new uniform variates.

Inversion, Rejection and Composition - Rejection

- This method is closely related to the process of evaluating an integral using Monte-Carlo technique. The probability density function is enclosed in a rectangle with side lengths b-a and c.
- In the rejection method the curve is probability density function so that the area under curve must be 1 i.e. c(b-a)=1.



Inversion, Rejection and Composition - Rejection

- The probability of X being less than or equal to X_0 is by definition,

$$F(X_0) = \frac{\int_a^{X_0} f(x)dx}{c(X_0 - a)} * \frac{(X_0 - a)}{(b - a)}$$

- Since c(b-a)=1, it follows that

$$F(X_0) = \int_a^{X_0} f(x)dx$$

which shows that X0 has the desired distribution

Inversion, Rejection and Composition - Composition

- Sometimes the random variables X of interest involves the sum of n>1 independent random variables:

$$X=Y_1+Y_2+Y_3+\ldots+Y_n$$

- To generate a value for X, we can generate a value for each of the random variables Y_1, Y_2, Y_3, Y_n and add them together. This is called composition.
- Composition can also be used to generate random numbers that are approximately normally distributed.
- The normal distribution is one of the most important and frequently used continuo
- The notion N(μ , σ) refers to the normal distribution with mean μ and variance σ^2.

Inversion, Rejection and Composition - Composition

- The central limit theorem in probability says that if Y_1, Y_2, Y_3, ….Y_n are independent and identically distributed random variables with mean μ and positive variance σ^2, then random variable is

$$Z = \frac{\sqrt{n}(\frac{x}{n} - \mu)}{\sigma}$$

Convolution Method

- The probability distribution of a sum of two or more independent random variables is called a convolution of the distributions of the original variables.
- The convolution method thus refers to adding together two or more random variables to obtain a new random variable with the desired distribution.
- Technique can be used for all random variables X that can be expressed as the sum of n random variables

$$X = Y1 + Y2 + Y3 + . . . + Yn$$

- In this case, one can generate a random variate X by generating n random variates, one from each of the Yi, and summing them.

Numerical

Use Chi-Square test to test the uniformity of following random numbers for 95% confidence level And given critical value for degree of freedom = 8 is 15.51.

25	33	5	54	9	31	14	40	17	52	33
49	61	62	26	67	6	28	55	22	68	34
50	2	66	77	86	12	41	88	19	96	70
81	47	85	3	59	94	8	42	71	37	79
82	51	91	11	75	43	39	44	64	58	46

Chapter 7
Verification and Validation of Simulation Models

Model Building

- Model building is an iterative process of domain knowledge acquisition and model development.
- The real system and their interactions among various components should be analyzed.
- The domain knowledge can also be acquired from the interaction with concerned people.
- As the model development proceeds, new questions arises and the process of learning system behaviour and structure takes place again.
- Then a conceptual model is constructed with a collection of assumptions and hypotheses and finally developed into a logical model.
- Finally, the logical model is implemented using various simulation software.

Model Verification

- In the context of computer simulation, verification of a model is the process of confirming that it is correctly implemented.
- Verification is concerned with building the model correctly(concerned with building the **model right.).**
- The objective of model verification is to ensure that the implementation of the model is correct.
- Verification answers for Is the developed model performing properly?
- During verification the model is tested to find and fix errors in the implementation of the model.
- Various processes and techniques are used to assure the model matches specifications and assumptions with respect to the model concept.
- If the input parameters and logical structure of the model are **correctly represented,** verification is completed.

Considerations to be used in Verification Process:

1. The operational model should be checked by someone other than the developer, preferably an expert in the simulation software being used.
2. Make a flow diagram that contains all logically possible action a system can perform when an event occurs.
3. Closely examine the model output for reasonableness under variety of input parameters.
4. Have the operational model print input parameters when the simulation ends to check if they are not altered.
5. Make the operational model as self-documenting as possible.
6. Verify animated operational model imitates the actual system.
7. Debugger should be used during simulation model building.
8. Graphical interfaces are recommended.

Model Validation

- Validation checks the accuracy of the model's representation of the real system.
- Validation is the determination that a model is an accurate representation of the real system.
- It attempts to confirm that a model represents the real system accurately.
- Validation is usually achieved through the calibration of the model.
- Calibration is the iterative process of comparing the model with real system, revising the model if necessary, comparing again, until a model is accepted (validated).
- This process is repeated until acceptable accuracy for model is achieved.

Calibration of Model

- Validation is a process of comparing the model and its behavior to the real system and its behavior.
- Calibration is the iterative process of comparing the model with real system, revising the model if necessary, comparing again, until a model is accepted (validated).
- Calibration deals with adjustment of the parameters of the model in order acquire the desired accuracy.
- The comparison of the model to reality is carried out by subjective and objective tests.
- A subjective test involves talking to people, who are knowledgeable about the system, people or experts having idea on making models and forming the judgment.
- Objective tests involve one or more statistical tests to compare some model output with the assumptions in the model.

Naylor and Finger Approach

- Naylor and Finger formulated a three step approach to the validation process.

1. **Build model with high Face Validity**

- Face validity, also called logical validity, is a simple form of validity where you apply a **superficial and subjective assessment** of whether or not your study or test measures what it is supposed to measure.
- A model should appear reasonable on its face to model users and to those who knows about the real system that is being simulated.
- A model should be designed with high degree of realism regarding system structure and behavior through reliable data.
- The potential users should also be involved in the validation process to aid in identification of model deficiencies and optimizing those deficiencies to produce better model. This process is termed as structural walkthrough.

- Sensitivity analysis is also used for face validity of the model. It analyses the effect on output when there is change in input parameters.
- Sensitivity analysis is done through appropriate statistical techniques.

2. Validate Model Assumptions

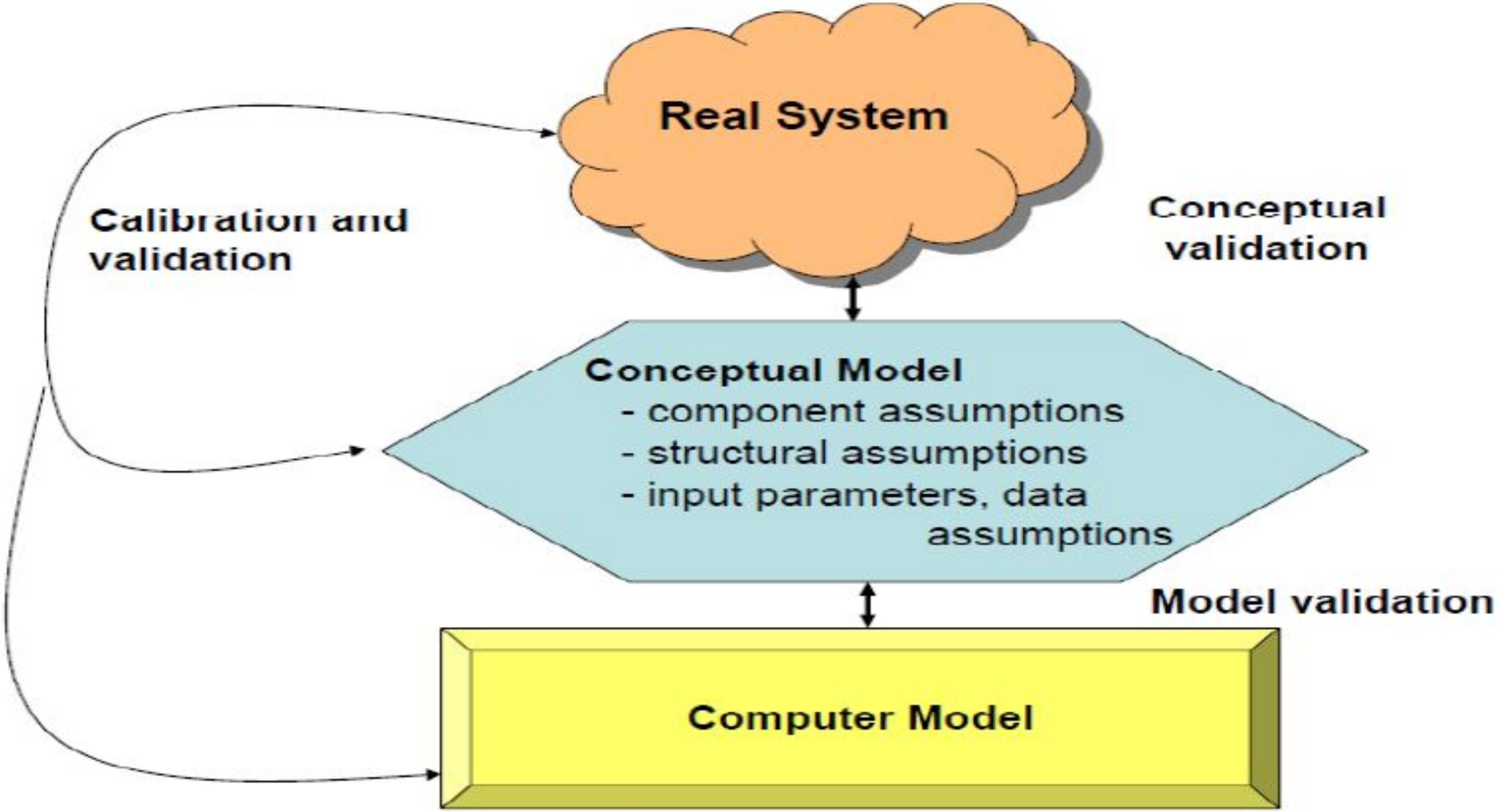
- Sensitivity analysis is done through appropriate statistical techniques. These assumptions are of two types: **structural assumption** and **data assumption**.
- Structural assumptions involves simplification and abstraction of reality.
- Structural assumptions deal with such questions as how the system operates, what kind of model should be used, queueing, inventory, reliability, and others.
- Data assumptions should be done based on collection of reliable data.
- Data assumptions deal with such questions as : what kind of input data model is? What are the parameter values to the input data model?

- For example - Consider a bank system in which customer are queued before providing service.
- The structural assumptions may be - Customers form a single queue which is served by multiple tellers - Customers form one line for each teller - The number of teller should be fixed or variable.
- These structural assumptions should be validated by actual observation during appropriate time periods and also by discussions with the managers and tellers.
- The data assumptions may be - interarrival times of customers during peak hour - interarrival times of customers during slack period - service time for personal accounts and so on.
- These data assumptions should be validated by consultation with bank managers. The validation is done by using goodness-of-fits tests such as chi-square test or Kolmogorov-Smirnov tests.

3. Validating Input-Output Transformation

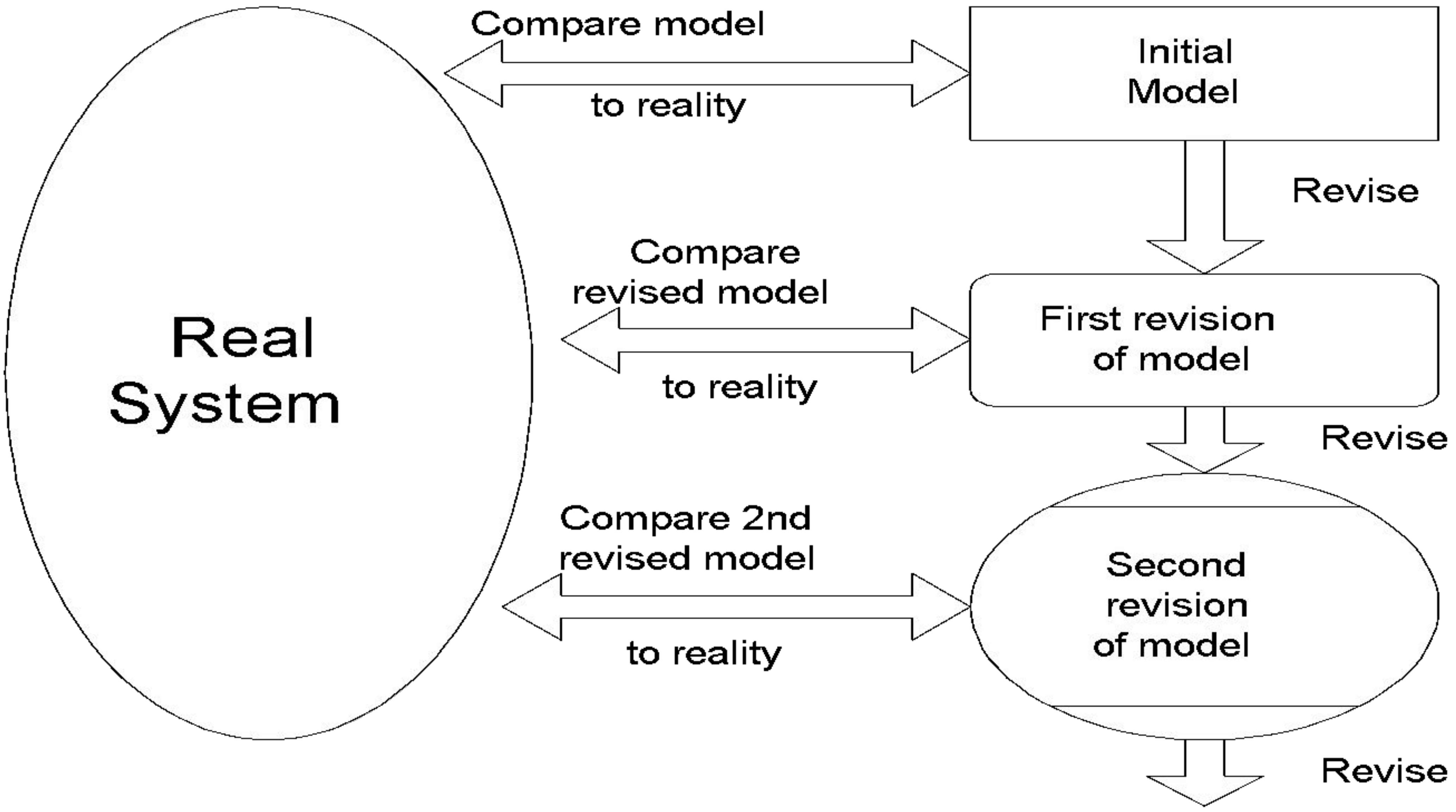
- It involves validating whether the model can predict the future behavior of the real system when the model input data match the real inputs and when a policy implemented in the model is implemented at some point in the system.
- Here the model is viewed as a black box.
- We feed the input at one end and examine the output at the other end.
- Use the same input for a real system, compare the output with the model output. If they fit closely, the black box seems working fine. Else something is wrong.
- If in future, the model is used for different purpose, it should be revalidated in terms of new response of interest.

From Model Building to Validation



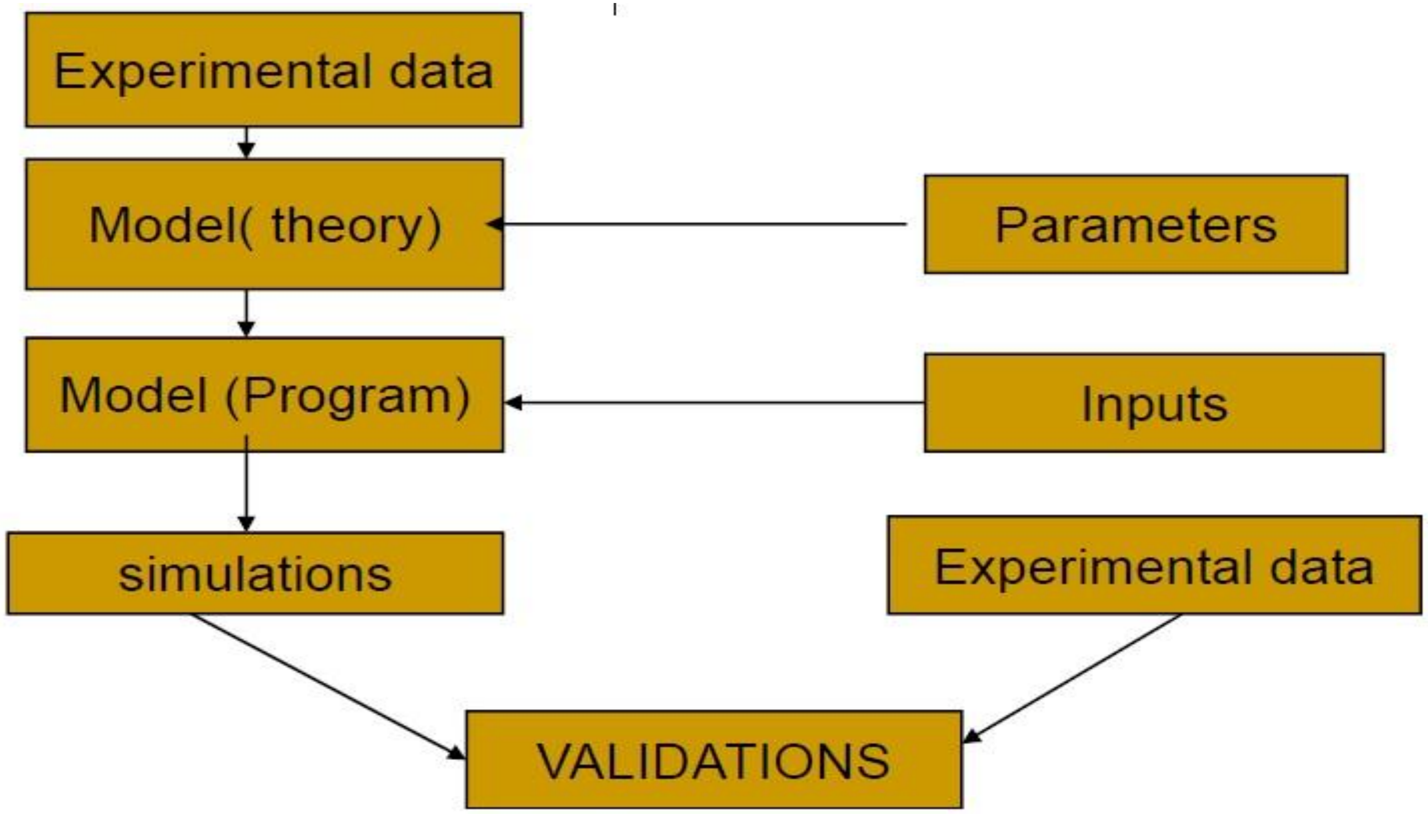
Iterative Process of Calibrating a Model

- Iterative calibration means to validate the model with the real system, look out for the places for betterment of the models and revising the model to form next better model repeatedly until a satisfiable model is not achieved.
- The initial model is developed and is calibrated using Naylor-Finger calibration steps with the real system. It is then revised and a first revision model is generated.
- The first revision model is then calibrated with the real system. It is revised to form a second revision model.
- This process is continued until the model becomes acceptable.


Real System
Compare model
to reality
Initial Model
Revise
Compare revised model
to reality
First revision of model
Revise
Compare 2nd revised model
to reality
Second revision of model
Revise

Model Validation

- Model validation is a necessary requirement for model application.
- To do a reliable validation, several steps must be taken and each of them may be a source of errors which will influence the final result.



Validation Errors

- During the validation phase, errors might be present.
- As a general rule, if there are discrepancies between observed and simulated data, the technical structure of a model should be the last factor to suspect.
- Following may be some reasons for errors:

1. **Model inadequate:** Model Adequacy deals with the following questions:
    - Are all the important processes for a given environment included?
    - Are the processes modeled correctly?
    - Was the range of data used to develop model components for process simulation wide enough to include our conditions?

2. **Lack of calibration:** Calibration should be done in order to adjust the parameters of the model so as to acquire the desired accuracy.

3. Errors in the code

 Errors might be present on the code that we write.

 Following steps can be taken to check a code:

a. Do manual calculations for instance using a spreadsheet and compare with model results.

b. Verify that simulation results are within the known physical and biological reality.

c. Run simulations with highly contrasting inputs.

4. Errors in the inputs

 There might be error while providing the input data or parameters.

5. Errors in the use

 There might be error while using the model.

6. Errors in Experimental Data

- Experimental data are used to test the predictive capabilities of model.
- These experimental data are affected by experimental error, which can be large.
- Only a large number of experimental data allows a meaningful evaluation of model performance in statistical terms.

Chapter 8
Analysis of Simulation Output

Simulation Output Introduction

- Whenever a random variable is introduced to the simulation model, all the system variables that describe its behavior become random or stochastic.
- The values of the variables involved in the system will fluctuate as the simulation proceeds.
- So, arbitrary measurement of the values of these variables can not represent the true value.
- In simulation study, it is assumed that the observations being made are mutually independent. But, in most of the real world problems, simulation results are mutually dependent.
- The various methods used to analyze simulation results are as follows:
  1. Estimation Methods
  2. Simulation Run Statistics
  3. Replication of Runs
  4. Elimination of Initial Bias

Estimation Methods

- Estimation Method estimates the range for the random variable so that the desired output can be achieved.
- It is assumed that the random variables are stationary and independent drawn from an infinite population with a finite mean μ and finite variance σ^2.
- Such random variables that meet all these conditions are called Independently and Identically Distributed (IID) random variable.
- The central limit theorem can be applied to IID data. It states that "the sum of n numbers of IID variables, drawn from a population that has a mean of μ and a variance of σ^2, is approximately distributed as a normal variable with a mean of $n\mu$ and a variance of $n\sigma^2$."
- Here Sample variable and time does not affect population distribution .

- Let x(i) be the n IID random variables i.e. random variables drawn from a sample of population with mean μ and variance σ^2, where (i=1,2,…,n) .
- Then

$$Z = \frac{\sum_{i=1}^{n} x_i - n\mu}{\sigma\sqrt{n}}$$

- In terms of sample mean

$$Z = \frac{\bar{x} - \mu}{\sigma/\sqrt{n}}$$

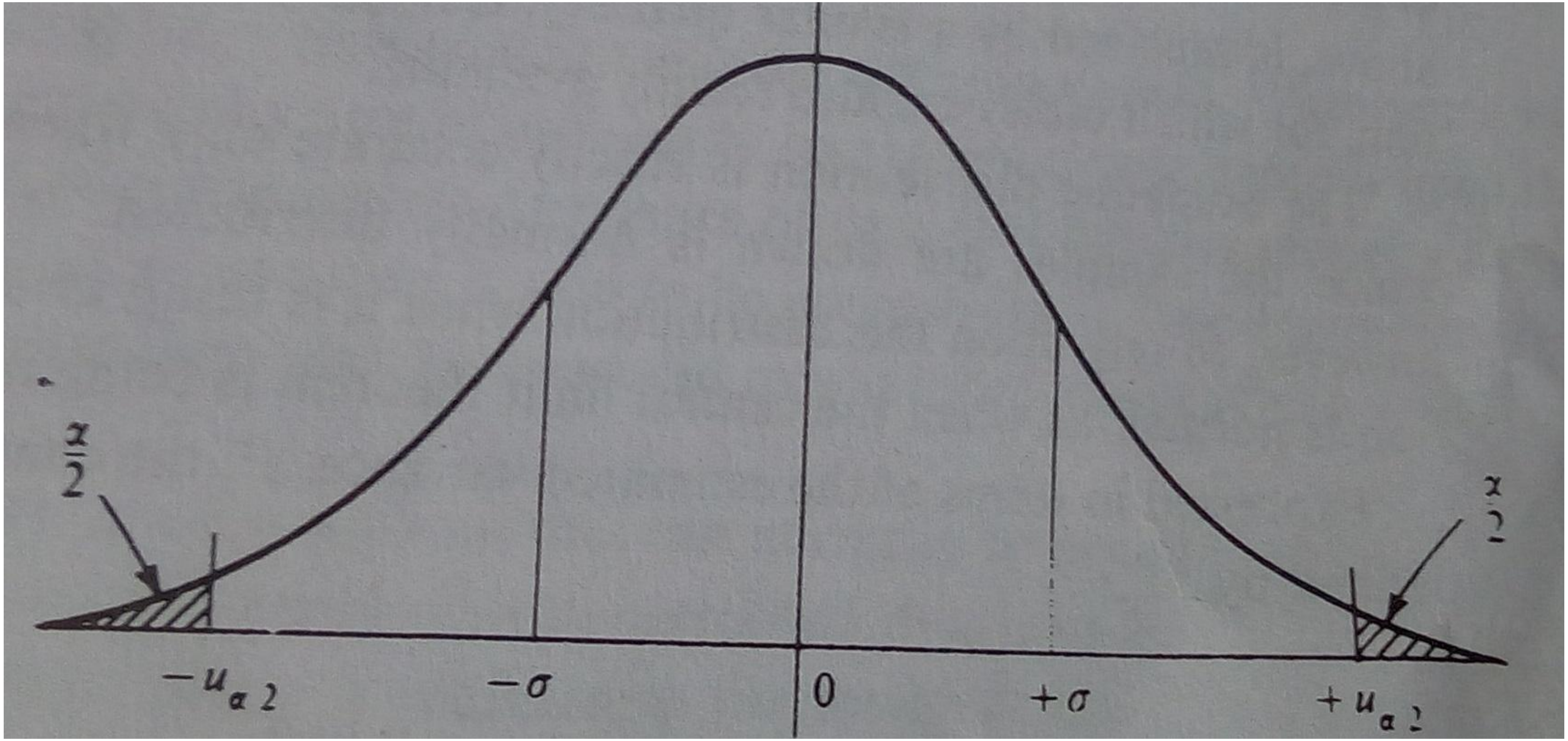


Figure: Probability Density Function of Standard Normal Variate

- For some constant α, known as level of significance, the probability that z lies between $-\mu_{\alpha/2}$ and $\mu_{\alpha/2}$ is given by:

$$\text{Prob}\{-\mu_{\alpha/2} \leq z \leq \mu_{\alpha/2}\} = 1 - \alpha$$

- In case of sample mean, this probability statement can be written as:

$$\text{Prob}\{\bar{x} + \frac{\sigma}{\sqrt{n}}\mu_{\alpha/2} \geq \mu \geq \bar{x} - \frac{\sigma}{\sqrt{n}}\mu_{\alpha/2}\} = 1 - \alpha$$

- The constant 1- α is call confidence level and the confidence interval is

$$\bar{x} \pm \frac{\sigma}{\sqrt{n}}\mu_{\alpha/2}$$

- Typically, the confidence level might be 90% in which case $\mu_{\alpha/2}$ is 1.65.
- Here the population variance σ^2 is usually not known. In this case it is can be replaced by an estimate calculated from the formula

$$s^2 = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})^2$$

- The normalized random variable based on σ^2 is replaced by a normalized random variable based on s^2.This has a **t distribution**, with n-1 degrees of freedom.
- The quantity $\mu_{\alpha/2}$ used in the distribution of a confidence interval given above is represented by a similar quantity $t_{n-1,\alpha/2}$ based on t-distribution.
- The t-distribution is strictly accurate only when the population from which the samples are drawn is normally distributed.

- Expressed in terms of the estimated variance s^2, the confidence interval for $\bar{x}$ is defined by

$$\bar{x} \pm \frac{s}{\sqrt{n}} t_{n-1,\alpha/2}$$

- Hence the estimation method gives the desired range of the sample variable taken from infinite population.

Example:

- Let us consider x(i) where i = 1, 2, 3, …………, n be the n number of random variables drawn from a sample of population with mean μ and variance σ^2.
- Using central limit theorem, and transforming to standard normal distribution, we get:

$$Z = \frac{\sum_{i=1}^{n} x_i - n\mu}{\sigma\sqrt{n}}$$

- Dividing top and bottom by n, we get:
- $Z = \frac{\bar{x}-\mu}{\sigma/\sqrt{n}}$ where $\bar{x}(n) = \frac{1}{n}\sum_{i=1}^{n} x_i$ = sample mean

Simulation Run Statistics

- One of the measure to analyze simulation result.
- This approach is used to obtain independent results by repeating the simulation.
- In most of the simulation study, the assumptions of stationary and mutually independent observations do not apply. An example of such case is queuing system.
- Correlation is necessary to analyze such scenario. In such cases, simulation run statistics method is used.
- Consider a single-server system in which the arrivals occur with a Exponential distribution and the service time has an exponential distribution.
- Suppose the study objective is to measure the mean waiting time, defined as the time entities spend waiting to receive service and excluding the service time itself.

- This system is commonly denoted by M/M/1 which indicates:
  - a. the inter-arrival time is distributed exponentially
  - b. the service time is distributed exponentially
  - c. there is one server
- In a simulation run, the simplest approach is to estimate the mean waiting time by accumulating the waiting time of n successive entities and dividing by n.
- This measure is denoted by $\bar{x}(n)$ which emphasizes the fact that its value depends on the number of observations taken.
- If x_i for i=1,2,….,n are the individual waiting times then

$$\bar{x}(n) = \frac{1}{n}\sum_{i=1}^{n} x_i$$

- Whenever a waiting line forms, the waiting time of each entity on the line clearly depends upon the waiting time of its predecessors.
- Any series of data that has this property of having one value affect other values is said to be autocorrelated.
- The sample mean of autocorrelated data can be shown to approximate a normal distribution as the sample size increases.
- The value of $\bar{x}(\mathrm{n}) = \frac{1}{n}\sum_{i=1}^{n} x_i$ remains a satisfactory estimate for the mean of autocorrelated data.
- A simulation run is started with the system in some initial state, frequently the idle state, in which no service is being given and no entities are waiting.
- The early arrivals then have a more than normal probability of obtaining service quickly. So a sample mean that includes the early arrivals will be biased.

- For a given sample size starting from a given initial condition, the sample mean distribution is stationary.
- But if the distributions could be compared for different sample sizes, the distribution would be slightly different.
- The following figure is based on theoretical results, which shows how the expected value of sample mean depends upon the sample length, for the M/M/1 system, starting from an initial empty state, with a server utilization of 0.9.

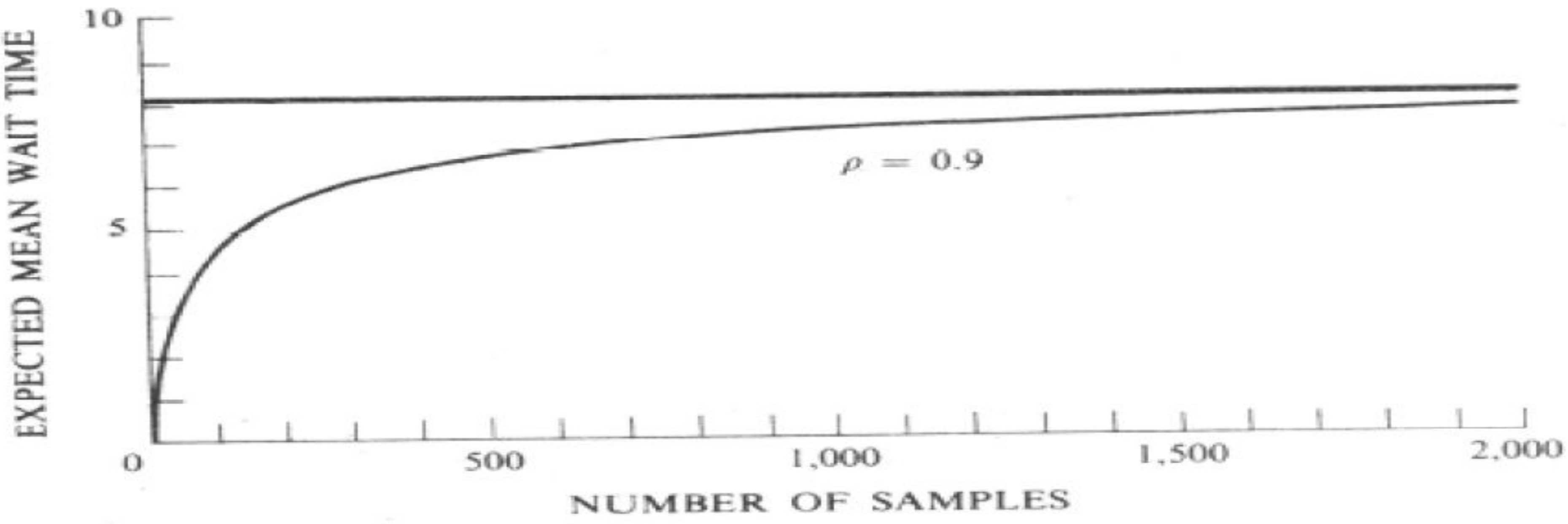


Figure: Mean waiting time in M/M/1 system for different sample size

Example

- Consider a system with Kendall's notation M/M/1/FIFO and the objective is to measure the mean waiting time.
- In simulation run approach, the mean waiting time is estimated by accumulating the waiting time of n successive entities and then it is divided by n. This measures the sample mean such that:
- $\bar{x}(\mathrm{n}) = \frac{1}{n}\sum_{i=1}^{n} x_i$
- Such series of data in which one value affect other values is said to be auto-correlated.
- The sample mean of auto-correlated data can be shown to approximate a normal distribution as the sample size increases.

Problem that may arise in Simulation Run Statistics

- The distribution may not be stationary.
- A simulation run is started with the system in some initial idle state. In this case, the early arrivals will obtain service quickly deviating from normal distribution. Hence, the sample means of the early arrivals is known as **initial bias**.
- As the sample size increases and the length of run is long, the effect of bias dies and the normal distribution is again established.

Replication of Runs

- The precision of results of a dynamic stochastic can be increased by repeating the experiment with different random numbers strings.
- For each replication of a small sample size, the sample mean is determined.
- The sample means of the independent runs can be further used to estimate the variance of distribution.
- Let X_{ij} be the i^{th} observation in j^{th} run, then the sample mean and variance for the j^{th} run are:

$$\overline{x_j}(n) = \frac{1}{n}\sum_{i=1}^{n} x_{ij}$$

$$s_j^2 = \frac{1}{n-1}\sum_{i=1}^{n}[x_{ij} - \overline{x_j}(n)]^2$$

- When we have similar means and variances for m independent measurements, we can obtain the combined mean and variance for the population by combining them.
- The mean of the means and the mean of the variances are then used to estimate the confidence interval.
- Combining the results of p independent measurements gives the following estimates for the mean waiting time $\bar{x}$ and variance s^2 of the population:

$$\overline{x} = \frac{1}{p} \sum_{j=1}^{p} \overline{x_j(n)}$$

$$s^2 = \frac{1}{p} \sum_{j=1}^{p} s_j^2(n)$$

- In repetitions of run, the length of run of replications is so selected that when length of all runs are combined, it comes to be equal with the sample size N. i.e. $p*n = N$
- By increasing the number of replications and shortening their length of run, the confidence interval can be narrowed. Confidence interval is the range of possible values for the parameter based on a set of data (e.g. the simulation results.)
- In the replication of simulation runs, if the number of runs is increased at the cost of shortening the individual runs, the estimate of the mean will be more biased.
- The results obtained will not be accurate.
- Thus, a compromise has to be made. However, it is suggested that the number of replications should not be very large, and that the sample means should approximate a normal distribution.

Elimination of Initial Bias

- A simulation run is started with the system in some initial idle state. In this case, the early arrivals will obtain service quickly deviating from normal distribution. Hence, the sample means of the early arrivals is known as **initial bias**.
- The initial bias in simulation should be removed.
- Following approaches can be used to remove initial bias:

1. Ignore the initial bias occurred during the simulation run i.e. the first part of the simulation can be ignored.
2. The system should be started in a more representative state than in the empty state.
3. Start the simulation in the empty state, then stop after initial bias and then start again.
4. Run the simulation for such a long period of time so that the initial bias has no any significance in the output result.

- The ideal situation is to **know the steady state distribution for the system**, and **select the initial condition** from that distribution.
- In most of the existing systems, there may be information available on the expected conditions that makes it feasible to select better initial conditions.
- The most common approach to remove the initial bias is to illuminate the initial section of the run.
- The run is started from an idle state and stopped after a certain period of time.
- The entities existing in the system at that time are left as they are.
- The run is then restarted with the statistics being gathered from the point of restart.
- It is usual to program the simulation so that statistics are gathered from the beginning, and simply wipe out the statistics gathered up to the point of restart.

- No simple rules can be given to decide how long an interval should be eliminated.
- The disadvantage of eliminating the first part of a simulation run is that the estimate of the variance, needed to establish a confidence limit, must be based on less information.
- The reduction in bias, therefore, is obtained at the price of increasing the confidence interval size.
- We can also run the simulation for such a long period of time so that the initial bias has no any significance in the output result.

Chapter 9
Simulation Software

Simulation Software

- Simulation software is a program that allows the user to **observe an operation through simulation** without actually performing that operation
- Simulation software helps us predict the behavior of a system.
- We can use simulation software to evaluate a new design, diagnose problems with an existing design, and test a system under conditions that are hard to reproduce, such as a satellite in outer space.
- Simulation software also includes visualization tools, such as data displays and 3D animation, to help monitor the simulation as it runs.
- Engineers and scientists use simulation software for a variety of reasons:
  - Creating and simulating models is less expensive than building and testing hardware prototypes.
  - We can use simulation software to test different designs before building one in hardware.
  - We can connect simulation software to hardware to test the integration of the full design.

History of Simulation Software

1955 - 1960 The Period of Search

- Search for unifying concepts and the development of reusable routines to facilitate simulation.
- Mostly conducted in FORTRAN

1961 - 1965 The Advent

- The Simulation Programming Language in use today appeared in this period.
- The first process interaction SPL(Simulation Programming Language), GPSS was developed at IBM.
- GPSS got popularity due to easy in use.

1966 - 1970 The Formative Period

- Concepts were reviewed and refined to promote a more consistent representation of each language’s worldview.
- In this phase due to rapid hardware advancements and user demands GPSS was forced to undergo major revision.

1971 - 1978 The Expansion Period

- Major advances in GPSS came from outside IBM
- Attempts were made to simplify modeling process.
- GPSS/NORDEN, a pioneering effort that offered an interactive, visual online environment
- GPSS/H(1997): For IBM mainframes, later for microcomputers and PC.
- GASP-IV(1971): It uses state events in addition to time events.

1979 - 1986 The Period of Consolidation and Regeneration

- Beginnings of PSLs written for, or adapted to, desktop computers and microcomputers.
- Two major descendants of GASP appeared: SLAM II and SIMAN(provide multiple modeling perspectives and combined modeling capabilities).

1987 – Now The Period of Integrated Environments

- Growth of SPLs on the personal computer and the emergence of simulation environments with graphical user interfaces, animation and other visualization tools.
- Many of these environment also contain input and output data analyzer.
- Recent advancements have been made in web-based simulation.

 Three types of software for simulation models developments:

1. General-purpose programming languages, e.g., Java, C.
   a. Not specifically designed for use in simulation.
   b. Simulation libraries, e.g., SSF, are sometimes available for standardized simulation functionality.
   c. Helps to understand the basic concepts and algorithms.
2. Simulation programming languages, e.g., GPSS/HTM, SIMAN V® and SLAM II®.
   a. Designed specifically for simulation of certain systems, e.g. queuing systems.
3. Simulation environment, e.g., Arena, AutoMod.
   a. Output analyzer is an important component, e.g. experimental design, statistical analysis.
   b. Many packages offer optimization tools as well.

Characteristics of Simulation Software

 Common characteristics:

- Graphical user interface, animation. For animation, some emphasize scale drawings in 2-D or 3-D; others emphasize iconic-type animation.
- Automatically collected outputs.
- Most provide statistical analyses, e.g., confidence intervals.

Advice for Selecting Simulation Software

 Following points are advised to be considered while selecting simulation software:

1. The language chosen should be ease of learning and using.
2. The accuracy should be high.
3. Execution speed is important.
4. It should have vendor support, and applicability to our applications
5. Beware of advertising claims and demonstrations.
6. Should provide Model status and statistics. Should provide standardized report and statistical analysis.
7. Tutorials and documentations should be available for the simulation software.
8. Runtime Environment: It is the feature that determines how the model acts during the simulation run. It includes execution speed, model size, interactive debugger, model statistics and so on.

Simulation In Java

- Java is widely used programming language that has been used extensively in simulation.
- It does not provide any modules directly aimed for simulation system.
- There are runtime libraries in java that provides random number generators.
- The components that all the simulation models written in JAVA are as follows:

1. Clock : It is a variable that defines the simulated time.
2. Initialization Method : It is a method to define the system state at initial time.
3. Min-time event method : It is a method that identifies the imminent (about to happen) event.
4. Event method: It is a method for each event that update the system state when it occurs.

5. Random variate generator : It is a method to generate random samples from the desired probability distributions.

6. Main program : It is the core of the simulation system that controls the overall event scheduling algorithms.

7. Report generator : It is a method that summarizes the collected statistics to give the report at the end of the simulation.

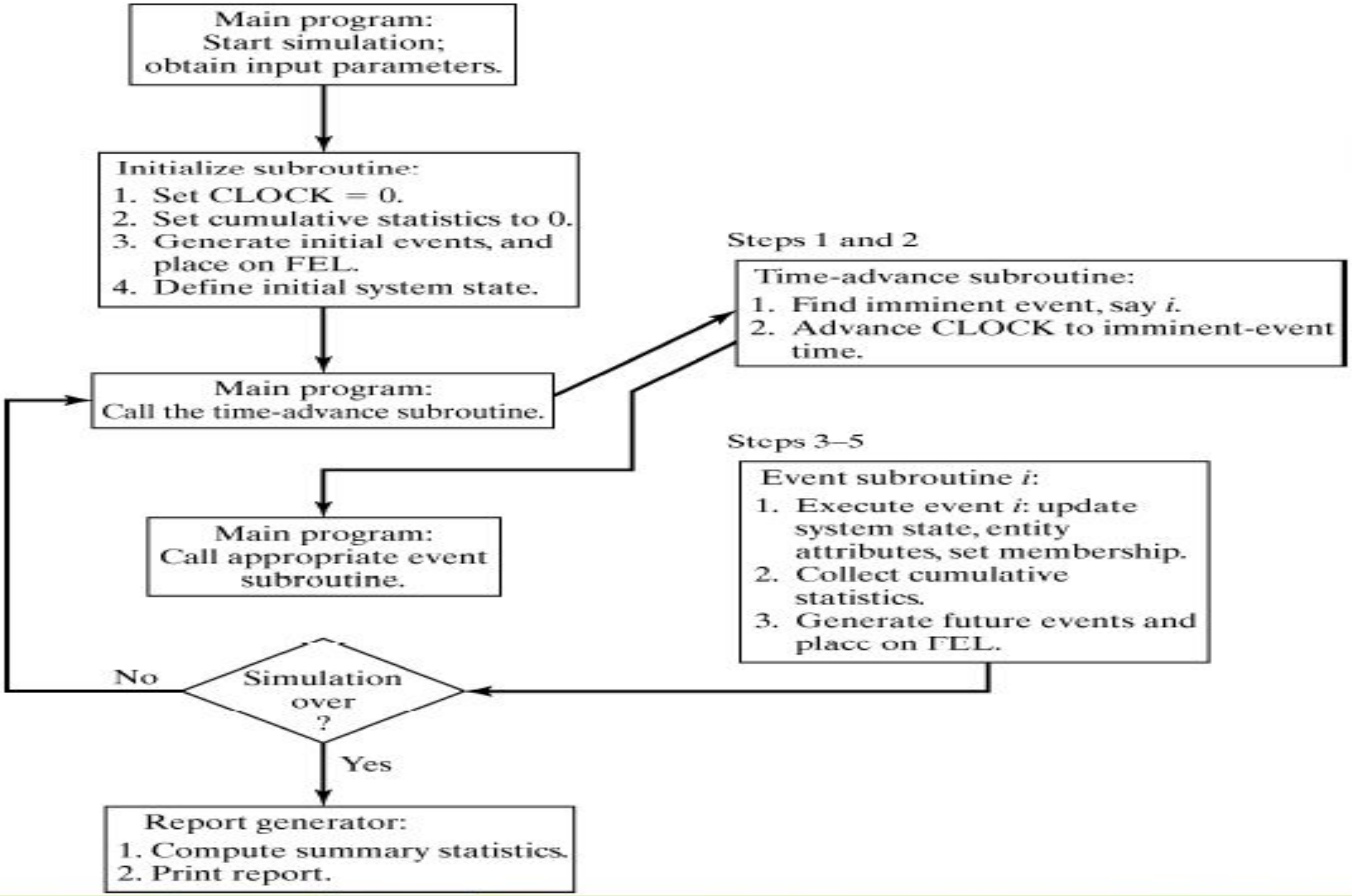


Figure: Overall Structure of Java Simulation

Explanation of Flowchart

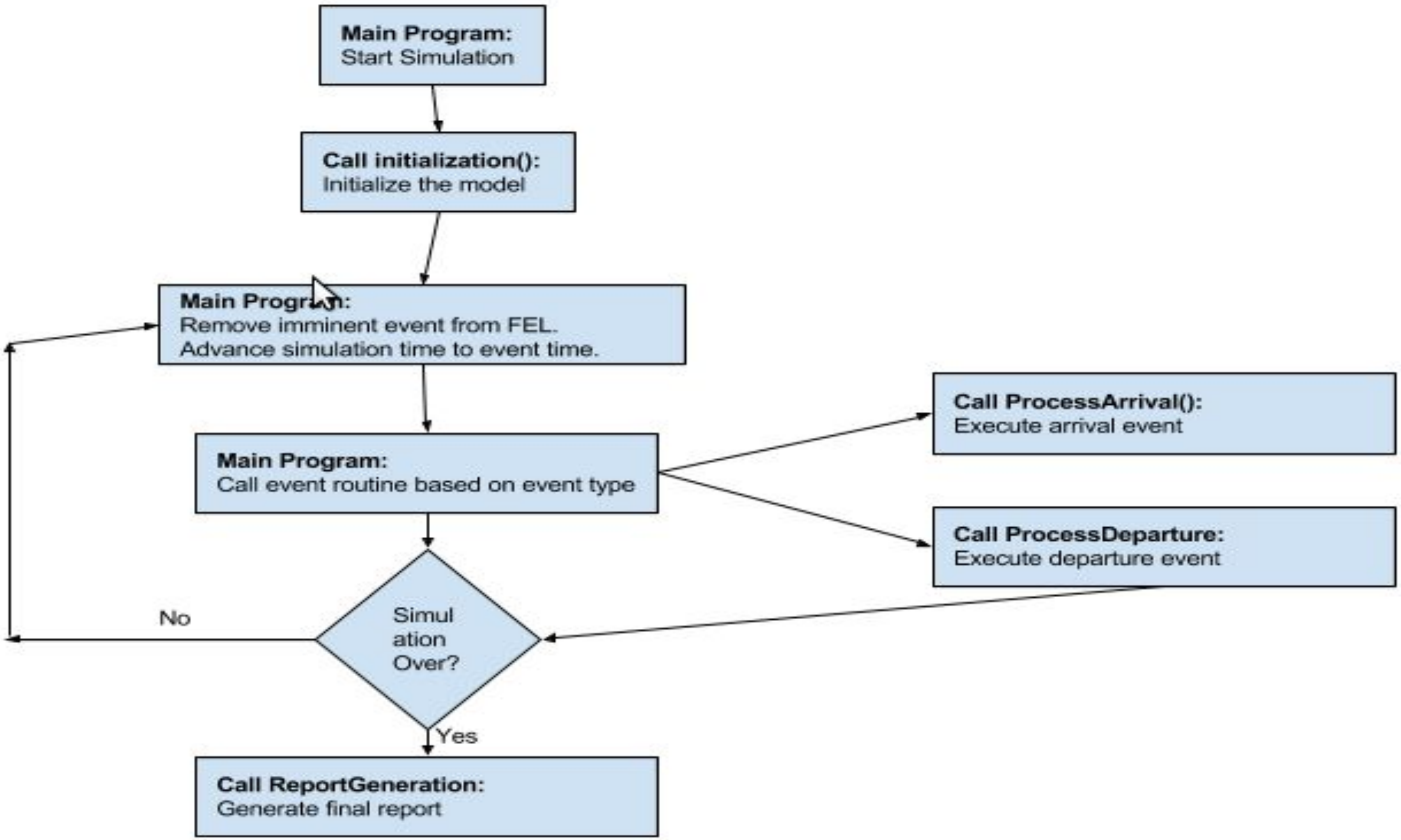
1. Simulation begins by setting clock to zero, initializing cumulative statistic to zero, generating any initial events and placing them in the Future Event List (FEL).
2. The simulation program then cycles repeatedly passing the current least time event to approximate event methods until the simulation is over.
3. At each step, clock is advanced to the time of the imminent event after finding the imminent event but before calling the event method.
4. The appropriate event method is called to execute imminent event, update cumulative statistics and generate future events.
5. All actions in an event method takes place at one instant of the simulated time.
6. When the simulation is over, the control passes to the report generator.

Example:

Single Server Queue Simulation

- Consider a grocery checkout counter. The simulation will run until 1000 customers have been serviced. It is assumed that the inter-arrival times are exponentially distributed with mean 4.5 minutes and the service times are normally distributed with mean of 3.2 minutes and standard deviation of 0.6 minutes. When cashier is busy, a queue forms with no customers turned away.

1. Class Event represents an event that stores code for arrival or departure and the event time stamp.
2. It consists of associated methods for creating an event and accessing its data.
3. It also has method compareTo that compares the event with another and reports whether the first event should be considered smaller, equal or greater than the argument event.



Main Program:
Start Simulation
Call initialization():
Initialize the model
Main Program:
Remove imminent event from FEL.
Advance simulation time to event time.
Main Program:
Call event routine based on event type
Call ProcessArrival():
Execute arrival event
Call ProcessDeparture:
Execute departure event
No
Simul
ation
Over?
Yes
Call ReportGeneration:
Generate final report

Variables Used

- System state: QueueLength ,NumberInService
- Entity attributes and set: Customers (FCFS queue of customers)
- Future event List: FutureEventList
- Activity durations: MeanInterArrivalTime, MeanServiceTime
- Input parameters: MeanInterArrivalTime, MeanServiceTime, SIGMA (standard deviation), TotalCustomers (The stopping criterion)
- Simulation variables: Clock
- Statistical accumulators: LastEventTime ,TotalBusy, Max QueueLength, SumResponseTime, NumberOfDepartures ,LongService (who spends 4 or more minutes)
- Summary statistics: RHO=BusyTime/Clock Proportion of time server is busy AVGR average response time ,PC4 proportion of customers who spent 4 or more minutes.

 Functions Used:

1. exponential(mu)
2. normal(xmu, Sigma)

 Methods Used:

1. Initialization()
2. ProcessArrival()
3. ProcessDeparture()
4. ReportGeneration()

Single Server Queue Example

```
class Sim {
    public static double Clock, MeanInterArrivalTime, MeanServiceTime,
    SIGMA, LastEventTime, TotalBusy, MaxQueueLength, SumResponseTime;
    public static long NumberOfCustomers, QueueLength, NumberInService,
    TotalCustomers, NumberOfDepartures, LongService;
    public final static int arrival = 1;
    public final static int departure = 2;
    public static EventList FutureEventList;
    public static Queue Customers;
    public static Random stream;
```

```
public static void main(String args[]) {
    MeanInterArrivalTime = 4.5;  MeanServiceTime = 3.2;
    SIGMA = 0.6; TotalCustomers = 1000; long seed =1000;
    stream = new Random(seed); // initialize rng stream
    FutureEventList = new EventList();
    Customers = new Queue();
    Initialization();
    // Loop until first "TotalCustomers" have departed
    while(NumberOfDepartures < TotalCustomers ) {
        Event evt = (Event)FutureEventList.getMin(); // get imminent event
        FutureEventList.dequeue(); // be rid of it
        Clock = evt.get_time(); // advance simulation time
        if( evt.get type() == arrival ) ProcessArrival(evt);
        else ProcessDeparture(evt);
        }
    ReportGeneration();
    }
}
```

Initialization Method

```
public static void Initialization() {
    Clock = 0.0;
    QueueLength = 0;
    NumberInService = 0;
    LastEventTime = 0.0;
    TotalBusy = 0 ;
    MaxQueueLength = 0;
    SumResponseTime = 0;
    NumberOfDepartures = 0;
    LongService = 0;
    // create first arrival event
    Event evt = new Event(arrival, exponential( stream, MeanInterArrivalTime));
    FutureEventList.enqueue( evt );
}
```

Arrival Event Method

Steps involved:

- Update server status
- Collect statistics
- Schedule next arrival

```
public static void ProcessArrival(Event evt) {
      Customers.enqueue(evt);
      QueueLength++;
// if the server is idle, fetch the event, do statistics and put into service
      if( NumberInService == 0)  ScheduleDeparture();
      else TotalBusy += (Clock - LastEventTime);  // server is busy
```

```
// adjust max queue length statistics
    if (MaxQueueLength < QueueLength) MaxQueueLength = QueueLength;
// schedule the next arrival
    Event next_arrival = new Event(arrival, Clock + exponential(stream,
            MeanInterArrivalTime));
    FutureEventList.enqueue( next arrival );
    next_LastEventTime = Clock;
}
```

Schedule Departure Method

```
public static void ScheduleDeparture() {
    double ServiceTime;
// get the job at the head of the queue
    while (( ServiceTime = normal(stream, MeanServiceTime, SIGMA)) < 0 );
    Event depart = new Event(departure, Clock+ServiceTime);
    FutureEventList.enqueue( depart );
    NumberInService = 1;
    QueueLength--;
}
```

Process Departure Method

```
public static void ProcessDeparture(Event e) {
// get the customer description
    Event finished = (Event) Customers.dequeue();
// if there are customers in the queue then schedule the departure of the next one
    if( QueueLength > 0 ) ScheduleDeparture();
        else NumberInService = 0;
// measure the response time and add to the sum
    double response = (Clock - finished.get_time());
    SumResponseTime += response;
    if( response > 4.0 ) LongService++; // record long service
    TotalBusy += (Clock - LastEventTime );
    NumberOfDepartures++;
    LastEventTime = Clock;
}
```

Report Generator Method

```
public static void ReportGeneration() {
    double RHO = TotalBusy/Clock;
    double AVGR = SumResponseTime/TotalCustomers;
    double PC4 = ((double)LongService)/TotalCustomers;
    System.out.println( "SINGLE SERVER QUEUE SIMULATION - GROCERY
STORE CHECKOUT COUNTER ");
    System.out.println( "\t MEAN INTERARRIVAL TIME "+  MeanInterArrivalTime
);
    System.out.println( "\t MEAN SERVICE TIME “ + MeanServiceTime );
    System.out.println( "\t STANDARD DEVIATION OF SERVICE TIMES”+
SIGMA );
```

```
    System.out.println( "\t NUMBER OF CUSTOMERS SERVED“ + TotalCustomers );
    System.out.println( "\t SERVER UTILIZATION“ + RHO );
    System.out.println( "\t MAXIMUM LINE LENGTH" + MaxQueueLength );
    System.out.println( "\t AVERAGE RESPONSE TIME" + AVGR + " MINUTES" );
    System.out.println( "\t PROPORTION WHO SPEND FOUR ");
    System.out.println( "\t MINUTES OR MORE IN SYSTEM " + PC4 );
    System.out.println( "\t SIMULATION RUNLENGTH" + Clock + " MINUTES" );
    System.out.println( "\t NUMBER OF DEPARTURES" + TotalCustomers );
}
```

Methods to generate exponential and normal random variates

```
public static double exponential(Random rng, double mean) {
    return -mean*Math.log( rng.nextDouble() );
    }
     public static double SaveNormal;
     public static int NumNormals = 0;
     public static final double PI = 3.1415927 ;
     public static double normal(Random rng, double mean, double sigma) {
    double ReturnNormal; // should we generate two normals?
```

```
if(NumNormals == 0 ) {
        double r1 = rng.nextDouble();
        double r2 = rng.nextDouble();
        ReturnNormal = Math.sqrt(-*Math.log(r1))*Math.cos(2*PI*r2);
        SaveNormal = Math.sqrt(-2*Math.log(r1))*Math.sin(2*PI*r2);
        NumNormals = 1;
    } else {
        NumNormals = 0;
        ReturnNormal = SaveNormal;
    }
    return ReturnNormal*sigma + mean ;
}
```

Simulation In GPSS

- GPSS (General Purpose Simulation System) is a highly structured and special purpose simulation language based on process interaction approach and oriented toward queuing systems.
- The system being simulated is described by the block diagram using various GPSS blocks. Blocks represents events, delays and other actions that affect transaction flow
- Provides a convenient way to describe the system with over 40 standard blocks.
- GPSS model is developed by converting the block diagram into block statements and adding the control statements.
- The 1st version was released by IBM in 1961.
- GPSS/H is the most widely used version today.
  - Released in 1977
  - Flexible yet powerful.
  - The animator is Proof AnimationTM.

Transaction in GPSS

- A process that represents the real-world system we are modeling.
- Transaction is executed by moving from block to block.
- Each transaction in the model is contained in exactly one block, but one block may contain many transactions.

BLOCKS IN GPSS

GENERATE Block

- It generates the transaction.
- GENERATE A,B,C,D,E means:

A: Mean interval between generation of two transactions

B: Half width of uniform distribution or function modifier used to generate random interval between generation of transactions

C: Delay starting time

D: Limit on total transactions to be created

E: Priority level

- GENERATE 300 means interval times = 300
- If B does not specify a function, both A and B are evaluated numerically and a random number between A-B and A+B is used as the time increment.
- GENERATE 300,100 means interval times = [200,400]

ASSIGN Block

- Used to place or modify a value in a Transaction Parameter(Local Variable).
- If no such Parameter exists, it is created.

SAVEVALUE Block

- Changes the value of a **Savevalue Entity**.
- SAVEVALUE A,B   A: Savevalue Entity Number
  B: Value to be stored

TERMINATE Block

- Destroys the active transaction.
- TERMINATE means transaction ends
- TERMINATE 1 means simulation ends

ADVANCE Block

- Delays the progress of a transactions for a specified amount of simulated time.
- ADVANCE 100, 50

Using Facilities

- GPSS provides the facility modeling concept to represent limited availability of a service.
- **A facility is a resource that can be used by only one transaction at a time.**
- To request a facility transaction should enter in SEIZE block.
- Once a transaction has entered the SEIZE block, it owns the facility and other transactions are not allowed to enter this block.

- SEIZE A : gets the ownership of A
- RELEASE A : release the ownership of A

Using Storage

- GPSS provides the storage modeling concept to represent a limited number of unit capacity.
- A storage is a resource that can be used by several transactions at a time until it becomes empty.
- To get units from a storage, transaction should call an ENTER block.
- To put units to a storage, transaction should call a LEAVE block.

ENTER A,B

- Either takes or waits for a specified number of storage units.
- Where A: Storage Entity name or number

  B: Number of units by which to decrease the available storage capacity

LEAVE A,B

- Increases the accessible storage units at a storage entity.
- A: Storage Entity name or number.

  B: Number of units by which to increase the available storage capacity.

Collecting Time Statistics

- To collect data about how long it takes transactions to traverse a given segment of model.

QUEUE A,B

- Updates queue entity statistics to reflect an increase in content
- A: Queue Entity name or number
- B: Number of units by which to increase the content of the Queue Entity. Default value is 1.
- Example: QUEUE WaitingLine, 1
- The content of the Queue Entity named WaitingLine is increased by one and the associated statistics accumulators are updated

DEPART A,B

- Register statistics, which indicates a reduction in the content of a Queue Entity.
- A: Queue Entity name or number
- B: Number of units by which to decrease the content of the Queue Entity. Default value is 1.
- Example: DEPART WaitingLine, 1
- The content of the Queue Entity named WaitingLine is reduced by one and the associated statistics accumulators are updated.

Branching Operation

TRANSFER Block

- Causes the Active Transaction to jump to a new Block location.
- TRANSFER Block operates in “Unconditional Mode”.

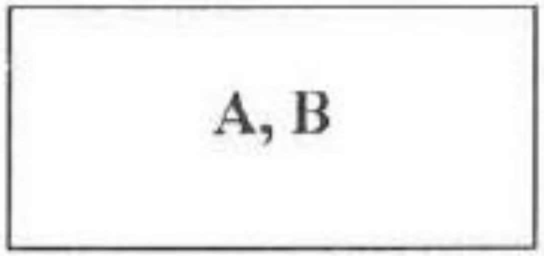
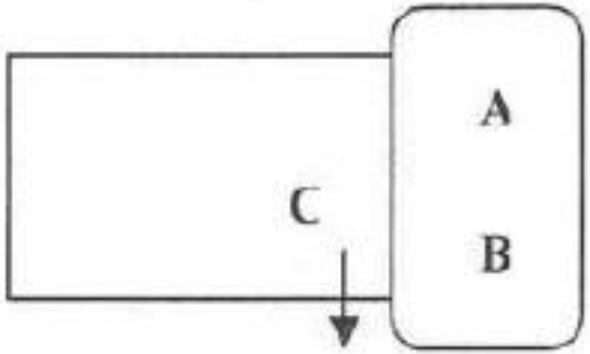
TEST Block

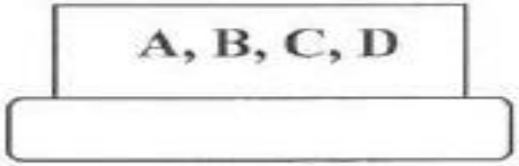
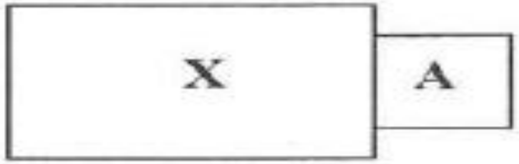
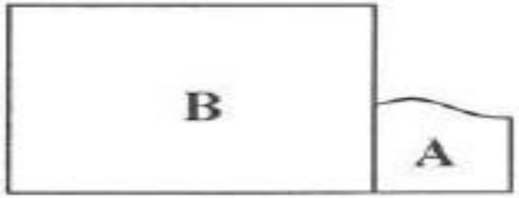
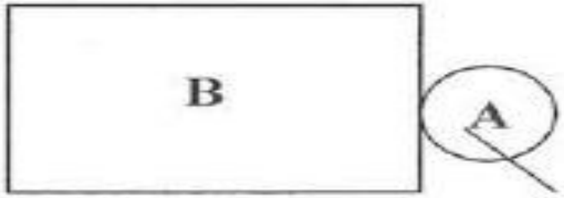
- To branch on some condition of the system.
- Transaction continues to next sequential program block if test is successful.
- Compares values, and controls the destination of the active transaction based on the result of the comparison.

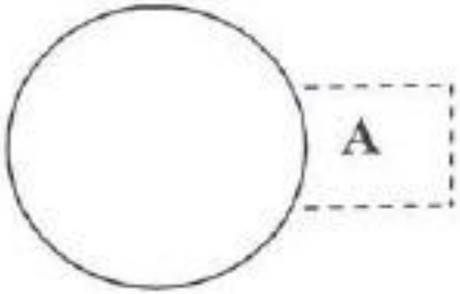
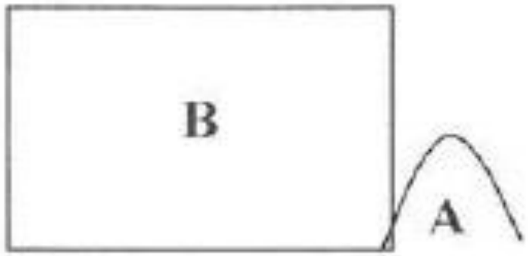
Sample1

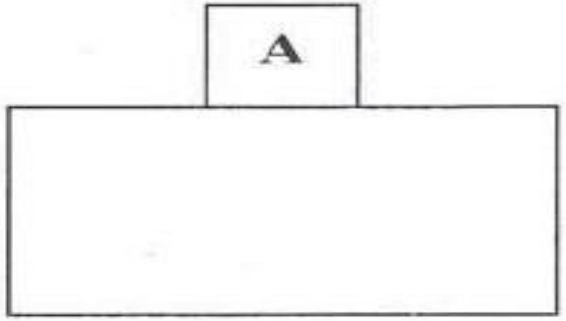
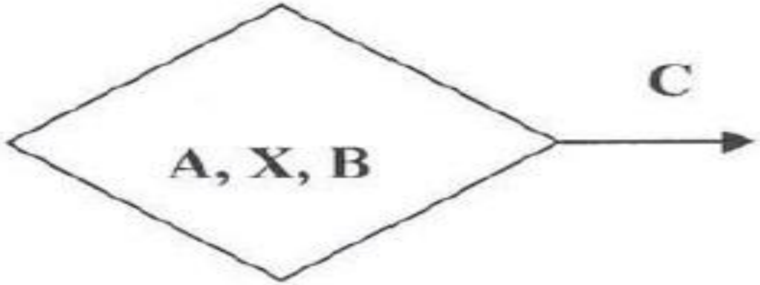
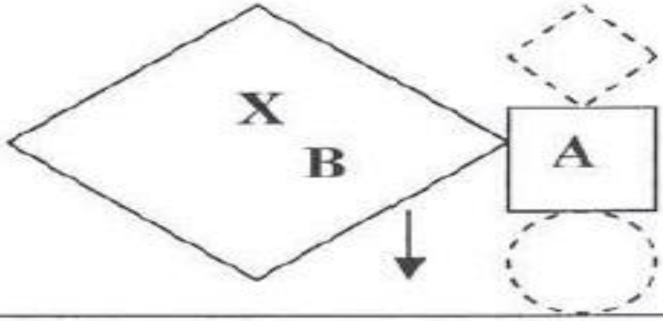
```
; GPSS World Sample File - SAMPLE1.GPS
***********************************************************************
*                                                                     *
*              Barber Shop Simulation                                 *
*                                                                     *
***********************************************************************
     GENERATE  300,100            ;Create next customer.
     QUEUE     Barber             ;Begin queue time.
     SEIZE     Barber             ;Own or wait for barber.
     DEPART    Barber             ;End queue time.
     ADVANCE   400,200            ;Haircut takes a few minutes.
     RELEASE   Barber             ;Haircut done. Give up the barber.
     TERMINATE 1                  ;Customer leaves.
```

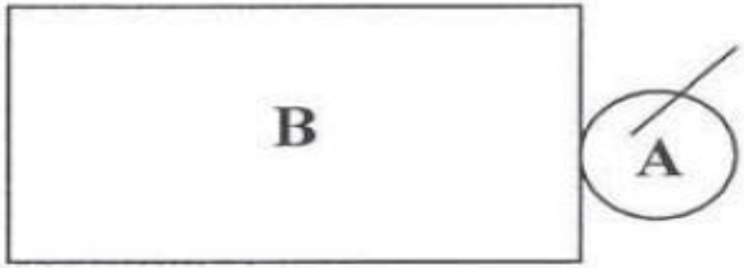
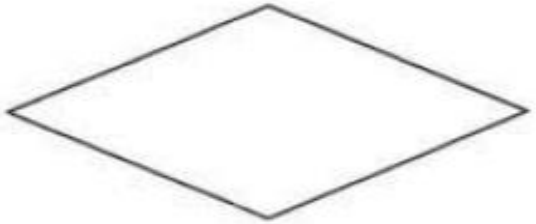
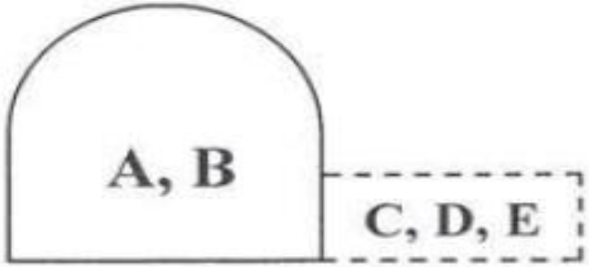
GPSS Block-Diagram Symbols

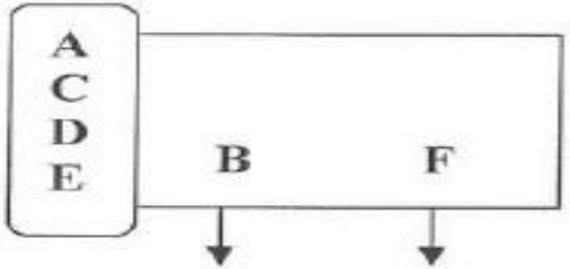
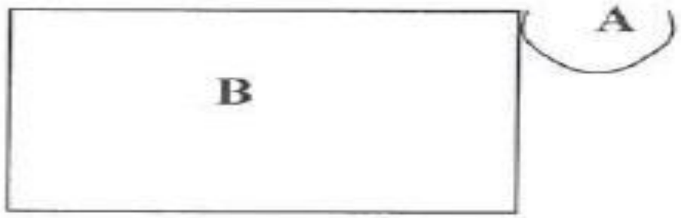
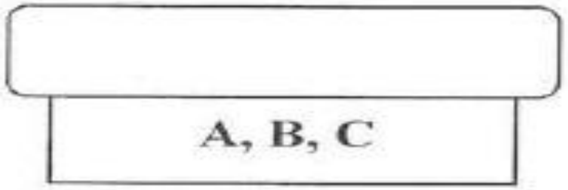
S.N.	Representation/Symbols	Meaning
1		Advance
2		Link

3		Seize
4		Assign
5		Logic
6		Tabulate
7		Depart

8		Mark
9		Terminate
10		Enter

11		Priority
12		Test
13		Gate

14		Queue
15		Transfer
16		Generate

<table>
<tr><td>17</td><td>
</td><td>Release</td></tr>
<tr><td>18</td><td>
</td><td>Unlink</td></tr>
<tr><td>19</td><td>
</td><td>Leave</td></tr>
<tr><td>20</td><td>
</td><td>Save Value</td></tr>
</table>

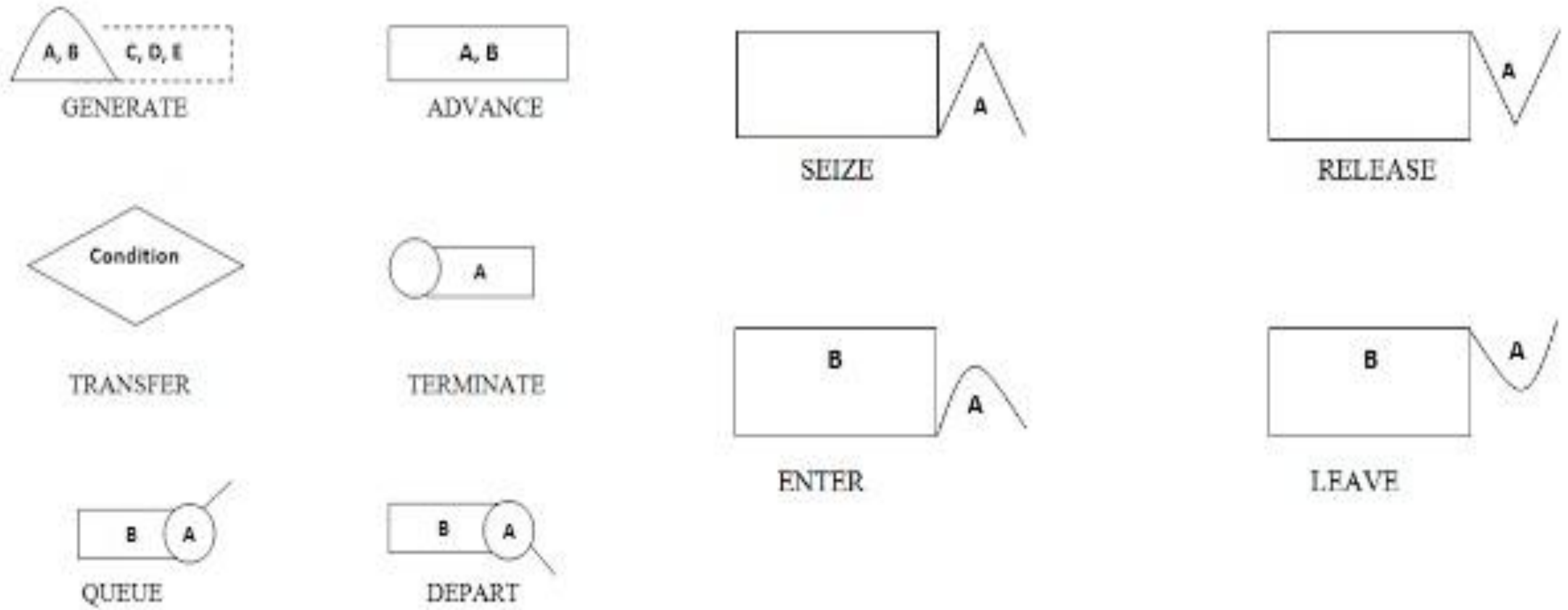
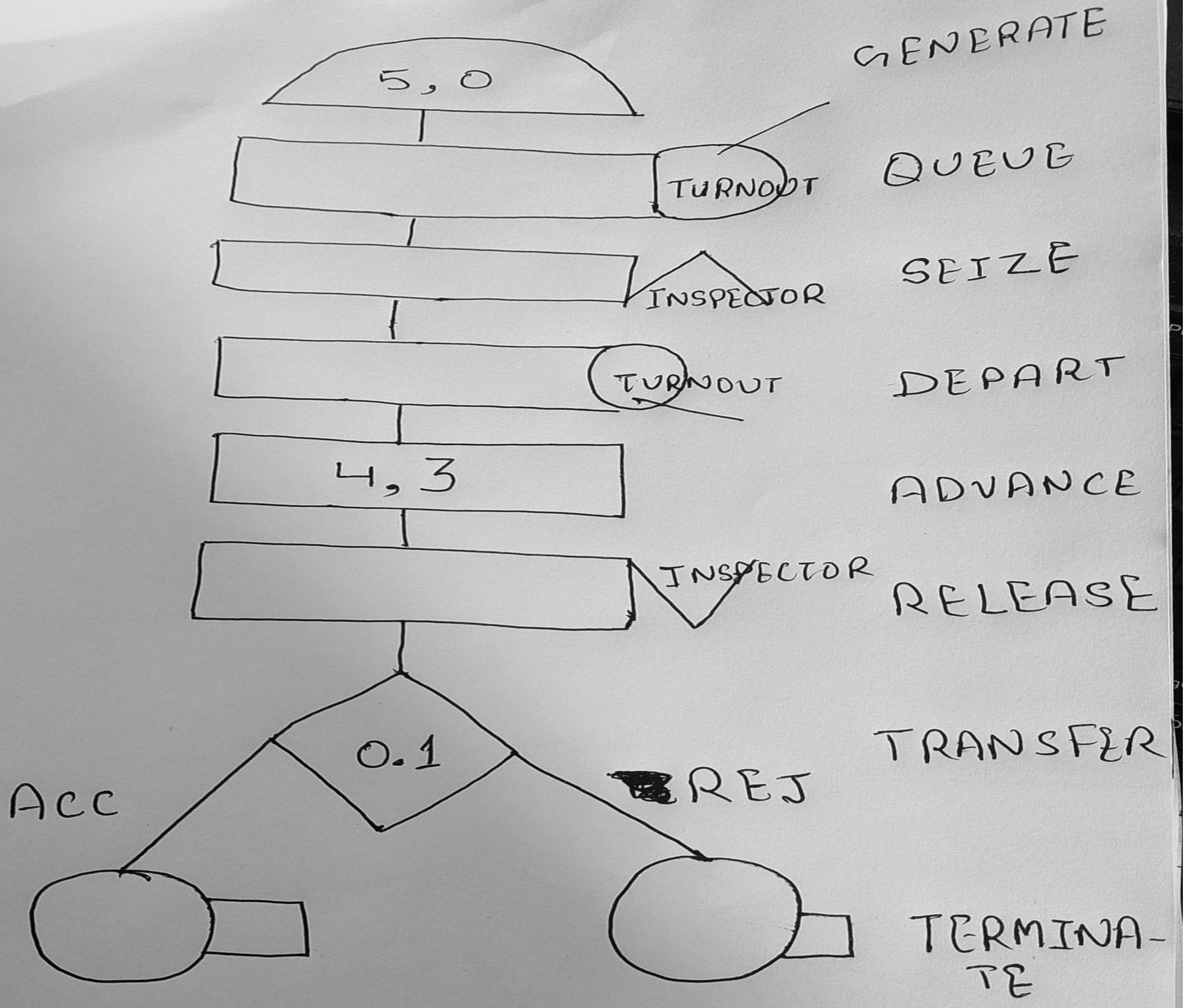


Figure: Mostly Used GPSS Models

Example 1 - Manufacturing Shop Model Simulation

A machine tool is turning out parse at a rate of 1 per every 5 minutes. As they are finished, the parse goes to an inspector, who takes 4 (+ or -) 3 minutes to examine each one and rejects about 10 % of the parse. Simulate the system using GPSS model.

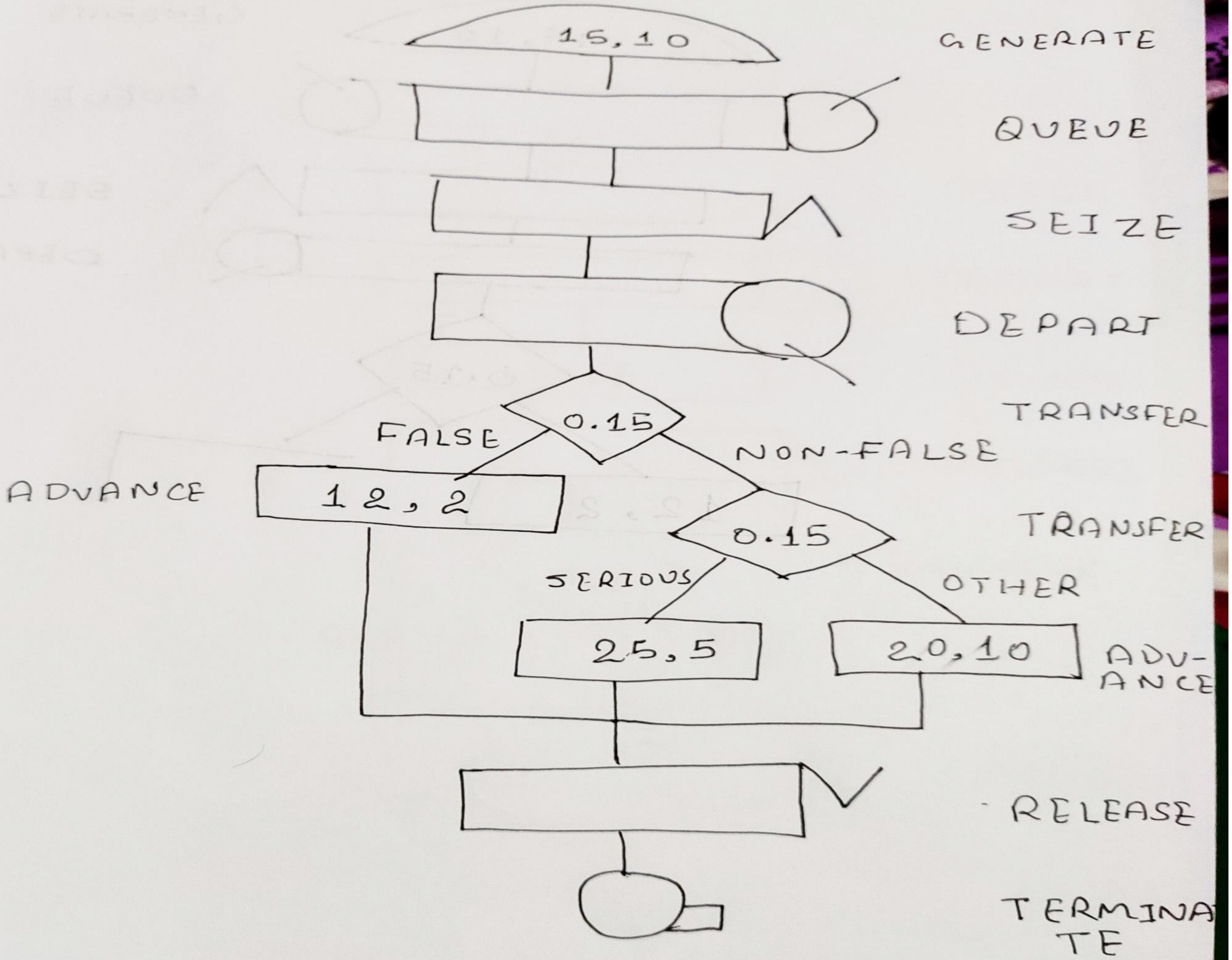

GENERATE
5, 0
QUEUE
TURNOUT
SEIZE
INSPECTOR
DEPART
TURNOUT
ADVANCE
4, 3
RELEASE
INSPECTOR
TRANSFER
0.1
ACC
REJ
TERMINA-
TE

```
GENERATE 5, 0
QUEUE TURNOUT
SEIZE INSPECTOR
DEPART TURNOUT

ADVANCE 4, 3
RELEASE INSPECTOR
TRANSFER 0.1 REJ ACC
TERMINATE
```

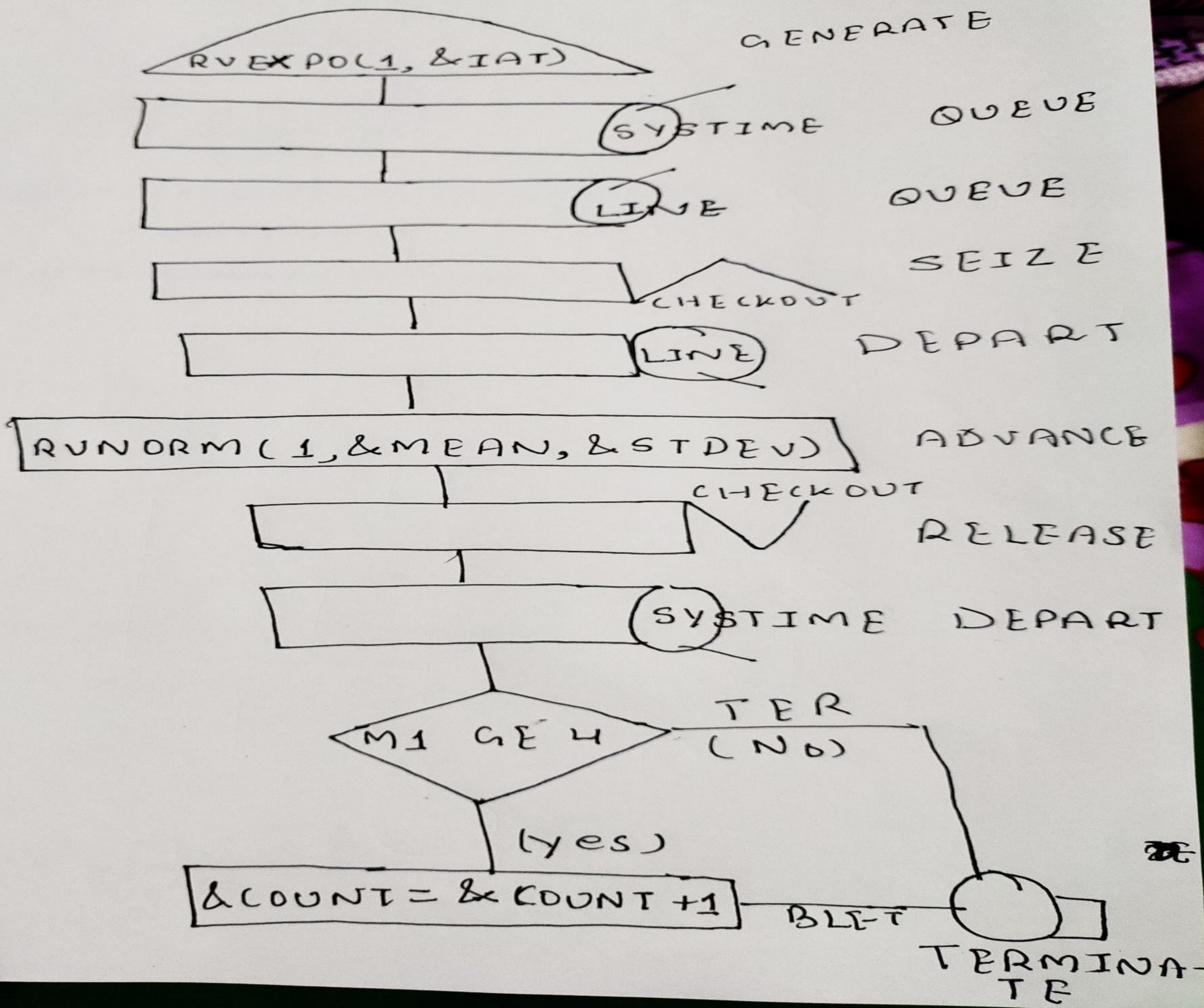
Example 2

Ambulances are dispatched at a rate of one every 15 (+ or -) 10 mins. Fifteen percent of the calls are false alarms, which require 12 (+ or -) 2 mins to complete. All other calls can be one of two kinds. The first kind are classified as serious. They constitute 15% of non false alarms and take 25 (+ or -) 5 mins to complete. The other calls take 20 (+ or -) 10 mins. Simulate the model using GPSS.


15, 10
GENERATE
QUEUE
SEIZE
DEPART
0.15
TRANSFER
FALSE
NON-FALSE
ADVANCE
12, 2
0.15
TRANSFER
SERIOUS
OTHER
25, 5
20, 10
ADV-
ANCE
RELEASE
TERMINA
TE

```
GENERATE 15, 10
QUEUE
SEIZE
DEPART
TRANSFER 0.15 FALSE NON-FALSE
FALSE ADVANCE 12, 2
NON-FALSE TRANSFER 0.15 SERIOUS OTHER
SERIOUS ADVANCE 25, 5
OTHER ADVANCE 20, 10
RELEASE
TERMINATE
```

```
*    GPSS/H Block Section
*
     GENERATE    RVEXPO(1,&IAT)             Exponential arrivals
     QUEUE       SYSTIME                    Begin response time data collection
     QUEUE        LINE                      Customer joins waiting line
     SEIZE       CHECKOUT                   Begin checkout at cash register
     DEPART      LINE                       Customer starting service leaves queue
     ADVANCE     RVNORM(1,&MEAN,&STDEV)      Customer's service time
     RELEASE     CHECKOUT                   Customer leaves checkout area
     DEPART      SYSTIME                    End response time data collection
     TEST GE     M1,4,TER                   Is response time GE 4 minutes?
     BLET        &COUNT=&COUNT+1            If so, add 1 to counter
     TER         TERMINATE                  1
*
     START    &LIMIT    Simulate for required number
```



RVEXPO(1, &IAT)
GENERATE
SYSTIME
QUEUE
LINE
QUEUE
SEIZE
CHECKOUT
DEPART
LINE
RVNORM(1, &MEAN, &STDEV)
ADVANCE
CHECKOUT
RELEASE
SYSTIME
DEPART
M1 GE 4
TER
(No)
(yes)
&COUNT = &COUNT +1
BLET
TERMINATE

Simulation In SSF

- SPF stands for **Scalable Simulation Framework.**
- SSF is an API(Application Program Interface).
- It describes a set of capabilities for **object-oriented**, **process-view simulation.**
- SSF provides a single, unified interface for discrete-event simulation.
- It is designed to achieve high performance.
- The API is sparse and allows implementations to achieve high performance, e.g. on parallel computers.
- Can be widely used in network simulation by using the add-on framework SSFNet.
- SSF bridges the gap between models developed in **pure Java** and models developed in languages **specifically designed for simulation.**
- It also provides the flexibility offered by a general-programming language, yet has essential support for simulation.

SSF Provides Five Base Classes

1. **Processes:** Processes implement threads of control (where the *action* method contains the execution body of the thread)
2. **Entity:** They describe the simulation objects.
3. **inChannel:** Communication endpoint
4. **outChannel:** Communication endpoint
5. **Events:** Events define the messages sent between entities.

SSF Model

1. Starting the Simulation :

- A simulation starts when any **entity's startAll() method is called**, commonly from the main() routine.
- The caller of startAll() specifies two timestamps: the simulation's start time, which defaults to 0, and the end time.

2. Initialization

- After **startAll()** has been called, but before any process has started executing, the framework calls the **init() routines of all processes and entities**.
- The Entity method shall return the simulation start time for the duration of initialization.

3. Process Execution

- After all processes have been initialized, they become eligible to run at the start time; **their actual execution is controlled by the framework**, which executes the process's action() callback method.

 Every time the action() method returns to the caller, the framework executes it again immediately, as many times are necessary to reach the end of the simulation.

4. Framework Inner Loop

 Within the framework, simulation time advances at a (possibly variable) rate determined by the arrival of events on channels, and the **duration of waitFor() statements.**

 When simulation time exceeds the end time specified in the original startAll() call, the simulation ends.

5. Start, Pause, Resume and Join

 The pauseAll() method allows the simulation to be paused gracefully. After pauseAll(), a call to resumeAll() resumes the simulation's forward progress.

 At any point, a call to joinAll() (e.g., from main()) will block without possibility of pause or resumption until simulation execution is complete.

 The startAll(), pauseAll(),resumeAll(), and joinAll() methods may not be called from within process code.

6. Framework Concurrency

- Each entity is said to be aligned to some object.
- Processes that are eligible to execute at each instant of simulation time will be scheduled by the framework: **sequentially within an alignment group** (but in implementation-dependent order), and **concurrently across alignment groups.**

7. Simulation Time

- The behavior of an SSF model is defined by the collective actions of its processes on its state.
- Every process resumption takes place at a particular simulation time, and each modification of model state by that resumption of the process takes place at that time.

Example : Single Server Queue System (Checkout Counter)

1. SSQueue Class : It is the class that contains the whole simulation experiment. It defines the experimental constants, contains SSF communication endpoints and defines inner class arrival.

2. Arrival process : It is a SSF process that stores identity of entity, creates a random number generator and enqueue the generated new arrivals, then blocks for inter arrival time.

3. Server process : It is the process that is called when a job has completed service or by a signal from the arrival process. It also updates statistics. Customers are dequeued from the waiting list or the process suspends if no customers were waiting.

Single Server Queue Example

SSQueue Class

```
class SSQueue extends Entity {

  private static Random rng;
  public static final double MeanServiceTime = 3.2;
  public static final double SIGMA = 0.6;
  public static final double MeanInterarrivalTime = 4.5;
  public static final long ticksPerUnitTime = 1000000000;
  public long generated=0;
  public Queue Waiting;
  outChannel out;
  inChannel in;

  public static long        TotalCustomers=0, MaxQueueLength=0,
                            TotalServiceTime=0;
  public static long        LongResponse=0, umResponseTime=0,
                            jobStart;




  }
```

Arrival class

```
class Arrivals extends process {
  private Random rng;
  private SSQueue owner;
  public Arrivals (SSQueue _owner, long seed) {
       super(_owner); owner = _owner;
       rng = new Random(seed);
  }
  public boolean isSimple() { return true; }
  public void action() {
     if ( generated++ > 0 ) {
       // put a new Customer on the queue with the present arrival time
       int Size = owner.Waiting.numElements();
       owner.Waiting.enqueue( new arrival(generated, now()));
       if( Size == 0) owner.out.write( new Event() );   // signal start of burst
     }
     waitFor(owner.d2t( owner.exponential(rng,
                owner.MeanInterarrivalTime)) );
 }
 }
}
```

Service Class

```
class Server extends process {
 private Random rng;
 private SSQueue owner ;
 private arrival in_service;
 private long service_time;

 public Server(SSQueue _owner, long seed) {
      super(_owner);
      owner = _owner;
      rng = new Random(seed);
 }
 public boolean isSimple() { return true; }
```

```
public void action() {
// if in_service is not null, we entered because of a job completion
    if( in_service != null ) {
        owner.TotalServiceTime += service_time;
        long in_system = (now() -in_service.arrival_time);
        owner.SumResponseTime  +=  in_system;
        if( owner.t2d(in_system) > 4.0 ) owner.LongResponse++;
        in_service = null;
        if( owner.MaxQueueLength < owner.Waiting.numElements() + 1 )
            owner.MaxQueueLength = owner.Waiting.numElements() + 1;
        owner.TotalCustomers++;
    }
    if( owner.Waiting.numElements() > 0 ) {
        in_service = (arrival)owner.Waiting.dequeue();
        service_time = -1;
        while ( service_time < 0.0 )
          service_time = owner.d2t(owner.normal( rng, owner.MeanServiceTime, owner.SIGMA));
    waitFor( service_time );
    } else {
        waitOn( owner.in );  // we await a wake-up call
    }
  }}
```

Other Simulation Software

1. Matlab

- Matlab is the easiest and most productive software environment for engineers and scientists.
- It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation.
- Typical uses include:
  - Math and computation
  - Algorithm development
  - Modeling, simulation, and prototyping
  - Data analysis, exploration, and visualization
  - Scientific and engineering graphics
  - Application development, including Graphical User Interface building

2. Arena

- Arena can be used for simulating discrete and continuous systems.
- At the heart of Arena is the **SIMAN** simulation language.
- Arena has a graphical user interface (GUI) built around the SIMAN language
- **Arena's Input Analyzer** automates the process of selecting the proper distribution and its inputs.
- The **Output Analyzer and Process Analyzer** automate comparison of different design alternatives.
- Arena is far more convenient than SIMAN, because it provides many handy features, such as high-level modules for model building, statistics definition and collection, animation of simulation runs (histories), and output report generation. Model building tends to be particularly intuitive, since many modules represent actual subsystems in the conceptual model or the real-life system under study.
- Complex models usually require both Arena modules and SIMAN blocks

The Arena Basic Edition:

- For modeling business processes and other systems in support of high-level analysis needs.

❖ The Arena Standard Edition:

- For modeling more detailed discrete and continuous systems.
- Models are built from graphical objects called modules to define system logic and physical components.
- Includes modules focused on specific aspects of manufacturing and material-handling systems.

❖ The Arena Professional Edition:

- With capability to craft custom simulation objects that mirror components system terminology logic data etc of real system, including terminology, process logic, data, etc.

AutoMod

- AutoMod is the 3-D simulation tool that can model the **largest and most complex manufacturing, distribution, and material handling systems** by combining the ease-of-use features of a simulation language.
- It provides detailed, large models used for planning, operational decision support, and control-system testing.
- It mainly focuses on manufacturing and material-handling systems.
- An AutoMod model consists of one or more systems:

  a. A system can be either a **process system** or a **movement system**.

  b. A model may contain any number of systems, which can be saved and reused as objects in other models.
- Optimization can be done which is based on an evolutionary strategies algorithm.

Assignment:

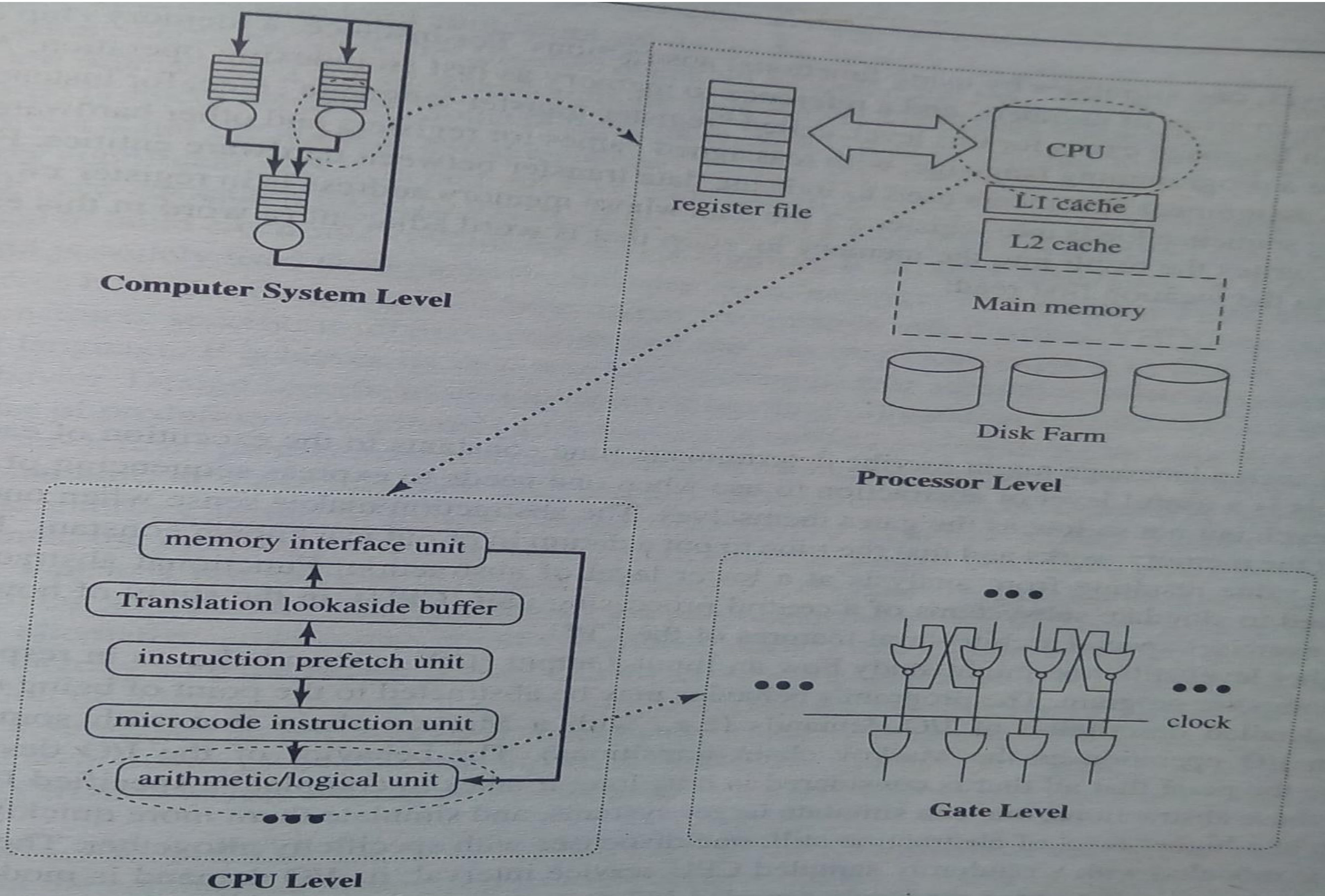
1. QUEST
2. Extend
3. Flexsim
4. Micro Saint
5. ProModel
6. Simul8
7. WITNESS

Chapter 10
Simulation of Computer System

Level of Abstraction in Computer System

- Level of Abstraction is defined as the amount of complexity by which a system is viewed or programmed.
- Computer system have complex time scale behaviour from **time to flipping transistor's state** to **time for human interaction**.
- It is designed hierarchically.
- The high level of abstraction is system level. In this level, one can view computational activity in terms of tasks circulating among servers, queuing for service when a server is busy.
- Below the system level is Processor level in which one can view components of the processor used.

- Below the Processor level is the CPU level in which one can view the activity of functional units that together make up a central processing unit.
- The lowest level is Gate level in which one can view the logical circuitry that is responsible for all the computations carried out by the computer system.
- Simulation is used in each level and the results of one level is used by another level.



Computer System Level
register file
CPU
L1 cache
L2 cache
Main memory
Disk Farm
Processor Level
memory interface unit
Translation lookaside buffer
instruction prefetch unit
microcode instruction unit
arithmetic/logical unit
CPU Level
clock
Gate Level

Simulation Tools

- Simulation tools are the tools that are used to perform and evaluate simulations at different abstraction levels of computer system.
- There are a number of powerful simulation tools available, all of them have advantages and disadvantages.
- An important characteristic of a tool is how it supports model building.
- The tools commonly used for simulation are:

1. CPU network simulation (Queueing network, Petri net simulators)
2. Processor simulation (VHDL(Very High Scale Integrated Circuit Hardware Description Language))
3. Memory simulation (VHDL)
4. ALU simulation (VHDL)
5. Logic network simulation (VHDL)
6. System Architecture Simulator(CSIM)

Activity, Process and Event Oriented Simulation

Activity Oriented Simulation

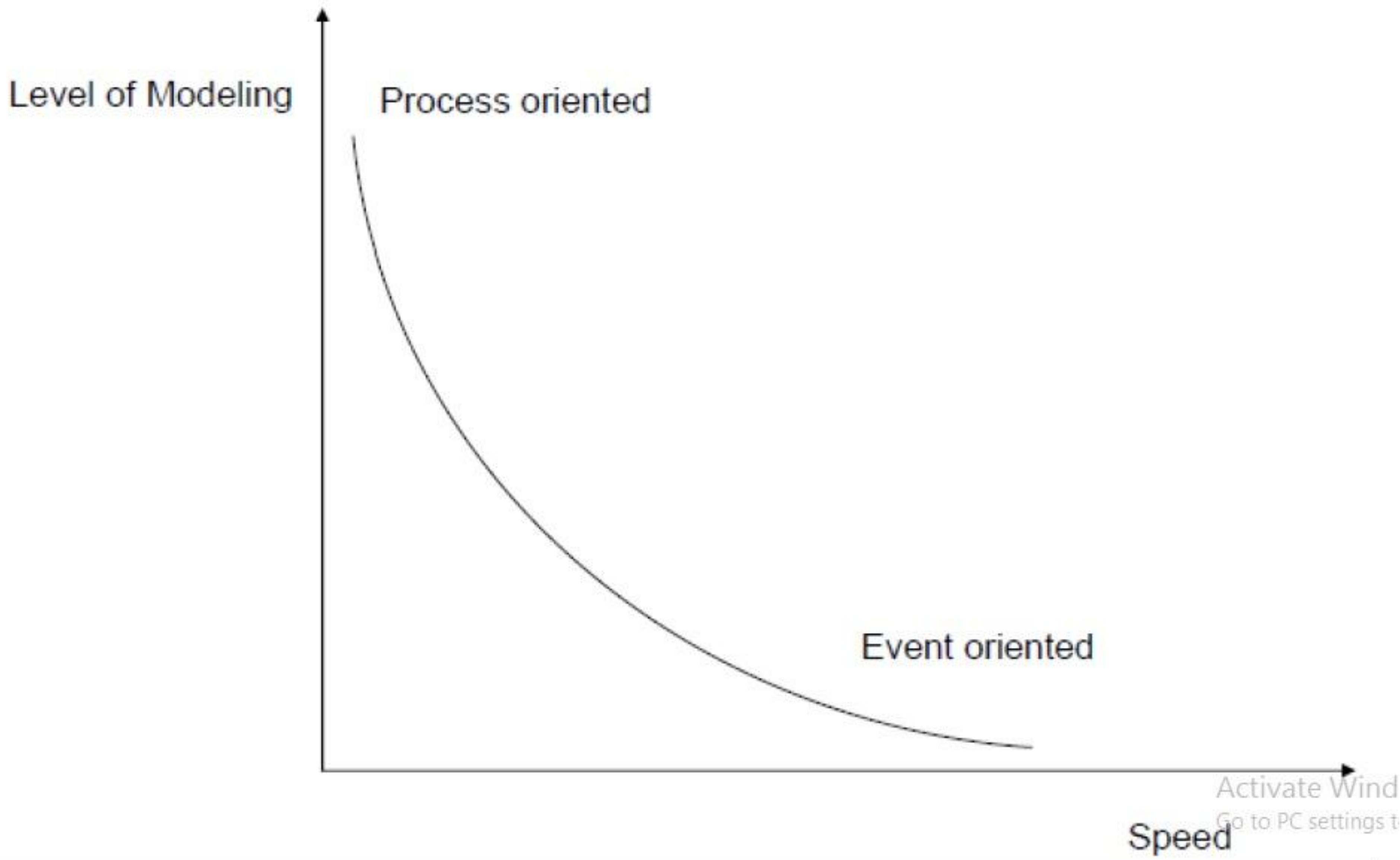
- The programmer defines the activities that are satisfied when certain conditions are satisfied.
- In many cases, this type of simulation uses a simulated clock which advance in constant increments of time.
- With each advance, list of activities is scanned and those which have become eligible are started.
- This type of model is used more often with simulating physical device.

Process Oriented Simulation

- The programmer defines the processes and the model in terms of interacting processes.
- A process is an independent program or procedure which can execute in parallel with other processes.
- The process will use the resource of the system.
- It implies that the tool must support separately schedulable threads of control.
- It allows continuous description with suspensions.

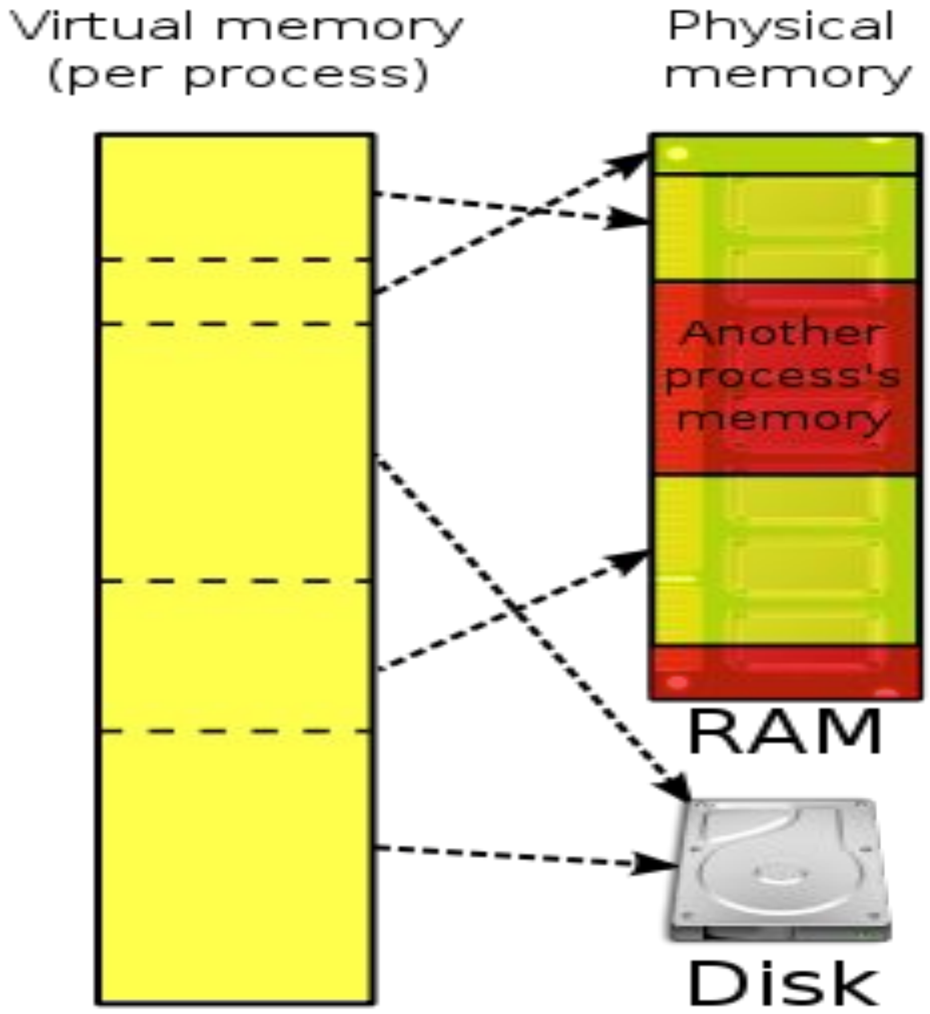
Event Oriented Simulation

- The simulation programmer defines events and then writes routines which are invoked as each kinds of events occur.
- It implies that the tool must support model description.
- It does not allows continuous description with suspensions.
- Usually a priority queue is used.


Level of Modeling
Process oriented
Event oriented
Speed

Virtual Memory Referencing

- Program is organized on units called pages.
- Physical memory is divided into page frames.
- Mapping is done by OS
- Replacement policy are used
- We can use computer simulation to find hit ratio(**ratio** of number of **hits** is divided by the total CPU reference of memory)

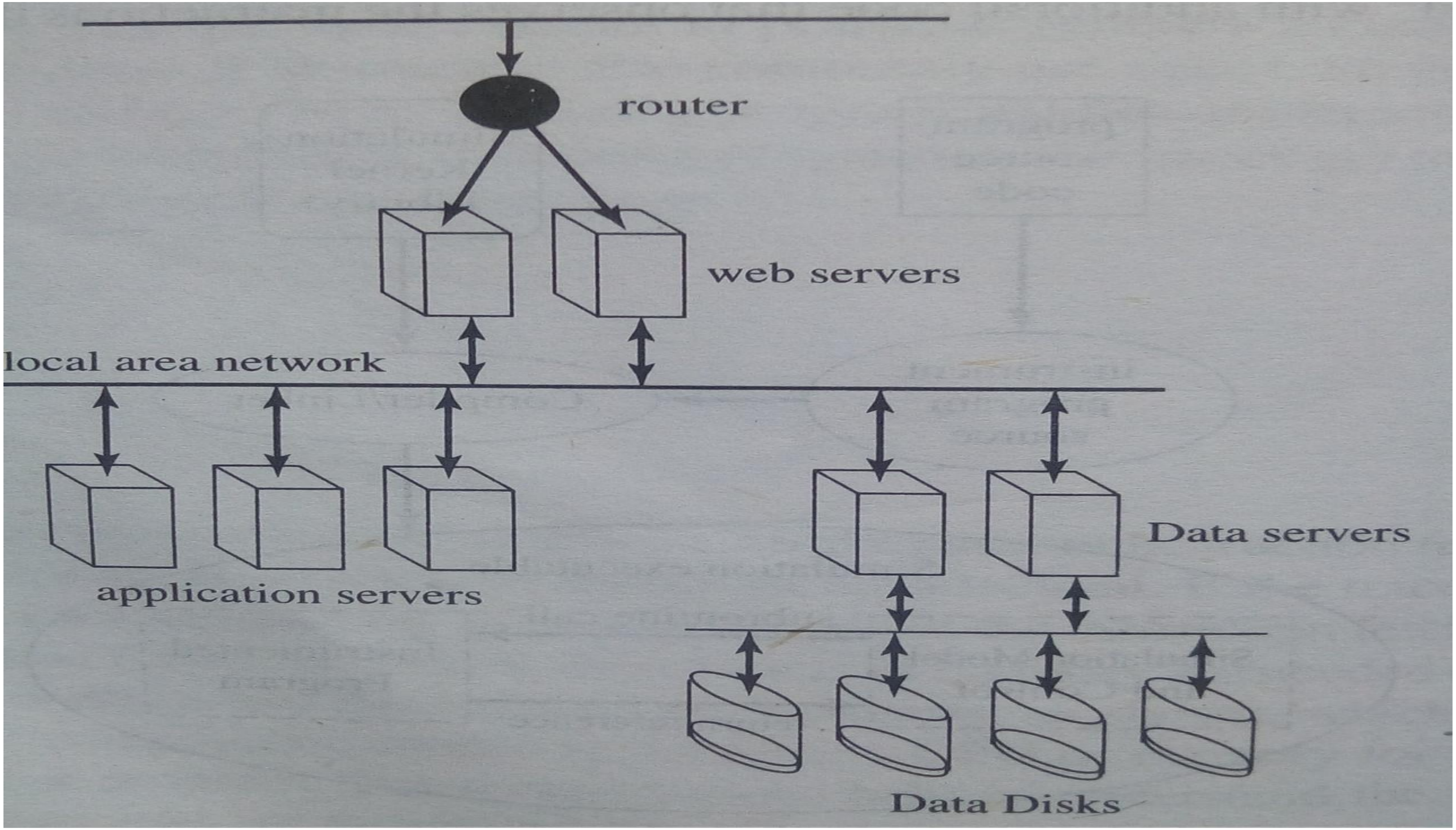


High Level Computer Simulation-System Simulation

Problem Definition

Consider a company provides a website for searching and links to sites for certain facilities. At the back end, there is **data servers** that handles specific queries and updates databases. Data servers receive requests for service from application servers. At front end, there is **web servers** that manage interaction of applications with the WWW. The whole system is connected with the users through the router. Let us consider that we need to study site's ability to handle load at peak periods i.e desired output is empirical distribution of the access response time.

Now, for this we need to focus on impact of timing at each level, factors that affect timing and effects of timing on contention for resources for designing this high level simulation model.



router
web servers
local area network
application servers
Data servers
Data Disks

Simulation Model

- All entries into the system are through dedicated router. It examines the request and forwards it to some web server.
- It takes some time to decide whether the request is a **new request or part of ongoing session.**
- One switching time is assumed for a pre-existing request and different time for a new request.
- It outputs the web server selection and enqueues request for service to the web server.
- Web server consists of one queue for **new requests**, one for **suspended requests that are waiting for response from application server** and **one for requests that are ready to process response from application server**.
- It is assumed that web server has enough memory to handle all the requests. It also has queuing policy.
- Associated application server is identified for each new requests.
- A request for service is formatted and forwarded to the application server and the request joins the suspended queue.

- Application server organizes the request for services. The new request for service joins the new-request queue.
- An application request is modeled as a sequence of sets of requests (organized in a burst) from data servers.
- For each application, a list of ready to execute and a list of suspended threads are maintained.
- Data servers create a new thread to respond to data request and places it in a queue of ready threads.
- When service is received, the thread requests data from a disk and then places in a suspended queue.
- Disk completes its operation for data request and the thread in data server, on receiving response from disk moves to ready list and reports back to application server associated with the request.
- The thread suspended at application server responds and finishes; then reports its completion to the web server.
- The thread in web server that initiates that request then communicates the results back to the Internet.

Note:

- Router have table of sessions
- Web server has three queues of threads
- Application server has two queues of threads
- Goal is to find response time distribution
- First we find bottleneck and then look how to reduce load at bottleneck during change of scheduling policy, biding applications to servers, increasing CPU and I/O devices.

Response Time

- Query-response-time distribution is estimated by measuring between the time at which a **request first hits the router** and the time at which **web server thread communicates the result.**
- The system can be analyzed by measuring behaviour at each server of each type.
- To assess system capacity at peak loads, we would simulate to identify bottlenecks, then look to see how to reduce load at bottleneck devices by changing various settings of simulation like scheduling policy, queue discipline and so on.

CPU Simulation

- In CPU simulation, we focus on discovering execution time and bottleneck situations that may appear.
- A bottleneck occurs when the **capacity of an application or a computer system is severely limited by a single component**, like the neck of a bottle slowing down the overall water flow.
- For CPU simulation, the input is the **stream of instructions** and the simulation must model the **logical design on what happens in response to the instruction stream.**
- Main challenges in CPU Simulation is to avoid stalling(Main challenges is to avoid stalling).

Problem Definition of ILP (Instruction Level Parallelism) CPU

 Pipelining has long been recognized as way of accelerating the execution of computer instructions.

The stages in an ILP CPU are as follows:

1. Instruction fetch - The instruction is fetched from memory.
2. Instruction decode - The memory word holding the instruction is interpreted to discover operations to be performed and registers involved.
3. Instruction Issue - An instruction is issued if no constraints hold it back from being executed.
4. Instruction Execute - The instruction operation is performed.
5. Instruction Complete - The results of instruction are stored in the destination register.
6. Instruction Graduate - Executed instructions are graduated in the order that they appear in the instruction stream.

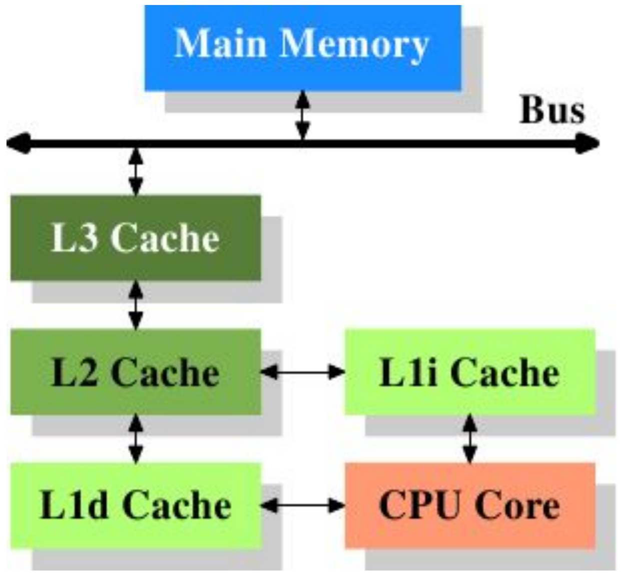
Simulation Model of ILP (Instruction Level Parallelism) CPU

- Instruction fetch interacts with the simulated memory system if present. If memory system is present, it can look into an instruction cache for the next referenced instruction, stalling if a miss is suffered. This stage makes instruction in the CPU's list of active instructions.
- Instruction Decode stage places an instruction in the list. A logical register that appears as the target of an operation is assigned a physical register. Registers used as operand are assigned physical registers that define their values. Branch instructions are identified and outcomes are predicted. Resources for the instruction execution are committed.
- Instruction Issue stage issue an decoded instruction for execution if values in its input registers are available and a functional unit needed to perform the instruction is available. It can be achieved by marking the registers and functional units as busy or pending. After the state is changed, the instruction waiting for that register or functional unit is reconsidered for issue.

- Instruction execute stage computes the result specified by the instruction. It means the actual operation intended by the instruction is performed.
- Instruction complete stage deposits the result into a register or memory as specified in the instruction.
- Instruction graduate reords the completed instruction in the same order as instruction stream. This is simulated by knowing the sequence number of the next instruction to be graduated.

Memory Simulation

- One of the great challenges of computer architecture is finding way to deal effectively with the increasing gap in operation speed between CPU and memory.
- Memory is arranged hierarchically with L1 cache, L2 cache, main memory and disks.
- Example: Cache Simulation
  - a. The input is cache parameters and memory access tree.
  - b. The output of simulation is cache hit rate or hit ratio.
- The **Cache Hit Ratio** is the **ratio** of the number of **cache hits** to the number of lookups(hit + miss), usually expressed as a percentage.



- Replacement Policy: Policy that determines which block in cache is removed in order to create space for coming block.
- Blocks can be removed in random fashion, using FIFO, LIFO,LFU(Least Frequently Used), LRU(Least Recently Used) strategies.
- LRU(Least Recently Used) is the widely used cache replacement strategy.

Simulation Model

- Maintain cache directory and LRU status of the lines within the set.
- When an access is made, update LRU status.
- If a hit, record it as such.
- If a miss, update the contents of the directory.
- Cache directory is implemented as an array, with array entries corresponding to directory entries.