WESLEY UNIVERSITY ONDO, NIGERIA

Faculty of Natural and Applied Sciences

Department of Computer Science

Course Code:	CCS 522
Course Title:	System Modelling and Simulation
Status:	Core
Semester:	Second
Mode of Delivery:	Classroom Lectures & Practical Sessions, online

Course Lecturer

N.A Udoh (MNCS)
Contact: +2347060553660
Email: nicholas.udoh@wesleyuni.edu.ng

Course Description

This course introduces students to the principles and techniques of system modelling and simulation. It focuses on the representation of real-world systems using mathematical, logical, and computational models, and the use of simulation to analyze system behavior over time. The course equips students with the skills required to design, implement, and evaluate simulation models for complex systems in science, engineering, and management.

Course Objectives

At the end of this course, students should be able to:

- Understand fundamental concepts of systems, models, and simulation.
- Identify and classify different types of systems and modelling approaches.
- Develop mathematical and logical models of real-world systems.
- Apply discrete-event and continuous simulation techniques.

.

Learning Outcomes

Upon successful completion of this course, students will be able to:

1. Explain key concepts and terminologies in system modelling and simulation.
2. Differentiate between deterministic and stochastic systems.
3. Construct models that represent real-life systems accurately.
4. Implement simulation models using appropriate tools and techniques.
5.

Assessment

Group Work Project (GWP)	15%
Collaborative Review (CR)	15%
Examination	70%
Total	100%

Module 1: MODELLING AND SIMULATION CONCEPTS

Module Introduction

This module is divided into six (6) units

Unit 1: Basics of Modelling and Simulation
Unit 2: Random Numbers
Unit 3: Random Number Generation
Unit 4: Monte Carlo Method
Unit 5: Statistical Distribution Functions
Unit 6: Common Probability Distributions

Unit 1: Basics of Modelling and Simulation

Contents

1.0 Introduction

The ability of man to define what may happen in the future and to choose among alternatives lie at the heart of contemporary societies. Our knowledge of the way things work, in society or nature are trailed with clouds of imprecision, and vast harms have followed a belief in certainty. To reduce the level of disparity between outcome and reality,

we require a decision analysis and support tool to enable us to evaluate, compare and optimize alternative. Such a tool should be able to provide explanations to various stakeholders and defend the decisions. One such tool that has been successfully employed is simulation which we use to vary the parameter of a model and observe the outcome.
Simulation has been particularly valuable:

a. When there is significant uncertainty regarding the outcome or consequences of a particular alternative under consideration. It allows you to deal with uncertainty and imprecision in a quantifiable way.
b. When the system under consideration involves complex interactions and requires input from multiple disciplines. In this case, it is difficult for any one person to easily understand the system. A simulation of the model can in such situations act as the framework to integrate the various components in order to better understand their interactions. As such, it becomes a management tool that keeps you focused on the "big picture" without getting lost in unimportant details.
c. when the consequences of a proposed action, plan or design cannot be directly and immediately observed (i.e., the consequences are delayed in time and/or dispersed in space) and/or it is simply impractical or prohibitively expensive to test the alternatives directly.

2.0 Intended Learning Outcomes (ILOs)

After studying this unit, you should be able to:

- Define a model and modelling.
- Explain when to and why we use models
- Describe the modelling process
- Describe different types of Models.

3.0 Main Content

Modelling and Simulation Concepts
Modern science would be inconceivable without computers to gather data and run model simulations. Whether it involves bringing back pictures of the surface of the planet Mars or detailed images to guide brain surgeons, computers have greatly extended our knowledge of the world around us and our ability to turn ideas into engineering reality. Thus modelling and computer simulation are important interdisciplinary tools.

3.1 Definitions

a. **Modelling** is the process of generating abstract, conceptual, graphical and/or mathematical models. Science offers a growing collection of methods, techniques and theory about all kinds of specialized scientific modelling.
Modelling also means to find relations between systems and models. Stated otherwise, models are abstractions of real or imaginary worlds we create to

understand their behaviour, play with them by performing "what if" experiments, make projections, animate or simply have fun.

b. A **model** in general is a pattern, plan, representation (especially in miniature), or description designed to show the main object or workings of an object, system, or concept.
c. A **model** (physical or hypothetical) is a representation of real-world phenomenon or elements (objects, concepts or events). Stated otherwise a model is an attempt to express a *possible structure of physical causality*.
   Models in science are often theoretical constructs that represent any particular thing with a set of variables and a set of logical and or quantitative relationships between them. Models in this sense are constructed to enable reasoning within an idealized logical framework about these processes and are an important component of scientific theories.
d. **Simulation -**is the manipulation of a model in such a way that it operates on time or space to compress it, thus enabling one to perceive the interactions that would not otherwise be apparent because of their separation in time or space.
e. Modelling and Simulation is a discipline for developing a level of understanding of the interaction of the parts of a system, and of the system as a whole. The level of understanding which may be developed via this discipline is seldom achievable via any other discipline.
f. A **computer model** is a simulation or model of a situation in the real world or an imaginary world which has parameters that the user can alter.

For example Newton considers movement (of planets and of masses) and writes equations, among which $f = ma$ (where f is force, m mass and a acceleration), that make the dynamics intelligible. Newton by this expression makes a formidable proposition, that force causes acceleration, with mass as proportionality coefficient. Another example, a model airplane is a physical representation of the real airplane; model of airplanes are useful in predicting the behaviour of the real airplane when subjected to different conditions; weather, speed, load, etc. Models help us frame our thinking about objects in the real world. It should be noted that more often than not we model dynamic (changing) systems.

3.2 What is Modelling and Simulation?

Modelling is a discipline for developing a level of understanding of the interaction of the parts of a system, and of the system as a whole. The level of understanding which may be developed via this discipline is seldom achievable via any other discipline.

A simulation is a technique (not a method) for representing a dynamic real world system by a model and experimenting with the model in order to gain information about the system and therefore take appropriate decision. Simulation can be done by hand or by a computer. Simulations are generally iterative in their development. One develops a model, simulates it, learns from the result, revises the model, and continues the iterations until an adequate level of understanding is attained.

Modelling and Simulation is a discipline, it is also very much an art form. One can learn

about riding a bicycle from reading a book. To really learn to ride a bicycle one must become actively engaged with a bicycle. Modelling and Simulation follows much the same reality. You can learn much about modelling and simulation from reading books and talking with other people. Skill and talent in developing models and performing simulations is only developed through the building of models and simulating them. It is very much ―learn as you go‖ process. From the interaction of the developer and the models emerges an understanding of what makes sense and what doesn't.

3.3 Type of Models

There are many types of models and different ways of classifying/grouping them. For simplicity, Models may be grouped into the following – Physical, Mathematical, Analogue, Simulation, Heuristic, Stochastic and Deterministic models.

a. Physical Models

These are call iconic models. Good examples of physical models are model cars, model railway, model airplane, scale models, etc. A model railway can be used to study the behaviour of a real railway, also scale models can be used to study a plant layout design. In simulation studies, iconic models are rarely used.

b. Mathematical Models

These are models used for predictive (projecting) purposes. They are abstract and take the form of mathematical expressions of relationships. For examples:

1. $x^2 + y^2 = 1$ (mathematical model of a circle of radius 1)

2. $$\text{Interest} = \frac{\text{Principal} \times \text{Rate} \times \text{Time}}{100}$$

3. Linear programming models and so on.

Mathematical models can be as simple as interest earnings on a savings account or as complex as the operation of an entire factory or landing astronauts on the moon. The development of mathematical models requires great deal of skill and knowledge.

c. Analogue Models

These are similar to iconic models. But here some other entities are used to represent directly the entities of the real world. An example is the analogue computer where the magnitudes of the electrical currents flowing in a circuit can be used to represent quantities of materials or people moving around in a system. Other examples are; the gauge used to check the pressure in a tyre. The movement of the dial represent the air pressure in the tyre. In medical examination, the marks of electrical current on paper, is the analogue representation of the working of muscles or organs.

d. Simulation Models

Here, instead of entities being represented physically, they are represented by sequences of random numbers subject to the assumptions of the model. These models represent (emulate) the behaviour of a real system. They are used where there are no suitable mathematical models or where the mathematical model is too complex or where it is not possible to experiment upon a working system without causing serious disruption.

e. Heuristic Models

These models use intuitive (or futuristic) rules with the hope that it will produce workable solutions, which can be improved upon. For example, the Arthur C Clerk‘s heuristic model was the forerunner of the communications satellite and today‘s international television broadcast.

f. Deterministic Models

These are models that contain certain known and fixed constants throughout their formulation e.g., Economic Order Quantity (EOQ) for inventory control under uncertainty.

g. Stochastic models

These are models that involve one or more uncertain variables and as such are subject to probabilities.

3.4 Advantages of Using Models

- They are safer.
- They are less expensive. For example, Practical Simulators are used to train pilots.
- They are easier to control than the real world counterparts.

3.5 Applications

One application of scientific modelling is the field of "Modelling and Simulation", generally referred to as "M&S". M&S has a spectrum of applications which range from concept development and analysis, through experimentation, measurement and verification, to disposal analysis. Projects and programs may use hundreds of different simulations, simulators and model analysis tools

3.6 Modelling Procedure

In modelling we construct a suitable representation of an identified real world problem, obtain solution(s) for that representation and interpret each solution in terms of the real situation. The steps involved in modelling are as follows:

1. Examine the real world situation.
2. Extract the essential features from the real world situation.
3. Construct a model of the real (object or system) using just the essential features identified.
4. Solve and experiment with the model.
5. Draw conclusions about the model.
6. If a further refinement necessary, then re-examine the model and readjust parameters

and continue at 4, otherwise continue at 7.

7. Proceed with implementation.

Explanation of the Steps

Begin with the real world situation, which is to be investigated with a view to solving some problem or improving that situation.

The first important step is to extract from the real world situation the essential features to be included in the model. Include only factors that make the model a meaningful representation of reality, while not creating a model, which is difficult by including many variables that do not have much effect. Factors to be considered include ones over which management has control and external factors beyond management control. For the factors included, assumptions have to be made about their behaviour.

Run (simulate) the model and measure what happens. For example, if we have simulation of a queuing situation where two servers are employed, we can run this for hundreds of customers passing through the system and obtain results such as the average length of the queue and the average waiting time per customer. We can then run it with three servers, say, and see what new values are obtained for these parameters. Many such runs can be carried out making different changes to the structure and assumptions of the model.

In the case of a mathematical model we have to solve a set of equations of some sort, e.g. linear programming problem where we have to solve a set of constraints as simultaneous equations, or in stock control – where we have to use previously accumulated data to predict the future value of a particular variable.

When we have solved our mathematical model or evaluated some simulation runs, we can now draw some conclusions about the model. For example, if we have the average queue length and the average waiting time for a queuing situation varied in some ways, we can use this in conjunction with information on such matters as the wage-rates for servers and value of time lost in the queue to arrive at decisions on the best way to service the queue.

Finally, we use our conclusions about the model to draw some conclusions about the original real world situation. The validity of the conclusions will depend on how well our model actually represented the real world situation.

Usually the first attempt at modelling the situation will almost certainly lead to results at variance with reality. We have to look back at the assumptions in the model and adjust them. The model must be rebuilt and new results obtained. Usually, a large number of iterations of this form will be required before acceptable model is obtained. When an acceptable model has been obtained, it is necessary to test the sensitivity of that model to possible changes in condition.

The modelling process can then be considered for implementation when it is decided that the

model is presenting the real world (object or system) sufficiently well for conclusions drawn from it to be a useful guide to action.

The model can be solved by hand, especially if it is simple. It could take time to arrive at an acceptable model. For complex models or models which involve tremendous amount of data, the computer is very useful.

4.0 Self-Assessment Exercise(s)

Answer the following questions:

1. Differentiate between Model, Modelling, Simulation and Computer model.
2. What are the steps followed in modelling?
3. State why we use models

5.0 Conclusion

In this unit we took a look at an overview of major concepts that underlie models to prepare us for the work in this course simulation and modelling.

6.0 Summary

In introducing this unit, it was stated that simulation is a decision support tool which enable us to evaluate, compare and optimize alternative ways of solving a problem and the following were discussed:

- Modelling was defined
- The concepts of modelling were outlined
- Why we use models
- The application of models especially for simulations
- The types of modelswhich include: Physical, Mathematical, Analogue, Simulation, Heuristic, Stochastic and Deterministic models were highlighted.

7.0 Further Readings

- Devore, J. L. (2018). *Probability and statistics for engineering and the sciences*. Toronto, Ontario: Nelson.
- Georgii, H. (2013). *Stochastics: Introduction to probability and statistics*. Berlin: De Gruyter.
- Giri, N. C. (2019). *Introduction to probability and statistics*. London: Routledge.
- Johnson, R. A., Miller, I., & Freund, J. E. (2019). *Miller & Freunds probability and statistics for engineers*. Boston: Pearson Education.
- Laha, R. G., & Rohatgi, V. K. (2020). *Probability theory*. Mineola, NY: Dover Publications.

- Mathai, A. M., & Haubold, H. J. (2018). *Probability and statistics: A course for physicists and engineers*. Boston: De Gruyter.
- Pishro-Nik, H. (2014). *Introduction to probability, statistics, and random processes*. Blue Bell, PA: Kappa Research, LLC.
- Spiegel, M. R., Schiller, J. J., & Srinivasan, R. A. (2013). *Schaums outline of probability and statistics*. New York: McGraw-Hill.

Unit 2: Random Numbers

Contents

1.0 Introduction

The use of Random numbers lies at the foundation of modelling and simulations. Computer applications such as simulations, games, graphics, etc., often need the ability to generate random numbers for such application.

The quality of a random number generator is proportional to its **period**, or the number of random numbers it can produce before a repeating pattern sets in. In large-scale simulations, different algorithms (called shift-register and lagged-Fibonacci) can be used, although these also have some drawbacks, combining two different types of generators may produce the best results.

2.0 Intended Learning Outcomes (ILOs)

By the end this unit, you should be able to:

- Describe how to generate pseudorandom numbers,
- Use QBasic RND function and describe how to simulate randomness,
- Use different Random number generators,
- Explain properties of good random number generator.

3.0 Main Content

Random Number can be defined as numbers that show no consistent pattern, with each number in a series and are neither affected in any way by the preceding number, nor predictable from it.

One way to get random digits is to simply start with an arbitrary number with a specified number of digits, for example 4 digits. The first number is called the **seed**. The seed is multiplied by a **constant** number of the same number of digits(length), and the desired number of digits is taken off the right end of the product. The result becomes the new seed. It is again multiplied by the original constant to generate a new product, and the process is repeated as often as desired. The result is a series of digits that appear randomly distributed as though generated by throwing a die or spinning a wheel. This type of algorithm is called a **congruential generator**.

Generating a random number series from a single seed works fine with most simulations that rely upon generating random events under the control of probabilities (Monte Carlo simulations). However, although the sequence of numbers generated from a given seed is randomly distributed, it is always the same series of numbers for the same seed. Thus, a computer poker game that simply used a given seed would always generate the same hands for each player.

What is needed is a large collection of potential seeds from which one can be more or less randomly chosen. If there are enough possible seeds, the odds of ever getting the same series of numbers become diminishingly small.

One way to do this is to read the time (and perhaps date) from the computer‘s system clock and generate a seed based on that value. Since the clock value is in milliseconds, there are millions of possible values to choose from. Another common technique is to use the interval between the user‘s keystrokes (in milliseconds). Although they are not perfect, these techniques are quite adequate for games.

The so-called true random number generators extract random numbers from physical phenomena such as a radioactive source or even atmospheric noise as detected by a radio receiver.

3.1 Pseudorandom Number Generation

In this section we look at how random numbers may be generated by human beings for use in simulating a system or by computer for use while simulating an event.

What we usually do is to take for instance ten pieces of papers and number them 0,1,2,3,4,5,6,7,8, and 9 , fold and place them in a box. Shake the box and thoroughly mix the slips of paper. Select a slip; then record the number that is on it. Replace the slip and repeat this procedure over and over. The resultant record of digits is a realized sequence of random numbers. Assuming you thoroughly mix the slips before every draw, the nth digit of the sequence has an equal or uniform chance of being any of the digits 0, 1, 2,3,4,5,6,7,8, 9 irrespective of all the preceding digits in the recorded sequence.

In some simulations, we use random numbers that are between 0 and 1. For example, if you need such numbers with four decimal digits, then you can take four at a time from the

recorded sequence of random digits, and place a decimal point in front of each group of four. To illustrate, if the sequence of digits is 358083429261… then the four decimal placed random numbers are .3580, .8342, and .9261.

3.2 Random Numbers in Computer

How does computer generate a sequence of random numbers?

One way is to perform the above ―slip-in-a-box‖ experiment and then store the recorded sequence in a computer-backing store.

The RAND Corporation using specially designed electronic equipment, to perform the experiment, actually did generate a table of a million random digits. The table can be obtained on tape, so that blocks of the numbers can be read into the memory of a high- speed computer, as they are needed. Their approach is disadvantageous since considerable computer time was expended in the delays of reading numbers into memory from a tape drive.

Experts in computer science have devised mathematical processes for generating digits that yield sequences satisfying many of the statistical properties of a truly random process. To illustrate, if you examine a long sequence of digits produced by deterministic formulas, each digit will occur with nearly the same frequency, odd numbers will be followed by even numbers about as often as by odd numbers, different pairs of numbers occur with nearly the same frequency, etc. Since such a process is not really random, it is called **pseudo-random number generator**.

The other ways of generating pseudo-random numbers are:

1. Computer simulation languages and indeed some programming languages such as BASIC have built-in pseudo-random number generators. In computer simulation situations where this facility is not available in the language you are using, you will have to write you own pseudo-random number generator (see how to do this later).
2. The results of experiments such as the one previously describe above are published in books of statistical tables. In hand simulation, it may be appropriate to use a published table of random numbers.
3. The conventional six-sided unbiased die may also be used to generate a sequence of random digits in the set (1, 2, 3, 4, 5, 6) where each digit has a probability 1/6 of occurrence.

Exercise

Suggest one or two experimental set-ups (analogous to the slip-in-a-box approach) for generating uniform random digits.

3.3 Using the RND Function in BASIC

The BASIC programming language has a numeric function named RND, which generates random numbers between 0 and 1. Each time RND is executed, a pseudo random number between 0 and 1 is generated. Using RND function at any time will always generate the

same sequence of pseudo random numbers unless we vary the random number seed using the BASIC statement:

RANDOMIZE

This way, we can control the sequence of random numbers generated. RANDOMIZE will result to the following prompt on the VDU:

Random Number Seed (-32768 to 32767)?

Suppose your response to the above prompt is 100. Then the computer would use this number, 100, to generate the first random number. This number generated is used to generate the next random number. Thus by specifying the seed for the first random number, we are in a way controlling all random numbers that will be generated until the seed is reset. A control such as this can be very useful in validating a simulation program or other computer programs that use random numbers.
Consider the following BASIC program:

```
FOR K% = 1 TO 5
PRINT RND NEXT K%
END
```

If the above program is run, some seven-digit decimal numbers like the following will be displayed: .6291626, .1948297, .6305799, .8625749, .736353. The particular digits displayed depend on the system time.

Every time you run the above program, different sequence of numbers will be displayed. Now add a RANDOMIZE statement to the program:

```
RANDOMIZE TIMER FOR K% = 1 TO 5
        PRINT RND NEXT K%
END
```

If you run this program with 300 as a response to the prompt for the random number seed, the following may be displayed: .1851404, .9877729, .806621, .8573399, .6208935

Exercise
Find out whether the same set of random numbers will be displayed each time the above program is run with seed 300.

3.4 Simulating Randomness

Suppose we want to simulate the throwing of a fair die. A random number between 0 and 1 will not always satisfy our needs. If the die is fair, throwing it several times will yield a series of uniformly distributed integers 1,2,3,4,5 and 6. Consequently we need to be able to generate a random integer with values in the range 1 and 6 inclusive.

Now the function RND generates a random number between 0 and 1. Specifically, the random variable X is in the range: $0 \leq X < 1$
The expression X = RND *6
Will generate a number in the range: $0 \leq X < 6$
We must convert these numbers to integers as follows: X% = INT (RND*6)
The expression produces an integer in the range: $0 \leq X < 5$

But we need the range: $0 \leq X < 6$

Therefore if we need to add 1 to the above expression in simulating the tossing of a die. Thus,

X% = INT (RND*6) + 1
In general, to generate a random integer between P and N we use the expression:
INT(RND*N+1-P) + P;
where N>P

While for integer number between 0 and N – 1 we use the expression INT (RND *N).

Example 1
A simple QBASIC program that will stimulate the tossing of two dice and display the value obtained after each toss, and the total value of the dice is shown below.

```
CLS
REM DI and D2 represent the individual dice. RANDOMIZE
DO
        D1% = INT(RND*6) + 1
        D2% = INT(RND*6) + 1
        TOTAL% = D1% + D2%
        PRINT —Die 1:‖; D1%; —Die 2:‖; D2%
        PRINT: PRINT
        INPUT —Toss Again (Y/N)?‖, Y$
        LOOP UNTIL UCASE$(Y$) = —N‖

END
```

Exercise
Run the program of example 1several times using different random number seeds to

determine if the integers generated for the individual die are uniformly distributed between 1 and 6 inclusive.

If we want the computer to be generating the random number seed automatically, we use
RANDOMIZE TIMER

In place of RANDOMIZE.

Example 2
Another QBASIC program to simulate the tossing of a fair coin 10 times. The program displays a H when a head appears and a T when a tail appears.

```
CLS
REM Program to simulate the tossing of a coin 10 times
REM and print the outcome
RANDOMIZE TIMER
FOR K% = 1 TO 10
RANDNO = RND
IF RANDNO <= 0.5 PRINT ―H‖
IF RANDNO > 0.5
PRINT ―T‖
NEXT  K%
END
```

Example 3
Suppose the output of the program of example 3 is: HHTHHTTTHH
and that there are two players X and Y involved in the tossing of the coin. Given that player X wins N50.00 from player Y if a head appears and loses it to player Y if a tail appears. Determine who won the game and by how much.

Solution
From the output there are 6 heads and 4 tails.
Player X wins N50.00 x 6 = N300.00 from player Y. He loses N50.00 x 4 = N200.00 to player Y.
Thus, player X won the game with N300.00 – N200.00 = N100.00.

3.5 Properties of a Good Random Number Generator

The random numbers generated should;

a. have as nearly as possible a uniform distribution.
b. should be fast
c. not require large amounts of memory.
d. have a long period.
e. be able to generate a different set of random numbers or a series of numbers.
f. not degenerate.

4.0 Self-Assessment Exercise(s)

Write a QBASIC program to generate thirty random integer numbers distributed between 20 and 50. Your program should ensure that no number is repeated.

Write a QBASIC program to accept a set of characters from the keyboard and then move the characters randomly across the screen. The movement of the characters should stop once a key is pressed on the keyboard. The set of characters should also change colors randomly at the point of the movement.

What is a seed and explain how you can generate random numbers using a seed.

Define a period and state how to improve a period.

5.0 Conclusion

In this unit, you have been introduced to Random Numbers generation. You have also learnt the how to manipulate the RND function of QBASIC.

6.0 Summary

What you have learnt in this unit concern:

- The different ways of generating pseudorandom numbers,
- The properties of good random number generator.
- The use of QBasic RND function to simulate randomness,
- The other Random number generating methods,

7.0 Further Readings

- Devore, J. L. (2018). *Probability and statistics for engineering and the sciences*. Toronto, Ontario: Nelson.
- Georgii, H. (2013). *Stochastics: Introduction to probability and statistics*. Berlin: De Gruyter.
- Giri, N. C. (2019). *Introduction to probability and statistics*. London: Routledge.
- Johnson, R. A., Miller, I., & Freund, J. E. (2019). *Miller & Freunds probability and statistics for engineers*. Boston: Pearson Education.
- Laha, R. G., & Rohatgi, V. K. (2020). *Probability theory*. Mineola, NY: Dover Publications.
- Mathai, A. M., & Haubold, H. J. (2018). *Probability and statistics: A course for physicists and engineers*. Boston: De Gruyter.
- Pishro-Nik, H. (2014). *Introduction to probability, statistics, and random processes*. Blue Bell, PA: Kappa Research, LLC.
- Spiegel, M. R., Schiller, J. J., & Srinivasan, R. A. (2013). *Schaums outline of probability and statistics*. New York: McGraw-Hill.

Unit 3: Congruential Random Number Generator

Contents

1.0 Introduction

As has been stated earlier, if you want to write a simulation program and you neither have a simulation language at your disposal nor a programming language with a random number generating function, you must design and write a random number generating function and call it whenever you need it.

Classical uniform random number generators have some major defects, such as, short period length and lack of higher dimension uniformity. However, nowadays there are a class of rather complex random number generators which are as efficient as the classical generators which enjoy the property of a much longer period and of higher dimension uniformity.

2.0 Intended Learning Outcomes (ILOs)

By the end of this unit, the reader should be able to:

- Explain the use Congruential method for generating Random numbers;
- Choose appropriate parameters for congruential method;
- Translate the method to computer programs;
- Use other very similar random number generating methods such as: Mid square, Mid product, Fibonacci

3.0 Main Content

3.1 The Congruential Method

The Congruential Method is widely used. The method is based on modulus arithmetic, which we now discuss briefly.

We say that two numbers x and y are congruent modulo m if (x-y) is an integral multiple of m. Thus we can write: x = y(modulo m)

For example, let m = 10, then we can write:

13 ≡ 3 (modulo 10)

84 ≡ 4 (modulo 10)

The ***congruential method*** generates random numbers by computing the next random number from the last random number obtained, given an initial random number say, X_0, called the seed.

The method uses the formula:

$X_{n+1} = (aX_n + c)$(modulo m) where X_0 = Seed and; a, c < m,

Where a, c and m are carefully chosen positive integer constants of which a and c must be less than m, X_0 is the seed or the last random number generated in the sequence. Stated in the computer language, the above formula becomes:

$X_{(N+1)} = (A^* X_{(N)} + C)$ MOD M (FORTRAN)

or

R = (A* SEED + C) MOD M (QBASIC)

From the above formula, it follows that the random number generated must be between 0 and (m-1) since MOD (modulo) produces remainder after division. Hence the above formula will produce the remainder after dividing ($aX_n + C$) by m. So to generate a random number between p and m we use:

X_{n+1}= (aX_n + C)(modulo m + 1-p) + p, for m > p.

If the value of c is zero, the congruential method is termed Multiplicative Congruential Method. If the value of c is not zero, the method is called Mixed Congruential Method.

The **multiplicative congruential** method is very handy. It is obtained using the general formula:

$r_n = ar_{n-1}$ (modulo m)

where the parameters a, m and the seed r_0 are specified to give desirable statistical properties of the resultant sequence. By virtue of modulo arithmetic, each r_n must be one of the numbers 0,1,2,3… m-1. Clearly, you must be careful about the choice of a and r_0. The

values of _a' and r_0 should be chosen to yield the largest cycle or period, that is to give the largest value for n at which $r_n = r_0$ for the first time.

Example 4

To illustrate the technique, suppose you want to generate ten decimal place numbers u_1, u_2, u_3,…. It can be shown that if you use

$u_n = r_n \times 10^{-1}$

where $r_n = 100003r_{n-1}$ (modulo 10^{10}), and r_0 = any odd number not divisible by 5,

then the period of the sequence will be 5×10^8, that is $r_n = r_0$ for the first time at $n = 5 \times 10^8$ and the cycle subsequently repeats itself.

As an example, using our mixed congruential formula

$X_{n+1} = (aX_n+c)$ (modulo m),

And suppose m = 8, a = 5, c = 7 and X_0 (seed) = 4 we can generate a random sequence of integer numbers thus:

n	$X_{n+1} = (5X_n+7) \bmod 8$		
0	X1 =	(5*X0+7)mod 8 =	(5*X4+7)mod 8 = 27 mod 8 = 3
1	X2 =	(5*X1+7)mod 8 =	(5*X3+7)mod 8 = 22 mod 8 = 6
2	X3 =	(5*X2+7)mod 8 =	(5*X6+7)mod 8 = 37 mod 8 = 5
3	X4 =	(5*X3+7)mod 8 =	(5*X5+7)mod 8 = 32 mod 8 = 0
4	X5 =	(5*X4+7)mod 8 =	(5*X0+7)mod 8 = 7 mod 8 = 7
5	X6 =	(5*X5+7)mod 8 =	(5*X7+7)mod 8 = 42 mod 8 = 2
6	X7 =	(5*X6+7)mod 8 =	(5*X2+7)mod 8 = 17 mod 8 = 1
7	X8 =	(5*X7+7)mod 8 =	(5*X1+7)mod 8 = 12 mod 8 = 4

Note that the value of X_8 is 4, which is the value of the seed X_0. So if we compute X_9, X_{10}, etc the same random numbers 3,6,5,0,7,2,1,4 will be generated once more.

Note also that if we divide the random integer values by 8, we obtain random numbers in the range $0 \leq X_{n+1} < 1$ which is similar to using the RND function of BASIC.

3.1 Choice of a, c and m

The method of this random number generation by *linear congruential method*, works by computing each successive random number from the previous. Starting with a seed, Xo, the linear congruential method uses the following formula:

$X_{i+1} = (A*Xi + C) \bmod M$

In his book, *The Art of Computer Programming*, Donald Knuth presents several rules for maximizing the length of time before the random number generator comes up with the same value as the seed. This is desirable because once the random number generator comes up with the initial seed, it will start to repeat the same sequence of random numbers (which

will not be so random since the second time around we can predict what they will be). According to Knuth's rules, if M is prime, we can let C be 0.
The LCM defined above has full period if and only if the following conditions are satisfied:

a) m and c are relatively prime
b) If q is a prime number that divides m , then q divides a-1
c) If 4 divides m, then 4 divides a-1

Therefore, the values for a, c and m are not generated randomly, rather they are carefully chosen based on certain considerations. For a binary computer with a word length of r bits, the normal choice for m is $m = 2^{r-1}$. With this choice of **m**, **a** can assume any of the values 1, 5,9,13, and **c** can assume any of the values 1, 3, 5, 7… However, experience shows that the congruential method works out very well if the value of a is an odd integer not divisible by either 3 or 5 and c chosen such that c mod 8 = 5 (for a binary computer) or c mod 200 = 21 (for a decimal computer).

Example 5
Develop a function procedure called RAND in QBASIC which generates a random number between 0 and 1 using the mixed congruential method. Assume a 16-bit computer.

Solution

```
FUNCTION RAND (SEED)
CONST M = 32767, A = 2743, C = 5923
IF SEED < 0 THEN SEED = SEED + M
SEED = (A* SEED + C) MOD M
RAND = SEED/M
END FUNCTION
```

Note that in the main program that references the above function in (a), the TIMER function can be used to generate the SEED to be passed to the function RAND as illustrated in example 2.

Example 5
Write a program that can generate that can generate 20 random integer number distributed between 1 and 64 inclusive using mixed congruential method.

Solution
QBASIC

```
DECLARE RAND (X)
CLS: REM Mixed Congruential Method
DIM SHARED SEED
SEED = TIMER
FOR K% = 1 TO 20
```

```
        SEED = RAND (SEED)        _Call of function RAND PRINT SEED: SPC(2)
NEXT K%
END _End of main program

FUNCTION RAND (SEED) _Beginning of function subprogram
CONST M = 64 A = 27, C = 13
IF SEED = THEN SEED = SEED + M
SEED = (a* SEED + C) MOD M + 1
RAND = SEED
END FUNCTION _End of the function program RAND
```

(b) Using FORTRAN

```
PROGRAM RANDNUM
COMMON SEED
CLS    _Clear screen
DO 50 K = 1, 25
WRITE (*, 5)
FORMAT(/)
50      CONTINUE
WRITE(*,*) _Enter the seed‘
READ(*,*) SEED
DO 30 J = 1, 20
SEED = RAND
WRITE (*, 6) SEED
6       FORMAT (I4)
30      CONTINUE
END

FUNCTION RAND
COMMON SEED
PARAMETER (M = 64, A = 27, C =13)
IF (SEED.LT.0) SEED = SEED + M
HOLD = (A*SEED + C)
SEED = AMOD (HOLD,M) + 1
RAND = SEED
RETURN END
```

3.3 RANECU Random Number Generator

A FORTRAN code for generating uniform random numbers on [0,1]. RANECU is multiplicative linear congruential generator suitable for a 16-bit platform. It combines three simple generators, and has a period exceeding 81012.

It is constructed for more efficient use by providing for a sequence of such numbers (Length), to be returned in a single call. A set of three non-zero integer seeds can be supplied,

failing which a default set is employed. If supplied, these three seeds, in order, should lie in the ranges [1,32362], [1,31726] and [1,31656] respectively. The program is given below.

```
      SUBROUTINE RANECU (RVEC,LEN)
C Portable random number generator for 16 bit computer.
C Generates a sequence of LEN pseudo-random numbers, returned
C in RVEC.
      DIMENSION RVEC(*)
      SAVE ISEED1,ISEED2, ISEED3
      DATA ISEED1,ISEED2,ISEED3/1234, 5678, 9876/
C Default values, used if none is suppliednvia an ENTRY
C call at RECUIN
      DO 100 I = 1,LEN
      K=ISEED1/206
ISEED1 = 157 * (ISEED1 - K * 206) - K * 21
IF(ISEED1.LT.0) ISEED1=ISEED1+32363
K=ISEED2/217
ISEED2 = 146 * (ISEED2 - K*217) - K* 45
IF(ISEED2.LT.O) ISEED2=ISEED2+31727
K=ISEED3/222
ISEED3 = 142 * (ISEED3 - K *222) - K * 133
IF(ISEED3.LT.0) ISEED3=ISEED3+31657
IZ=ISEED1-ISEED2
IF(IZ.GT.706)IZ = Z - 32362 IZ = 1Z+ISEED3
IF(IZ.LT.1)IZ = 1Z + 32362
RVEC(I)=REAL(IZ) * 3.0899E – 5
100   CONTINUE
RETURN
ENTRY RECUIN(IS1, IS2, IS3)
ISEED1=IS1
ISEED2=IS2
ISEED3=IS3
RETURN
ENTRY RECUUT(IS1,IS2,IS3)
IS1=ISEED1
IS2=ISEED2
IS3=ISEED3
RETURN
END
```

3.4 Other Methods of Generating Random Numbers

Some other methods of generating random numbers are Quadratic Congruential Method, Mid-square method, Mid-product Method and Fibonnachi Method.

3.4.1 The Quadratic congruential method

This method uses the formula:

$Xn=1 = (dX^2 + cX + a) \text{ modulo } m_n$ n

Where d is chosen in the same way as c and m should be a power of 2 for the method to yield satisfactory results.

3.4.2 The Mid-square method

The first random number is generated from the seed by squaring the seed and discarding all the digits except the middle four digits. This number is subsequently used as the new seed to generate the next random number in the same manner and so on.

The formula is: $X_{n+1} = X^2$ n

The mid-square method is rarely used these days as it has the tendency to degenerate rapidly. Also, if the number zero is ever generated, then all subsequent numbers generated will be zero. Furthermore, the method is slow when simulated in the computer since many multiplications and divisions are required to access the middle four digits.

3.4.3 The mid-product method

This method is similar to the mid-square method, except that a successive random number is obtained by multiplying the current number by a constant c, and taking the middle digits.

The formula is: $X_{n+1} = cX_n$

The mid-product method has a longer period and it is more uniformly distributed than the mid-square method.

3.4.4 The Fibonacci method

Fibonacci method uses the formula: $X_{n+1} = (X_n + X_{n-1}) \text{ modulo } m$

In this method, two initial seeds need to be provided. However, experience has shown that the random numbers generated by using Fibonacci method fail to pass tests for randomness. Therefore, the method does not give satisfactory results.

From the foregoing discussions, it is obvious that the last three methods – mid-square, mid-product and Fibonacci are of historical significance and have detrimental and limiting characteristics.

4.0 Self-Assessment Exercise(s)

1. Write a QBASIC program using Quadratic congruential method to generate 15 random integer numbers between 1 and 50.
2. Produce a table of random numbers using multiplicative congruential method, using a =5, m =8 and $X_0 = 4$. Draw an inference from your solution.
3. Write a QBASIC function that can be referenced as computer random number between 30 and 100 using mixed congruential method.
4. Use the mixed congruential method to generate the following sequences of random numbers:

a. A sequence of 10 one-digit random numbers given that $X_{n+1} \equiv (X_n + 3)$(modulo 10) and $X_0 = 2$
b. A sequence of eight random numbers between 0 and 7 given that $X_{n+1} \equiv (5X_n + 1)$(modulo 8) and $X_0 = 4$
c. A sequence of two-digit random numbers such that $X_{n+1} \equiv (61X_n + 27)$(modulo 100) and $X_0 = 40$
d. A sequence of five-digit random numbers such that $X_{n+1} \equiv (21X_n + 53)$(modulo 100) and $X_0 = 33$

5. Define a methods period and state how to improve a period. Show two examples of such improvement.
6. Consider the multiplicative congruential method for generating random digits. In each part below, assume modulo 10 arithmetic and determine the length of the cycle:
    a. Let a = 2 and r_0 = 1, 3 and 5
    b. Let a = 3 and r_0 = 1,2 and 5

5.0 Conclusion

In this unit, you have been introduced to Random Numbers generation. You have also learnt how to design random number generator.

6.0 Summary

What you have learnt in this unit concern:

- The Congruential methods of generating random numbers,
- The use of QBasic RND function to simulate randomness,
- The other Random number generating methods,
- The properties of good random number generator.

7.0 Further Readings

- Devore, J. L. (2018). *Probability and statistics for engineering and the sciences*. Toronto, Ontario: Nelson.
- Georgii, H. (2013). *Stochastics: Introduction to probability and statistics*. Berlin: De Gruyter.
- Giri, N. C. (2019). *Introduction to probability and statistics*. London: Routledge.
- Johnson, R. A., Miller, I., & Freund, J. E. (2019). *Miller & Freunds probability and statistics for engineers*. Boston: Pearson Education.
- Laha, R. G., & Rohatgi, V. K. (2020). *Probability theory*. Mineola, NY: Dover Publications.
- Mathai, A. M., & Haubold, H. J. (2018). *Probability and statistics: A course for physicists and engineers*. Boston: De Gruyter.

- Pishro-Nik, H. (2014). *Introduction to probability, statistics, and random processes*. Blue Bell, PA: Kappa Research, LLC.
- Spiegel, M. R., Schiller, J. J., & Srinivasan, R. A. (2013). *Schaums outline of probability and statistics*. New York: McGraw-Hill.

Unit 4: Monte Carlo Methods

Contents

1.0 Introduction

Monte Carlo methods (or Monte Carlo experiments) are a class of computational algorithms that rely on repeated random sampling to compute their results. Monte Carlo methods are often used in simulating physical and mathematical systems. The methods are especially useful in studying systems with a large number of coupled (interacting) degrees of freedom, such as fluids, disordered materials, strongly coupled solids, and cellular structures. More broadly, Monte Carlo methods are useful for modelling phenomena with significant uncertainty in inputs, such as the calculation of risk in business. These methods are also widely used in mathematics: a classic use is for the evaluation of definite integrals, particularly multidimensional integrals with complicated boundary conditions. It is a widely successful method in risk analysis when compared with alternative methods or human intuition. When Monte Carlo simulations have been applied in space exploration and oil exploration, actual observations of failures, cost overruns and schedule overruns are routinely better predicted by the simulations than by human intuition or alternative "soft" methods.

2.0 Intended Learning Outcomes (ILOs)

By the end this unit, you should be able to:

- Describe Monte Carlo method
- Trace the origin of Monte Carlo method
- Give examples of the application of Monte Carlo method

3.0 Main Content

3.1 Overview of Monte Carlo Method

There is no single Monte Carlo method; instead, the term describes a large and widely- used

class of approaches.
A **Monte Carlo algorithm** is an algorithm for computers. It is used to simulate the behaviour of other systems. It is not an exact method, but a heuristic one. Usually it uses randomness and statistics to get a result.

It is a computation process that uses random numbers to produce an outcome(s). Instead of having fixed inputs, probability distributions are assigned to some or all of the inputs. This will generate a probability distribution for the output after the simulation is run

However, these methods tend to follow the algorithm below:

1. Define a domain of possible inputs.
2. Generate inputs randomly from the domain using a certain specified probability distribution.
3. Perform a deterministic computation using the inputs.
4. Aggregate the results of the individual computations into the final result.

For example, to approximate the value of π using a Monte Carlo method:

1. Draw a square on the ground, then inscribe a circle within it. From planegeometry, the ratio of the area of an inscribed circle to that of the surrounding square is π / 4.
2. Uniformly scatter some objects of uniform size throughout the square. For example, grains of rice or sand.
3. Since the two areas are in the ratio π / 4, the objects should fall in the areas in approximately the same ratio. Thus, counting the number of objects in the circle and dividing by the total number of objects in the square will yield an approximation for π / 4.
4. Multiplying the result by 4 will then yield an approximation for π itself.

Notice how the π approximation follows the general pattern of Monte Carlo algorithms. First, we define an input domain: in this case, it's the square which circumscribes our circle. Next, we generate inputs randomly (scatter individual grains within the square), then perform a computation on each input (test whether it falls within the circle). At the end, we aggregate the results into our final result, the approximation of π.
Note, also, two other common properties of Monte Carlo methods: the computation's reliance on good random numbers, and its slow convergence to a better approximation as more data points are sampled. If grains are purposefully dropped into only, for example, the center of the circle, they will not be uniformly distributed, and so our approximation will be poor. An approximation will also be poor if only a few grains are randomly dropped into the whole square. Thus, the approximation of π will become more accurate both as the grains are dropped more uniformly and as more are dropped.

To understand the Monte Carlo method theoretically, it is useful to think of it as a general technique of numerical integration. It can be shown, at least in a trivial sense, that every application of the Monte Carlo method can be represented as a definite integral.

Suppose we need to evaluate a multi-dimensional definite integral of the form:

$$\Psi = \int_0^1 \int_0^1 \cdots \int_0^1 f(u_1, u_2, \ldots, u_n)\, du_1 du_2 \cdots du_n = \int_{(0,1)^n} f(\boldsymbol{u})\, d\boldsymbol{u} \qquad \text{……..6}$$

Most integrals can be converted to this form with a suitable change of variables, so we can consider this to be a general application suitable for the Monte Carlo method.

The integral represents a non-random problem, but the Monte Carlo method approximates a solution by introducing a random vector $\boldsymbol{U}$ that is uniformly distributed on the region of integration. Applying the function f to $\boldsymbol{U}$, we obtain a random variable $f(\boldsymbol{U})$. This has expectation:

$$E[f(\boldsymbol{U})] = \int_{(0,1)^n} f(\boldsymbol{u})\, \phi(\boldsymbol{u})\, d\boldsymbol{u} \qquad \text{……..(7)}$$

where ϕ is the probability density function of $\boldsymbol{U}$. Because ϕ equals 1 on the region of integration, [7] becomes:

$$E[f(\boldsymbol{U})] = \int_{(0,1)^n} f(\boldsymbol{u})\, d\boldsymbol{u} \qquad \text{……..(8)}$$

Comparing [6] and [8], we obtain a probabilistic expression for the integral Ψ:

$$\Psi = E[f(\boldsymbol{U})] \qquad \text{……..(9)}$$

so random variable $f(\boldsymbol{U})$ has mean Ψ and some standard deviation ζ. We define

$$H = f(\boldsymbol{U}^{[1]}) \qquad \text{…….(10)}$$

as an unbiased estimator for Ψ with standard error ζ. This is a little unconventional, since [10] is an estimator that depends upon a sample $\{\boldsymbol{U}^{[1]}\}$ of size one, but it is a valid estimator nonetheless.

To estimate Ψ with a standard error lower than ζ, let's generalize our estimator to accommodate a larger sample $\{\boldsymbol{U}^{[1]}, \boldsymbol{U}^{[2]}, \ldots, \boldsymbol{U}^{[m]}\}$. Applying the function f to each of these yields m independent and identically distributed (IID) random variables $f(\boldsymbol{U}^{[1]})$, $f(\boldsymbol{U}^{[2]})$, … , $f(\boldsymbol{U}^{[m]})$, each with expectation Ψ and standard deviation ζ. The generalization of [10]

$$H = \frac{1}{m}\sum_{k=1}^{m} f(\boldsymbol{U}^{[k]}) \qquad \text{…….(11)}$$

is an unbiased estimator for Ψ with standard error

$$\frac{\sigma}{\sqrt{m}} \qquad \text{……..(12)}$$

If we have a realization $\{\boldsymbol{u}^{[1]}, \boldsymbol{u}^{[2]}, \ldots, \boldsymbol{u}^{[m]}\}$ for our sample, we may estimate Ψ as:

$$h = \frac{1}{m}\sum_{k=1}^{m} f(\boldsymbol{u}^{[k]}) \qquad \text{…….(13)}$$

We call [11] the **crude Monte Carlo estimator**. Formula [12] for its standard error is important for two reasons. First, it tells us that the standard error of a Monte Carlo analysis decreases with the square root of the sample size. If we quadruple the number of

realizations used, we will half the standard error. Second, standard error does not depend upon the dimensionality of the integral [6]. Most techniques of numerical integration such as the trapezoidal rule or Simpson's method suffer from the **curse of dimensionality**. When generalized to multiple dimensions, the number of computations required to apply them, increases exponentially with the dimensionality of the integral. For this reason, such methods cannot be applied to integrals of more than a few dimensions. The Monte Carlo method does not suffer from the curse of dimensionality. It is as applicable to a 1000-dimensional integral as it is to a one-dimensional integral.
While increasing the sample size is one technique for reducing the standard error of a Monte Carlo analysis, doing so can be computationally expensive. A better solution is to employ some technique of variance reduction. These techniques incorporate additional information about the analysis directly into the estimator. This allows them to make the Monte Carlo estimator more deterministic, and hence have a lower standard error.

Due to high mathematics required and burden of understanding at this level, we have to stop this discussion here.

3.2 History of Monte Carlo Method

Physicists at Los Alamos Scientific Laboratory were investigating radiation shielding and the distance that neutrons would likely travel through various materials. Despite having most of the necessary data, such as the average distance a neutron would travel in a substance before it collided with an atomic nucleus or how much energy the neutron was likely to give off following a collision, the problem could not be solved with analytical calculations. John von Neumann and Stanislaw Ulam suggested that the problem be solved by modelling the experiment on a computer using chance. Being secret, their work required a code name. Von Neumann chose the name "Monte Carlo". The name is a reference to the Monte Carlo Casino in Monaco where Ulam's uncle would borrow money to gamble.

Random methods of computation and experimentation (generally considered forms of stochastic simulation) can be arguably traced back to the earliest pioneers of probability theory but are more specifically traced to the pre-electronic computing era. The general difference usually described about a Monte Carlo form of simulation is that it systematically "inverts" the typical mode of simulation, treating deterministic problems by *first* finding a probabilistic analogy. Previous methods of simulation and statistical sampling generally did the opposite: using simulation to test a previously understood deterministic problem. Though examples of an "inverted" approach do exist historically, they were not considered a general method until the popularity of the Monte Carlo method spread.

It was only after electronic computers were first built (from 1945 on) that Monte Carlo methods began to be studied in depth. In the 1950s they were used at Los Alamos for early work relating to the development of the hydrogen bomb, and became popularized in the fields of physics, physical chemistry, and operations research. The Rand Corporation and

the U.S. Air Force were two of the major organizations responsible for funding and disseminating information on Monte Carlo methods during this time, and they began to find a wide application in many different fields.
Uses of Monte Carlo methods require large amounts of random numbers, and it was their use that spurred the development of pseudorandom number generators, which were far quicker to use than the tables of random numbers which had been previously used for statistical sampling.

3.4 Applications of Monte Carlo Methods

As stated above, Monte Carlo simulation methods are especially useful for modelling phenomena with significant uncertainty in inputs and in studying systems with a large number of coupled degrees of freedom. Areas of application include:

a. Physical sciences:

Monte Carlo methods are very important in computational physics, physical chemistry, and related applied fields, and have diverse applications from complicated quantum calculations to designing heat shields and aerodynamic forms. The Monte Carlo method is widely used in statistical physics, particularly Monte Carlo molecular modelling as an alternative for computational molecular dynamics as well as to compute statistical field theories of simple particle and polymer models. In experimental particle physics, these methods are used for designing detectors, understanding their behaviour and comparing experimental data to theory, or on vastly large scale of the galaxy modelling.

Monte Carlo methods are also used in the models that form the basis of modern weather forecasting operations.

b. Engineering

Monte Carlo methods are widely used in engineering for sensitivity analysis and quantitative probabilistic analysis in process design. The need arises from the interactive, co-linear and non-linear behaviour of typical process simulations. For example,

- in **microelectronics** engineering, Monte Carlo methods are applied to analyze correlated and uncorrelated variations in analog and digital integrated circuits. This enables designers to estimate realistic 3 sigma corners and effectively optimise circuit yields.
- in **geostatistics and geometallurgy**, Monte Carlo methods strengthen the design of mineral processing flow sheets and contribute to quantitative risk analysis.

c. Visual Designs

Monte Carlo methods have also proven efficient in solving coupled integral differential equations of radiation fields and energy transport, and thus these methods have been used in global illumination computations which produce photorealistic images of virtual 3D models, with applications in video games, architecture, design and computer generated films.

d. Finance and business

Monte Carlo methods in finance are often used to calculate the value of companies, to evaluate investments in projects at a business unit or corporate level, or to evaluate financial derivatives. Monte Carlo methods used in these cases allow the construction of stochastic or probabilistic financial models as opposed to the traditional static and deterministic models, thereby enhancing the treatment of uncertainty in the calculation.

e. Telecommunications

When planning a wireless network, design must be proved to work for a wide variety of scenarios that depend mainly on the number of users, their locations and the services they want to use. Monte Carlo methods are typically used to generate these users and their states. The network performance is then evaluated and, if results are not satisfactory, the network design goes through an optimization process.

f. Games

Monte Carlo methods have recently been applied in game playing related artificial intelligence theory. Most notably the game of Battleship have seen remarkably successful Monte Carlo algorithm based computer players. One of the main problems that this approach has in game playing is that it sometimes misses an isolated, very good move. These approaches are often strong strategically but weak tactically, as tactical decisions tend to rely on a small number of crucial moves which are easily missed by the randomly searching Monte Carlo algorithm.

Monte Carlo simulation versus "what if" scenarios

The opposite of Monte Carlo simulation might be considered deterministic modelling using single-point estimates. Each uncertain variable within a model is assigned a ―best guess‖ estimate. Various combinations of each input variable are manually chosen (such as best case, worst case, and most likely case), and the results recorded for each so-called ―what if‖ scenario.

By contrast, Monte Carlo simulation considers random sampling of probability distribution functions as model inputs to produce hundreds or thousands of possible outcomes instead of a few discrete scenarios. The results provide probabilities of different outcomes occurring.

For example, a comparison of a spreadsheet cost construction model run using traditional ―what if‖ scenarios, and then run again with Monte Carlo simulation and Triangular probability distributions shows that the Monte Carlo analysis has a narrower range than the ―what if‖ analysis. This is because the ―what if‖ analysis gives equal weight to all scenarios.

A **randomized algorithm** or **probabilistic algorithm** is an algorithm which employs a degree of randomness as part of its logic. The algorithm typically uses uniformly distributed random bits as an auxiliary input to guide its behaviour, in the hope of achieving good performance in the "average case" over all possible choices of random bits. Formally, the algorithm's performance will be a random variable determined by the

random bits; thus either the running time, or the output (or both) are random variables
In common practice, randomized algorithms are approximated using a pseudorandom number generator in place of a true source of random bits; such an implementation may deviate from the expected theoretical behaviour.

g. Uses in mathematics

In general, Monte Carlo methods are used in mathematics to solve various problems by generating suitable random numbers and observing that fraction of the numbers which obeys some property or properties. The method is useful for obtaining numerical solutions to problems which are too complicated to solve analytically. The most common application of the Monte Carlo method in mathematics are:

i. Integration

Deterministic methods of numerical integration usually operate by taking a number of evenly spaced samples from a function. In general, this works very well for functions of one variable. However, for functions of vectors, deterministic quadrature methods can be very inefficient. To numerically integrate a function of a two-dimensional vector, equally spaced grid points over a two-dimensional surface are required. For instance a 10x10 grid requires 100 points. If the vector has 100 dimensions, the same spacing on the grid would require 10^{100} points which is far too many to be computed. But 100 dimensions is by no means unusual, since in many physical problems, a "dimension" is equivalent to a degree of freedom.
Monte Carlo methods provide a way out of this *exponential time-increase*. As long as the function in question is reasonably well-behaved, it can be estimated by randomly selecting points in 100-dimensional space, and taking some kind of average of the function values at these points. By the law of large numbers, this method will display convergence (i.e. quadrupling the number of sampled points will halve the error, regardless of the number of dimensions).

ii. Optimization

Most Monte Carlo optimization methods are based on **random walks**. The program will move around a marker in multi-dimensional space, tending to move in directions which lead to a lower function, but sometimes moving against the gradient.

Another popular application for random numbers in numerical simulation is in numerical optimization (choosing the best element from some set of available alternatives). These problems use functions of some often large-dimensional vector that are to be minimized (or maximized). Many problems can be phrased in this way: for example a computer chess program could be seen as trying to find the optimal set of, say, 10 moves which produces the best evaluation function at the end. The travelling salesman problem is another optimization problem. There are also applications to engineering design, such as design optimization.

iii. Inverse problems

Probabilistic formulation of inverse problems leads to the definition of a probability distribution in the space models. This probability distribution combines ***a priori*** (prior knowledge about a population, rather than that estimated by recent observation) information with new information obtained by measuring some observable parameters (data). As, in the general case, the theory linking data with model parameters is nonlinear, the **aposteriori** probability in the model space may not be easy to describe (it may be multimodal, some moments may not be defined, etc.).

When analyzing an inverse problem, obtaining a maximum likelihood model is usually not sufficient, as we normally also wish to have information on the resolution power of the data. In the general case we may have a large number of model parameters, and an inspection of the marginal probability densities of interest may be impractical, or even useless. But it is possible to pseudorandomly generate a large collection of models according to the posterior probability distribution and to analyze and display the models in such a way that information on the relative likelihoods of model properties is conveyed to the spectator. This can be accomplished by means of an efficient Monte Carlo method, even in cases where no explicit formula for the a priori distribution is available.

h. Computational mathematics

Monte Carlo methods are useful in many areas of computational mathematics, where a *lucky choice* can find the correct result. A classic example is Rabin's algorithm for primality testing (algorithm which determines whether a given number is prime). It states that for any *n* which is not prime, a random *x* has at least a 75% chance of proving that *n* is not prime. Hence, if *n* is not prime, but *x* says that it might be, we have observed at most a 1-in-4 event. If 10 different random *x* say that "*n* is probably prime" when it is not, we have observed a one-in-a-million event. In general a Monte Carlo algorithm of this kind produces one correct answer with a guarantee that ***n* is composite, and *x* proves it so**, but another one without, but with a guarantee of not getting this answer when it is wrong too often; in this case at most 25% of the time.

Remark:

In physics, two systems are **coupled** if they are interacting with each other. Of special interest is the **coupling** of two (or more) vibratory systems (e.g. pendula or resonant circuits) by means of springs or magnetic fields, etc. Characteristic for a coupled oscillation is the effect of beat.

4.0 Self-Assessment Exercise(s)

Answer the following questions:

1. How is Monte Carlo method different in approach from the typical mode of simulation, in deterministic problems?
2. How is Monte Carlo method used in Engineering and Mathematics?

5.0 Conclusion

Monte Carlo methods, relies on repeated computation of random or pseudo-random numbers. These methods are most suited to computations by a computer and tend to be used when it is unfeasible or impossible to compute an exact result with a deterministic algorithm (i.e. an algorithm whose behaviour can be completely predicted from the input)

6.0 Summary

In this unit we discussed the following:

- The algorithm of Monte Carlo method
- The history of Monte Carlo method which spurred the development of pseudorandom number generator
- The application of Monte Carlo methods in areas such as physical sciences, Engineering, Finance and Business, telecommunications, Games, Mathematics, etc.

7.0 Further Readings

- Devore, J. L. (2018). *Probability and statistics for engineering and the sciences*. Toronto, Ontario: Nelson.
- Georgii, H. (2013). *Stochastics: Introduction to probability and statistics*. Berlin: De Gruyter.
- Giri, N. C. (2019). *Introduction to probability and statistics*. London: Routledge.
- Johnson, R. A., Miller, I., & Freund, J. E. (2019). *Miller & Freunds probability and statistics for engineers*. Boston: Pearson Education.
- Laha, R. G., & Rohatgi, V. K. (2020). *Probability theory*. Mineola, NY: Dover Publications.
- Mathai, A. M., & Haubold, H. J. (2018). *Probability and statistics: A course for physicists and engineers*. Boston: De Gruyter.
- Pishro-Nik, H. (2014). *Introduction to probability, statistics, and random processes*. Blue Bell, PA: Kappa Research, LLC.
- Spiegel, M. R., Schiller, J. J., & Srinivasan, R. A. (2013). *Schaums outline of probability and statistics*. New York: McGraw-Hill.

Unit 5: Statistical Distribution Functions

Contents

1.0 Introduction

Although simulation can be a valuable tool for better understanding the underlying mechanisms that control the behaviour of a system, using simulation to make ***predictions*** of the future behaviour of a system can be difficult. This is because, for most real-world systems, at least some of the controlling parameters, processes and events are often stochastic, uncertain and/or poorly understood. The objective of many simulations is to identify and quantify the risks associated with a particular option, plan or design. Simulating a system in the face of such uncertainty and computing such risks requires that the uncertainties be quantitatively included in the calculations. To do this we collect data about the system parameters and subject them to statistical analysis.

2.0 Intended Learning Outcomes (ILOs)

After studying this unit the reader should be able to

- Define Statistics
- Explain Statistical Distributions

- Compute measures of Central Tendency and Variations
- Explain the Components of Statistical Distributions
    - Normal Distributions,
    - z-score
    - percentile,
    - Skewed Distributions
    - Ways to transform data to Graphs

3.0 Main Content

3.1 What is Statistics?

The field of statistics is concerned with the collection, description, and interpretation of data (data are numbers obtained through measurement). In the field of statistics, the term ―statistic‖ denotes a measurement taken on a sample (as opposed to a population). In general conversation, ―statistics‖ also refers to facts and figures.

3.2 What is a Statistical Distribution?

A statistical distribution describes the numbers of times each possible outcome occurs in a sample. If you have 10 test scores with 5 possible outcomes of A, B, C, D, or F, a statistical distribution describes the relative number of times an A,B,C,D or F occurs. For example, 2 A‘s, 4 B‘s, 4 C‘s, 0 D‘s, 0 F‘s.

3.3 Measures of Central Tendency

Suppose we have a sample with the following 4 observations: 4, 1, 4, 3.

Mean - the sum of a set of numbers divided by the number of observations.

$$\text{Mean} = \frac{4+1+4+3}{4} = \frac{12}{4} = 3$$

Median - the middle point of a set of numbers (for odd numbered samples). the mean of the middle two points (for even samples).

$$\text{Median}=1,\underline{3},\underline{4},4 \text{ or } \frac{3+4}{2}=\frac{7}{2}=3.5$$

Mode - the most frequently occurring number.

Mode = 4 (4 occurs most).

The mean, median and mode are called measures of central tendency.

3.4 Measures of Variation

Range - the maximum value minus the minimum value in a set of numbers. Range = 4-1 = 3.

Standard Deviation - the average distance a data point is away from the mean.

$$\text{standard deviation} = \frac{|4-3|+|1-3|+|4-3|+|3-3|}{4} = \frac{1+2+1+0}{4} = \frac{4}{4} = 1$$

Standard deviation computes the difference between each data point and the mean. Take the absolute value of each difference. Sum the absolute values. Divide this sum by the number of data points. Median: first arrange data points in increasing order.

Mean, Median, Mode, Range, and Standard Deviations are measurements in a sample (statistics) and can also be used to make inferences on a population.

3.5 Showing Data Distribution in Graphs

- **Bar graphs** use bars to compare frequencies of possible data values (see Fig a).
- **Double bar graphs** use two sets of bars to compare frequencies of data values between two levels of data (e.g. boys and girls) (see fig b).
- **Histograms** use bars to show how frequently data occur within equal spaces within an interval (see fig c & d).
- **Pie Charts** use portion of a circle to show contributions of data values (see fig c & d).

3.6 The Difference between a Continuous and a Discrete Distribution

Continuous distributions describe an infinite number of possible data values (as shown by the curve). For example someone‘s height could be 1.7m, 1.705m, 1.71m, ...

Discrete distributions describe a finite number of possible values. (shown by the bars)

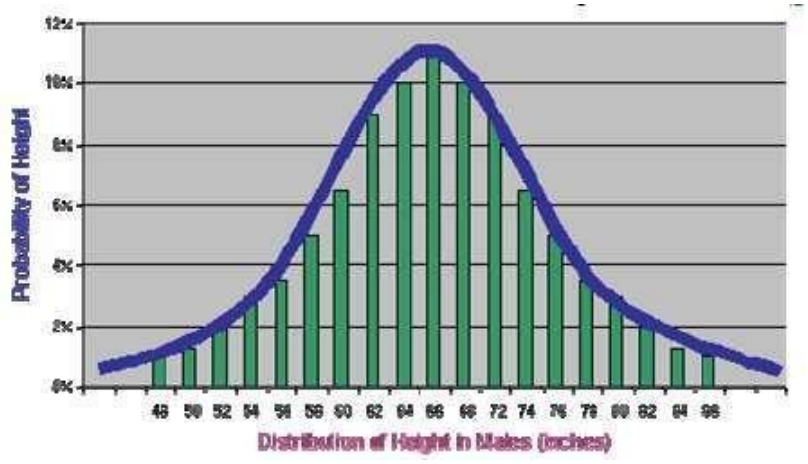


Fig 2: Distribution of Height in Males

3.7 Normal Distribution

A normal distribution is a continuous distribution that is ―bell-shaped‖. Data are often

assumed to be normal. Normal distributions can estimate probabilities over a continuous interval of data values.
The **normal distribution** refers to a family of continuous probability distributions described by the normal equation.

In a normal distribution, data are most likely to be at the mean. Data are less likely to be farther away from the mean.
The normal distribution is defined by the following equation:

$$Y = [\ 1/\zeta * \text{sqrt}(2\pi)\ ] * e^{-(x\ -\ \mu)2/2\zeta2}$$

where X is a normal random variable, μ is the mean, ζ is the standard deviation, π is approximately 3.14159, and e is approximately 2.71828.
The random variable X in the normal equation is called the **normal random variable**. The normal equation is the probability density function for the normal distribution.

The graph of the normal distribution depends on two factors - the mean and the standard deviation. The mean of the distribution determines the location of the center of the graph, and the standard deviation determines the height and width of the graph. When the standard deviation is large, the curve is short and wide; when the standard deviation is small, the curve is tall and narrow. All normal distributions look like a symmetric, bell-shaped curve, as shown in figure 3a and 3b.

(a) (b)

Fig. 3: Graph of Normal Distribution Based on Size of Mean and Standard Deviation

The curve on the left is shorter and wider than the curve on the right, because the curve on the left has a bigger standard deviation.

3.7.1 Standard Normal Distribution

The **standard normal distribution** is a special case of the normal distribution. It is the distribution that occurs when a normal random variable has a mean of zero and a standard deviation of one.

The normal random variable of a standard normal distribution is called a **standard score** or a **z-score**. Every normal random variable X can be transformed into a z score via the following equation:

$$z = (X - \mu) / \zeta$$

where X is a normal random variable, μ is the mean of X, and ζ is the standard deviation of X.

3.7.2 The Normal Distribution as a Model for Measurements

Often, phenomena in the real world follow a normal (or near-normal) distribution. This allows researchers to use the normal distribution as a model for assessing probabilities associated with real-world phenomena. Typically, the analysis involves two steps.

- Transform raw data. Usually, the raw data are not in the form of z-scores. They need to be transformed into z-scores, using the transformation equation presented earlier: $z = (X - \mu) / \zeta$.
- Find the probability. Once the data have been transformed into z-scores, you can use standard normal distribution tables, online calculators (e.g., Stat Trek's free normal distribution calculator) to find probabilities associated with the z-scores.

The problem in the next section demonstrates the use of the normal distribution as a model for measurement.

Example 1 - Ada earned a score of 940 on a national achievement test. The mean test score was 850 with a standard deviation of 100. What proportion of students had a higher score than Ada? (Assume that test scores are normally distributed.)

Solution - As part of the solution to this problem, we assume that test scores are normally distributed. In this way, we use the normal distribution as a model for measurement. Given an assumption of normality, the solution involves three steps.

- First, we transform Ada's test score into a z-score, using the z-scoretransformation equation.
  $z = (X - \mu) / \zeta = (940 - 850) / 100 = 0.90$
- Then, using a standard normal distribution table, we find the cumulative probability associated with the z-score. In this case, we find $P(Z < 0.90) = 0.8159$.
- Therefore, the $P(Z > 0.90) = 1 - P(Z < 0.90) = 1 - 0.8159 = 0.1841$.

Thus, we estimate that 18.41 percent of the students tested had a higher score than Ada.

Example 2 - An average light bulb manufactured by the Acme Corporation lasts 300 days with a standard deviation of 50 days. Assuming that bulb life is normally distributed, what is the probability that an Acme light bulb will last at most 365 days?

Solution: Given a mean score of 300 days and a standard deviation of 50 days, we want to find the cumulative probability that bulb life is less than or equal to 365 days. Thus, we know the following:

- The value of the normal random variable is 365 days.
- The mean is equal to 300 days.
- The standard deviation is equal to 50 days.

We enter these values into the formula and compute the cumulative probability. The answer is: $P(X \leq 365) = 0.90$. Hence, there is a 90% chance that a light bulb will burn out within 365 days.

3.7.3 Conversion to a Standard Normal Distribution

The values for points in a standard normal distribution are **z-scores**. We can use a standard normal table to find the probability of getting at or below a z-score. (a percentile).

- Subtract the mean from each observation in your normal distribution, the new mean=0.
- Divide each observation by the standard deviation, the new standard deviation=1.

3.7.4 Skewed Distributions μ

Skewness is the degree of asymmetry or departure from symmetry, of a distribution. Skewed distributions are not symmetric. If the frequency curve of a distribution has a longer tail to the right of the right of the central maximum than to the left, the distribution is said to be skewed to the right, or have a positive skewness. If the reverse is the case, it is said to be skewed to the left or negative skewness.

For skewed distributions, the mean tend to lie on the same side of the mode as the longer tail. Thus a measure of the asymmetry is supplied by the difference:

Mean – mode. This can be made dimensionless if we divide it by a measure of dispersion, such as the standard deviation, leading to the definition:

$$\text{Skewness} = \frac{mean - \text{mod}\, e}{SD} = \frac{\mu - \text{mod}\, e}{s} \quad (1)$$

To avoid using mode, we can use the empirical formula:

$$\text{Skewness} = \frac{3(mean - \text{median})}{SD} = \frac{3(\mu - \text{median})}{s} \quad (2)$$

Equations (1) and (2) are called; Pearson‘s first and second coefficients of skewness.

3.8 What is a Percentile?

A percentile (or **cumulative probability**) is the proportion of data in a distribution less than or equal to a data point. If you scored a 90 on a math test and 80% of the class had scores of 90 or lower; your percentile is 80. In the figure 4, b=90 and P(Z<b)=80.

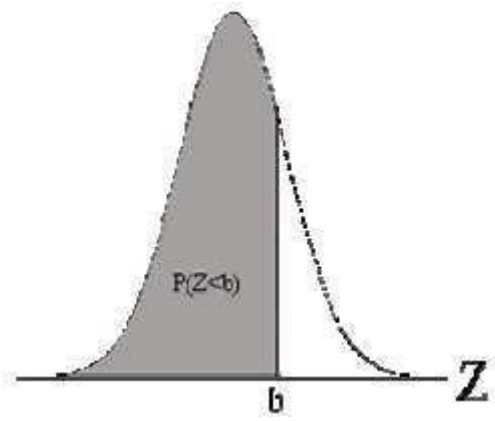


Fig. 4: illustration of Percentiles

3.9 Probabilities in Discrete Distributions

Suppose for your 10 tests you received 5 As, 2 Bs, 2 Cs, 1 D and want to find the probability of receiving an A or a B. Sum the frequencies for A and B and divide by the sample size. The probability of receiving an A or a B is (5+2)/10 = .7 (a 70% chance).

3.10 Probability and the Normal Curve

The normal distribution is a continuous probability distribution. This has several implications for probability.

- The total area under the normal curve is equal to 1.
- The probability that a normal random variable *X* equals any particular value is 0.
- The probability that *X* is greater than b equals the area under the normal curve bounded by *b* and plus infinity (as indicated by the *non-shaded* area in the figure 4).
- The probability that *X* is less than *a* equals the area under the normal curve bounded by b and minus infinity (as indicated by the *shaded* area in the figure below).

4.0 Self-Assessment Exercise(s)

Answer the following questions:

1. Why Convert to a Standard Normal Distribution?
2. What is the difference between a Continuous and a Discrete Distribution?
3. Given the following: mean=279.76, median=279.06, mode=277.5 and SD=15.6, find the first and second coefficients of skewness
4. Find the mode, median and mean deviation of the following sets of data: (a) 3, 7, 9, 5 and (b) 8, 10, 9, 12, 4, 8, 2.

5.0 Conclusion

We use Statistical distributions to: investigate how a change in one variable relates to a change in a second variable, represent situations with numbers, tables, graphs, and verbal descriptions, understand measurable attributes of objects and their units, systems, and processes of measurement, identify relationships among attributes of entities or systems and their association.

6.0 Summary

In this unit:

- We defined Statistics as field of study that is concerned with the collection, description, and interpretation of data.
- We saw that Statistical Distributions describe the numbers of times each possible outcome occurs in a sample.
- We computed various measures of Central Tendency and Variations which can be used to make inferences.
- And explained the following components of Statistical Distributions:
    - Normal Distributions,
    - z-score
    - percentile,
    - Skewed Distributions
    - Ways to transform data to Graphs

7.0 Further Readings

- Devore, J. L. (2018). *Probability and statistics for engineering and the sciences*. Toronto, Ontario: Nelson.
- Georgii, H. (2013). *Stochastics: Introduction to probability and statistics*. Berlin: De Gruyter.
- Giri, N. C. (2019). *Introduction to probability and statistics*. London: Routledge.
- Johnson, R. A., Miller, I., & Freund, J. E. (2019). *Miller & Freunds probability and statistics for engineers*. Boston: Pearson Education.
- Laha, R. G., & Rohatgi, V. K. (2020). *Probability theory*. Mineola, NY: Dover Publications.
- Mathai, A. M., & Haubold, H. J. (2018). *Probability and statistics: A course for physicists and engineers*. Boston: De Gruyter.
- Pishro-Nik, H. (2014). *Introduction to probability, statistics, and random processes*. Blue Bell, PA: Kappa Research, LLC.
- Spiegel, M. R., Schiller, J. J., & Srinivasan, R. A. (2013). *Schaums outline of probability and statistics*. New York: McGraw-Hill.

Unit 6: Common Probability Distributions

Contents

1.0 Introduction

In this section we look at the branch of statistics that deals with analysis of random events. Probability is the numerical assessment of likelihood on a scale from 0 (impossibility) to 1 (absolute certainty). Probability is usually expressed as the ratio between the number of ways an event can happen and the total number of things that can happen (e.g., there are 13 ways of picking a diamond from a deck of 52 cards, so the probability of picking a diamond is 13/52, or ¼). Probability theory grew out of attempts to understand card games and gambling. As science became more rigorous, analogies between certain biological, physical, and social phenomena and games of chance became more evident (e.g., the sexes of newborn infants follow sequences similar to those of coin tosses). As a result, probability became a fundamental tool of modern genetics and many other disciplines.

2.0 Intended Learning Outcomes (ILOs)

By the end of this unit, the reader should be able to:

- Explain the role of probability distribution functions in simulations
- Describe Probability theory
- Explain the fundamental concepts of Probability theory
- Explain Random Variable
- Explain Limiting theorems
- Describe Probability distributions in simulations

- List common Probability distributions.

3.0 Main Content

3.1 Distribution Functions and Simulation

Many simulation tools and approaches are ***deterministic***. In a deterministic simulation, the input parameters for a model are represented using single values (which typically are described either as "the best guess" or "worst case" values). Unfortunately, this kind of simulation, while it may provide some insight into the underlying mechanisms, is not well-suited to making predictions to support decision-making, as it cannot quantitatively address the risks and uncertainties that are inherently present.

However, it is possible to quantitatively represent uncertainties in simulations. ***Probabilistic simulation*** is the process of explicitly representing these uncertainties by specifying inputs as probability distributions. If the inputs describing a system are uncertain, the prediction of future performance is necessarily uncertain. That is, the result of any analysis based on inputs represented by probability distributions is itself a probability distribution. Hence, whereas the result of a deterministic simulation of an uncertain system is a *qualified statement* ("if we build the dam, the salmon population could go extinct"), the result of a probabilistic simulation of such a system is a *quantified probability* ("if we build the dam, there is a 20% chance that the salmon population will go extinct"). Such a result (in this case, quantifying the risk of extinction) is typically much more useful to decision-makers who might utilize the simulation results.

3.2 Probability Definitions

The word *probability* does not have a consistent direct definition. In fact, there are two broad categories of **probability interpretations**, whose adherents possess different (and sometimes conflicting) views about the fundamental nature of probability:

1. Frequentists talk about probabilities only when dealing with experiments that are random and well-defined. The probability of a random event denotes the *relative frequency of occurrence* of an experiment's outcome, when repeating the experiment. Frequentists consider probability to be the relative frequency "in the long run" of outcomes.
2. Bayesians, however, assign probabilities to any statement whatsoever, even when no random process is involved. Probability, for a Bayesian, is a way to represent an individual's *degree of belief* in a statement, or an objective degree of rational belief, given the evidence

   The scientific study of probability is a modern development. Gambling shows that there has been an interest in quantifying the ideas of probability for millennia, but exact mathematical descriptions of use in those problems only arose much later.

Probability Distribution

A probability distribution gathers together all possible outcomes of a random variable (i.e. any quantity for which more than one value is possible), and summarizes these outcomes by indicating the probability of each of them. While a probability distribution is often associated with the bell-shaped curve, recognize that such a curve is only indicative of one specific type of probability, the so-called normal probability distribution. However, in real life, a probability distribution can take any shape, size and form.

Example: Probability Distribution

For example, if we wanted to choose a day at random in the future to schedule an event, and we wanted to know the probability that this day would fall on a Sunday, as we will need to avoid scheduling it on a Sunday. With seven days in a week, the probability that a random day would happen to be a Sunday would be given by one-seventh or about 14.29%. Of course, the same 14.29% probability would be true for any of the other six days.

In this case, we would have a uniform probability distribution: the chances that our random day would fall on any particular day are the same, and the graph of our probability distribution would be a straight line.

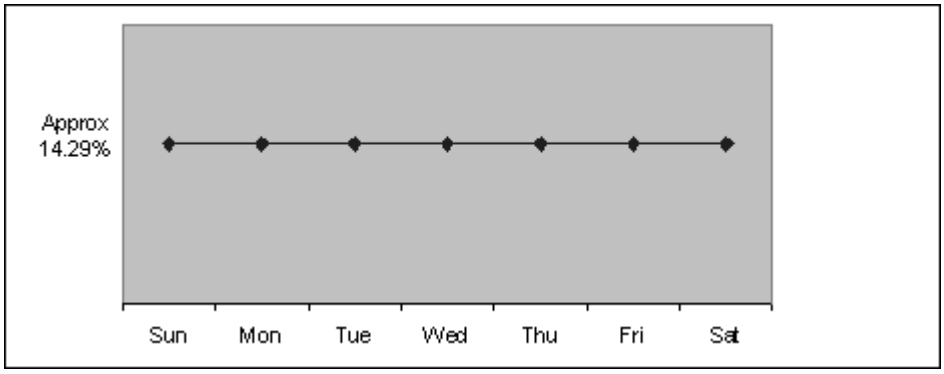


fig.1: Uniform Probability Distribution

Probability distributions can be simple to understand as in this example, or they can be very complex and require sophisticated techniques (e.g., option pricing models, Monte Carlo simulations) to help describe all possible outcomes.

3.3 Random Variables

Random variable is **discrete** random variables if it can take on a finite or countable number of possible outcomes. The previous example asking for a day of the week is an example of a discrete variable, since it can only take seven possible values. Monetary variables expressed in dollars and cents are always discrete, since money is rounded to the nearest $0.01.

A random variable is **continuous** random variable if it has has infinite possible outcomes. A rate of return (e.g. growth rate) is continuous:

- a stock can grow by 9% next year or by 10%, and in between this range it could grow by 9.3%, 9.4%, 9.5%
- Clearly there is no end to how precise the outcomes could be broken down; thus it's described as a continuous variable.

Outcomes in Discrete vs. Continuous Variables

The rule of thumb is that a discrete variable can have all possibilities listed out, while a continuous variable must be expressed in terms of its upper and lower limits, and greater-than or less-than indicators. Of course, listing out a large set of possible outcomes (which is usually the case for money variables) is usually impractical – thus money variables will usually have outcomes expressed as if they were continuous.

Examples:

- Rates of return can theoretically range from –100% to positive infinity.
- Time is bound on the lower side by 0.
- Market price of a security will also have a lower limit of $0, while its upper limit will depend on the security – stocks have no upper limit (thus a stock price's outcome $\geq$ $0),
- Bond prices are more complicated, bound by factors such as time-to-maturity and embedded call options. If a face value of a bond is $1,000, there's an upper limit (somewhere above $1,000) above which the price of the bond will not go, but pinpointing the upper value of that set is imprecise.

3.4 Probability Function

A probability function gives the probabilities that a random variable will take on a given list of specific values. For a discrete variable, if (x_1, x_2, x_3, x_4 …) are the complete set of possible outcomes, p(x) indicates the chances that X will be equal to x. Each x in the list for a discrete variable will have a p(x). For a continuous variable, a probability function is expressed as f(x).

The two key properties of a probability function, p(x) (or f(x) for continuous), are the following:

1. $0 \leq p(x) \leq 1$, since probability must always be between 0 and 1.
2. Add up all probabilities of all distinct possible outcomes of a random variable, and the sum must equal 1.

Determining whether a function satisfies the first property should be easy to spot since we know that probabilities always lie between 0 and 1. In other words, p(x) could never be 1.4 or –0.2. To illustrate the second property, say we are given a set of three possibilities

for X: (1, 2, 3) and a set of three for Y: (6, 7, 8), and given the probability functions f(x) and g(y).

x	f(x)	y	g(y)
1	0.31	6	0.32
2	0.43	7	0.40
3	0.26	8	0.23

For all possibilities of f(x), the sum is 0.31+0.43+0.26=1, so we know it is a valid probability function. For all possibilities of g(y), the sum is 0.32+0.40+0.23 = 0.95, which violates our second principle. Either the given probabilities for g(y) are wrong, or there is a fourth possibility for y where g(y) = 0.05. Either way it needs to sum to 1.

Probability Density Function

A probability density function (or pdf) describes a probability function in the case of a continuous random variable. Also known as simply the ―density‖, a probability density function is denoted by ―f(x)‖. Since a pdf refers to a continuous random variable, its probabilities would be expressed as ranges of variables rather than probabilities assigned to individual values as is done for a discrete variable. For example, if a stock has a 20% chance of a negative return, the pdf in its simplest terms could be expressed as:

x	f(x)
< 0	0.2
≥ 0	0.8

3.5 Mathematical Treatment of Probability

In mathematics, a probability of an event *A* is represented by a real number in the range from 0 to 1 and written as P(*A*), p(*A*) or Pr(*A*). An impossible event has a probability of 0, and a certain event has a probability of 1. However, the converses are not always true: probability 0 events are not always impossible, nor probability 1 events certain. The rather subtle distinction between "certain" and "probability 1" is treated at greater length in the article on "almost surely".

The *opposite* or *complement* of an event *A* is the event [not *A*] (that is, the event of *A* not occurring); its probability is given by P(not *A*) = 1 - P(*A*). As an example, the chance of not rolling a six on a six-sided die is 1 – (chance of rolling a six) $1 - \frac{1}{6} = \frac{5}{6}$.

Joint Probabilty

If both the events *A* and *B* occur on a single performance of an experiment this is called

the ***intersection or joint probability*** of *A* and *B*, denoted as and $P(A \cap B)$.

If two events, *A* and *B* are ***independent*** then the joint probability is:

$$P(A \text{ and } B) = P(A \cap B) = P(A)P(B),$$

for example, if two coins are flipped the chance of both being heads is: $\frac{1}{2} \times \frac{1}{2} = \frac{1}{4}$.

Mutually Exclusive Events

If either event *A* or event *B* or both events occur on a single performance of an experiment this is called the union of the events *A* and *B* denoted as $P(A \cup B)$. If two events are ***mutually exclusive*** then the probability of either occurring is:

$$P(A \text{ or } B) = P(A \cup B) = P(A) + P(B) - P(A \cap B) = P(A) + P(B) - 0 = P(A) + P(B)$$

.

For example, the chance of rolling a 1 or 2 on a six-sided die is

$$P(1 \text{ or } 2) = P(1) + P(2) = \tfrac{1}{6} + \tfrac{1}{6} = \tfrac{1}{3}.$$

If the events are not mutually exclusive then

$$P(A \text{ or } B) = P(A \cup B) = P(A) + P(B) - P(A \text{ and } B).$$

For example, when drawing a single card at random from a regular deck of cards, the chance of getting a heart or a face card (J,Q,K) (or one that is both) is $\frac{13}{52} + \frac{12}{52} - \frac{3}{52} = \frac{11}{26}$, because of the 52 cards of a deck 13 are hearts, 12 are face cards, and 3 are both: here the possibilities included in the "3 that are both" are included in each of the "13 hearts" and the "12 face cards" but should only be counted once.

Conditional Probability

This is the probability of some event *A*, given the occurrence of some other event *B*. Conditional probability is written *P*(*A*|*B*), and is read "the probability of *A*, given *B*". It is defined by:

$$P(A \mid B) = \frac{P(A \cap B)}{P(B)}.$$

If *P*(*B*) = 0 then $P(A \mid B)$ is undefined.

Summary of probabilities

Event	Probability
A	P(A) □ [0,1]
Not A	P(A) = 1-P(A)
A or B	P(A U B) = P(A) + P(B)-P(A∩B) = P(A)+P(B) If A and B are mutually exclusive

A and B	$P(A \cap B) = P(A\|B)P(B)$ $= P(A)P(B)$ If A and B are independent
A given B	$P(A\|B) = P(A \cap B)/P(B)$

Two or more events are mutually exclusive if the occurrence of any one of them excludes the occurrence of the others.

3.6 Probability theory

Like other theories, the theory of probability is a representation of probabilistic concepts in formal terms—that is, in terms that can be considered separately from their meaning. These formal terms are manipulated by the rules of mathematics and logic, and any results are then interpreted or translated back into the problem domain.

There have been at least two successful attempts to formalize probability, namely the Kolmogorov formulation and the Cox formulation. In Kolmogorov's formulation sets are interpreted as events and probability itself as a measure on a class of sets. In Cox's theorem, probability is taken as a primitive (that is, not further analyzed) and the emphasis is on constructing a consistent assignment of probability values to propositions. In both cases, the laws of probability are the same, except for technical details.

Probability theory is a mathematical science that permits one to find, using the probabilities of some random events, the probabilities of other random events connected in some way with the first.

The assertion that a certain event occurs with a probability equal, for example, to 1/2, is still not, in itself, of ultimate value, because we are striving for definite knowledge. Of definitive, cognitive value are those results of probability theory that allow us to state that the probability of occurrence of some event *A* is very close to 1 or (which is the same thing) that the probability of the non-occurrence of event *A* is very small. According to the principle of ―disregarding sufficiently small probabilities,‖ such an event is considered practically reliable. Such conclusions, which are of scientific and practical interest, are usually based on the assumption that the occurrence or non-occurrence of event *A* depends on a large number of factors that are slightly connected with each other.

Consequently, it can also be said that **probability theory** is a mathematical science that clarifies the regularities that arise in the interaction of a large number of random factors.

To describe the regular connection between certain conditions *S* and event *A*, whose occurrence or non-occurrence under given conditions can be accurately established, natural science usually uses one of the following schemes:

(a) For each realization of conditions *S*, event *A* occurs. All the laws of classical mechanics have such a form, stating that for specified initial conditions and forces acting on an object or system of objects, the motion will proceed in an unambiguously definite manner.

(b) Under conditions *S*, event *A* has a definite probability *P*(*A*/*S*) equal to *p*.

Thus, for example, the laws of radioactive emission assert that for each radioactive substance there exists the specific probability that, for a given amount of a substance, a certain number of atoms N will decompose within a given time interval.

Let us call the frequency of event A in a given set of n trials (that is, of n repeated realizations of conditions S) the ratio $h = m/n$ of the number m of those trials in which A occurs to the total number of trials n. The existence of a specific probability equal to p for an event A under conditions S is manifested in the fact that in almost every sufficiently long series of trials, the frequency of event A is approximately equal to p.

Statistical laws, that is, laws described by a scheme of type (b), were first discovered in games of chance similar to dice. The statistical rules of birth and death (for example, the probability of the birth of a boy is 0.515) have also been known for a long time. A great number of statistical laws in physics, chemistry, biology, and other sciences were discovered at the end of the 19th and in the first half of the 20th century.

The possibility of applying the methods of probability theory to the investigation of statistical laws, which pertain to a very wide range of scientific fields, is based on the fact that the probabilities of events always satisfy certain simple relationships, which will be discussed in the next section. The investigation of the properties of probabilities of events on the basis of these simple relationships is also a topic of probability theory.

3.6.1 Fundamental concepts of Probability theory.

The fundamental concepts of probability theory as a mathematical discipline are most simply defined in the framework of so-called elementary probability theory. Each trial T considered in elementary probability theory is such that it is ended by one and only one of the events $E_1, E_2, \dots, E_s$ (by one or another, depending on the case). These events are called outcomes of the trial. Each outcome E_k is connected with a positive number p_k, the probability of this outcome. The numbers p_k must add up to 1. Events A, which consist of the fact that ―either E_i, or E_j … , or E_k occurs,‖ are then considered. The outcomes $E_i, \dots, E_k$ are said to be favorable to A, and according to the definition, it is assumed that the probability $P(A)$ of event A is equal to the sum of the probabilities of the outcomes favorable to it:

$$(1)\quad P(A) = p_i + p_j + \dots + p_k$$

The particular case $p_1 = p_2 = p_s = 1/s$ leads to the formula:

$$(2)\quad P(A) = r/s$$

Formula (2) expresses the so-called classical definition of probability according to which the probability of some event A is equal to the ratio of the number r of outcomes favorable to A to the number s of all ―equally likely‖ outcomes. The classical definition of probability only reduces the concept of probability to the concept of equal possibility,

which remains without a clear definition.

EXAMPLE. In the tossing of two dice, each of the 36 possible outcomes can be designated by *(i, j),* where *i* is the number of pips that comes up on the first dice and *j*, the number on the second. The outcomes are assumed to be equally likely. To the event *A*, ―the sum of the pips is 4,‖ three outcomes are favorable: (1,3); (2,2); (3,1). Consequently, *P(A)* = 3/36 = 1/12.

Starting from certain given events, it is possible to define two new events: their union (sum) and intersection (product). Event *B* is called the **union** of events A_1, A_2, …, A_r if it has the form ―A_1 or A_2, …, or A_r occurs.‖

Event *C* is called the **intersection** of events A_1, A_2 …, A_r if it has the form ―A_1, and A_2, … , and A_r occurs.‖
The union of events is designated by the symbol ∪, and the intersection, by ∩. Thus, we write:

$B = A_1 \cup A_2 \cup ... \cup A_r$ $C =$
$A_1 \cap A_2 \cap ... \cap A_r$
Events *A* and *B* are called disjoint if their simultaneous occurrence is impossible—that is, if among the outcomes of a trial not one is favourable to *A* and *B* simultaneously.
Two of the basic theorems of probability theory are connected with the operations of union and intersection of events; these are the theorems of addition and multiplication of probabilities.

3.6.2 Theorem of Addition of Probabilities.

If events A_1, A_2, …, A_r are such that each two of them are disjoint, then the probability of their union is equal to the sum of their probabilities.

Thus, in the example presented above of tossing two dice, event *B*, ―the sum of the pips does not exceed 4,‖ is the union of three disjoint events A_2, A_3, A_4, consisting of the fact the sum of the pips is equal to 2, 3, and 4, respectively. The probabilities of these events are 1/36, 2/36, and 3/36, respectively. According to the theorem of addition of probabilities, probability *P(B)* is:

1/36 + 2/36 + 3/36 = 6/36 = 1/6

The conditional probability of event *B* under condition *A* is determined by the formula:

$$P(B|A) = \frac{P(A \cap B)}{P(A)}$$

which, as can be proved, corresponds completely with the properties of frequencies.
Events A_1, A_2, …, A_r are said to be independent if the conditional probability of each of them, under the condition that some of the remaining events have occurred, is equal to its

—absolute‖ probability.

3.6.3 Theorem of Multiplication of Probabilities.

The probability of the intersection of events A_1, A_2, …, A_r is equal to the probability of event A_1 multiplied by the probability of event A_2 under the condition that A_1 has occurred, …, multiplied by the probability of A_r under the condition that A_1, A_2, …, A_{r-1} have occurred. For independent events, the multiplication theorem reduces to the formula:

$$P(A_1 \cap A_2 \cap \ldots \cap A_r) = P(A_1) \times P(A_2) \times \ldots \times P(A_r) \quad \ldots\ldots (3)$$

that is, the probability of the intersection of independent events is equal to the product of the probabilities of these events. Formula (3) remains correct, if on both sides some of the events are replaced by their inverses.

EXAMPLE:

Four shots are fired at a target, and the hit probability is 0.2 for each shot. The target hits by different shots are assumed to be independent events. What is the probability of hitting the target three times?

Each outcome of the trial can be designated by a sequence of four letters [for example, (s, f, f, s) denotes that the first and fourth shots hit the target (success), and the second and third miss (failure)]. There are $2 \cdot 2 \cdot 2 \cdot 2 = 16$ outcomes in all. In accordance with the assumption of independence of the results of individual shots, one should use formula (3) and the remarks about it to determine the probabilities of these outcomes. Thus, the probability of the outcome (s, f, f, f) is set equal to 0.2 x 0.8 x 0.8 x 0.8 = 0.1024; here, 0.8 = 1 - 0.2 is the probability of a miss for a single shot. For the event —three shots hit the target,‖ the possible outcomes are: (s, s, s, f), (s, s, f, s), (s, f, s, s), and (f, s, s, s) are favorable and the probability of each is the same:

$0.2 \cdot 0.2 \cdot 0.2 \cdot 0.8 = \cdot \cdot \cdot = 0.8 \cdot 0.2 \cdot 0.2 \cdot .02 = 0.0064$

Consequently, the desired probability is 4 x 0.0064 = 0.0256.

Generalizing the discussion of the given example, it is possible to derive one of the fundamental formulas of probability theory: if events A_1, A_2, …, A_n are independent and each has a probability p, then the probability of exactly m such events occurring is:

$$P_n(m) = C_n^m (1-p)^{n-m} \quad \ldots\ldots (4)$$

Here, C^m_n denotes the number of combinations of n elements taken m at a time. For large n, the calculation using formula (4) becomes difficult. In the preceding example, let the number of shots equal 100; the problem then becomes one of finding the probability x that the number of hits lies in the range from 8 to 32. The use of formula (4) and the addition theorem gives an accurate, but not a practically useful, expression for the desired probability

$$x = \sum_{m=8}^{32} C_{100}^{m} (0.2)^m (0.8)^{100-m}$$

The approximate value of the probability x can be found by the **Laplace theorem**

$$x \approx \frac{1}{\sqrt{2\pi}} \int_{-3}^{+3} e^{-z^2/2}\, dz = 0.9973$$

with the error not exceeding 0.0009. The obtained result demonstrates that the event $8 \leq m \leq 32$ is practically certain. This is the simplest, but a typical, example of the use of the limit theorems of probability theory.

Another fundamental formula of elementary probability theory is the so-called total probability formula: if events $A_1, A_2, \ldots, A_r$ are disjoint in pairs and their union is a certain event, then the probability of any event B is the sum

$$P(B) = \sum_{k=1}^{r} P(B/A_k)P(A_k)$$

The theorem of multiplication of probabilities turns out to be particularly useful in the consideration of compound trials. Let us say that trial T consists of trials $T_1, T_2, \ldots, T_{n-1}, T_n$, if each outcome of trial T is the intersection of certain outcomes $A_i, B_i, \ldots, x_k, Y_l$ of the corresponding trials $T_1, T_2, \ldots, T_{n-1}, T_n$. From one or another consideration, the following probabilities are often known:

$$\boldsymbol{P(A_1), P(B_j/A_i), \ldots, P(Y_i/A_i \cap B_j \cap \ldots \cap X_k)} \quad \textbf{(5)}$$

According to the probabilities of (5), probabilities $P(E)$ for all the outcomes of E of the compound trial and, in addition, the probabilities of all events connected with this trial can be determined using the multiplication theorem (just as was done in the example above).

Two types of compound trials are the most significant from a practical point of view:

(a) the component trials are independent, that is, the probabilities (5) are equal to the unconditional probabilities $P(A_i), P(B_j), \ldots, P(Y_l)$; and

(b) the results of only the directly preceding trial have any effect on the probabilities of the outcomes of any trial—that is, the probabilities (5) are equal, respectively, to $P(A_i), P(B_j/A_i), \ldots, P(Y_l/X_k)$.

In this case, it is said that the trials are connected in a Markov chain. The probabilities of all the events connected with the compound trial are completely determined here by the initial probabilities $P(A_i)$ and the transition probabilities $P(B_j/A_i), \ldots, P(Y_l/X_k)$.

Often, instead of the complete specification of a probability distribution of a random variable, it is preferable to use a small number of numerical characteristics. The most frequently used are the mathematical expectation and the dispersion.

In addition to mathematical expectations and dispersions of these variables, a joint distribution of several random variables is characterized by correlation coefficients and so forth. The meaning of the listed characteristics is to a large extent explained by the **limit theorems**

3.7 The Limit theorems.

In the formal presentation of probability theory, limit theorems appear in the form of a superstructure over its elementary sections, in which all problems have a finite, purely arithmetic character. However, the cognitive value of probability theory is revealed only by the limit theorems. Thus, the **Bernoulli theorem** proves that in independent trials, the frequency of appearance of any event, as a rule, deviates little from its probability, and the **Laplace theorem** indicates the probabilities of one or another deviation. Similarly, the meaning of such characteristics of a random variable as its mathematical expectation and dispersion is explained by the law of large numbers and the **central limit theorem**.

Let $X_1, X_2, \ldots, X_n$, be independent random variables that have one and the same probability distribution with $\mathbf{E}X_K = a$, $\mathbf{D}X_K = \zeta^2$ and Y_n be the arithmetic mean of the first n variables of sequence such that:

$Y_n = (X_1 + X_2 + X_2 + \cdots + X_n)/n$

In accordance with the law of large numbers, for any $\varepsilon > 0$, the probability of the inequality $| Y_n - a | \leq \varepsilon$ has the limit 1 as $n \to \infty$, and thus Y_n, as a rule, differs little from a.

The **central limit theorem** makes this result specific by demonstrating that the deviations of Y_n from a are approximately subordinate to a normal distribution with mean zero and dispersion $8^2/n$. Thus, to determine the probabilities of one or another deviation of Y_n from a for large n, there is no need to know all the details about the distribution of the variables X_n; it is sufficient to know only their dispersion.

In the 1920‘s it was discovered that even in the scheme of a sequence of identically distributed and independent random variables, limiting distributions that differ from the normal can arise in a completely natural manner. Thus, for example, if X_1 is the time until the first reversion of some randomly varying system to the original state, and X_2 is the time between the first and second reversions, and so on, then under very general conditions the distribution of the sum $X_1 + \cdots + X_n$ (that is, of the time until the Aith reversion), after multiplication by $n^{-1/\alpha}$(α is a constant less than 1), converges to some limiting distribution. Thus, the time until the *nth* reversion increases, roughly speaking, as $n^{1/\alpha}$, that is, more rapidly than n (in the case of applicability of the law of large numbers, it is of the order of n).

The mechanism of the emergence of the majority of limiting regularities can be understood ultimately only in connection with the theory of random processes.

Random processes.

In a number of physical and chemical investigations of recent decades, the need has arisen to consider, in addition to one-dimensional and multidimensional random variables, random processes—that is, processes for which the probability of one or another of their courses is defined. In probability theory, a random process is usually considered as a one-parameter family of random variables $X(t)$. In an overwhelming number of applications, the parameter t represents time, but this parameter can be, for example, a point in space,

and then we usually speak of a random function. In the case when the parameter t runs through the integer-valued numbers, the random function is called a **random sequence**. Just as a random variable is characterized by a distribution law, a random process can be characterized by a set of joint distribution laws for $X(t_1)$, $X(t_2)$, …, $X(t_n)$ for all possible moments of t_1, t_2, …, t_n for any $n > 0$.

3.8 Probability Distribution Functions

In probability theory and statistics, a **probability distribution** identifies either the probability of each value of a random variable (when the variable is discrete), or the probability of the value falling within a particular interval (when the variable is continuous). The probability distribution describes the range of possible values that a random variable can attain and the probability that the value of the random variable is within any (measurable) subset of that range.

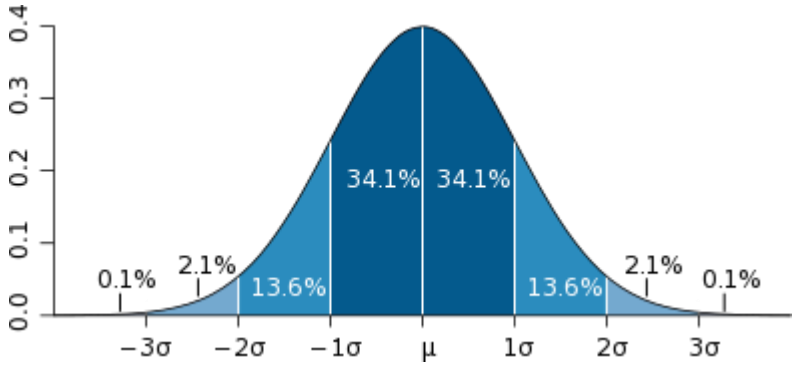


Figure 2: The Normal distribution, often called the "bell curve".

When the random variable takes values in the set of real numbers, the probability distribution is completely described by the *cumulative distribution function*, whose value at each real x is the probability that the random variable is smaller than or equal to x.

The concept of the probability distribution and the random variables which they describe underlies the mathematical discipline of probability theory, and the science of statistics. There is spread or variability in almost any value that can be measured in a population (e.g. height of people, durability of a metal, sales growth, traffic flow, etc.); almost all measurements are made with some intrinsic error; also in physics many processes are described probabilistically, from the kinetic properties of gases to the quantum mechanical description of fundamental particles. For these and many other reasons, simple numbers are often inadequate for describing a quantity, while probability distributions are often more appropriate.

3.8.1 Probability distributions of real-valued random variables

Because a probability distribution Pr on the real line is determined by the probability of a real-valued random variable X being in a half-open interval $(-\infty, x]$, the probability distribution is completely characterized by its cumulative distribution function given as:

$$F(x) = \Pr[X \leq x] \qquad \forall x \in \mathbb{R}.$$

a. Discrete probability distribution

A probability distribution is called ***discrete*** if its cumulative distribution function only increases in jumps. More precisely, a probability distribution is discrete if there is a finite or countable set whose probability is 1.
For many familiar discrete distributions, the set of possible values is discrete in the sense that all its points are isolated points. But, there are discrete distributions for which this countable set is dense on the real line.
Discrete distributions are characterized by a probability mass function, *p* such that:

$$\Pr[X = x] = p(x).$$

b. Continuous probability distribution

By one convention, a probability distribution μ is called ***continuous*** if its cumulative distribution function $F(x) = \mu(-\infty, x]$ is continuous and, therefore, the probability measure of singletons $\mu\{x\} = 0$.

Another convention reserves the term ***continuous probability distribution*** for absolutely continuous distributions. These distributions can be characterized by a probability density function: a non-negative Lebesgue integrable function *f* defined on the real numbers such that

$$F(x) = \mu(-\infty, x] = \int_{-\infty}^{x} f(t)\,dt.$$

Discrete distributions and some continuous distributions do not admit such a density.

Terminologies
The **support** of a distribution is the smallest closed interval/set whose complement has probability zero. It may be understood as the points or elements that are actual members of the distribution.

A **discrete random variable** is a random variable whose probability distribution is discrete. Similarly, a **continuous random variable** is a random variable whose probability distribution is continuous.

Some properties

- The probability density function of the sum of two independent random variables is the **convolution** of each of their density functions.
- The probability density function of the difference of two independent random variables is the **cross-correlation** of their density functions.
- Probability distributions are not a vector space – they are not closed under linear

combinations, as these do not preserve non-negativity or total integral 1 – but they are closed under convex combination, thus forming a convex subset of the space of functions (or measures).

In mathematics and, in particular, functional analysis, **convolution** is a mathematical operation on two functions *f* and *g*, producing a third function that is typically viewed as a modified version of one of the original functions. Convolution is similar to cross-correlation. It has applications that include statistics, computer vision, image and signal processing, electrical engineering, and differential equations

3.9 Summary of Common Probability Distributions

The following is a list of some of the most common probability distributions, grouped by the type of process that they are related to.

Note that all of the univariate distributions below are singly-peaked; that is, it is assumed that the values cluster around a single point. In practice, actually-observed quantities may cluster around multiple values. Such quantities can be modeled using a mixture distribution.

3.9.1 Related to real-valued quantities that grow linearly (e.g. errors, offsets)

- Normal distribution (aka Gaussian distribution), for a single such quantity; the most common continuous distribution;
- Multivariate normal distribution (aka multivariate Gaussian distribution), for vectors of correlated outcomes that are individually Gaussian-distributed;

3.9.2 Related to positive real-valued quantities that grow exponentially (e.g. prices, incomes, populations)

- Log-normal distribution, for a single such quantity whose log is normally distributed
- Pareto distribution, for a single such quantity whose log is exponentially distributed; the prototypical power law distribution.

3.9.3 Related to real-valued quantities that are assumed to be uniformly distributed over a (possibly unknown) region

- Discrete uniform distribution, for a finite set of values (e.g. the outcome of a fair die)
- Continuous uniform distribution, for continuously-distributed values.

3.9.4 Related to Bernoulli trials (yes/no events, with a given probability)

- *Bernoulli* distribution, for the outcome of a single Bernoulli trial (e.g. success/failure, yes/no);
- *Binomial* distribution, for the number of "positive occurrences" (e.g. successes, yes votes, etc.) given a fixed total number of independent occurrences;
- *Negative binomial* distribution, for binomial-type observations but where the quantity

of interest is the number of failures before a given number of successes occurs‘

- *Geometric* distribution, for binomial-type observations but where the quantity of interest is the number of failures before the first success; a special case of the negative binomial distribution.

3.9.5 Related to sampling schemes over a finite population

- *Binomial* distribution, for the number of "positive occurrences" (e.g. successes, yes votes, etc.) given a fixed number of total occurrences, using sampling with replacement
- *Hypergeometric* distribution, for the number of "positive occurrences" (e.g. successes, yes votes, etc.) given a fixed number of total occurrences, using sampling without replacement
- *Beta-binomial* distribution, for the number of "positive occurrences" (e.g. successes, yes votes, etc.) given a fixed number of total occurrences, sampling using a Polya urn scheme (in some sense, the "opposite" of sampling without replacement)

3.9.6 Related to categorical outcomes (events with *K* possible outcomes, with a given probability for each outcome)

- *Categorical* distribution, for a single categorical outcome (e.g. yes/no/maybe in a survey); a generalization of the Bernoulli distribution;
- *Multinomial* distribution, for the number of each type of categorical outcome, given a fixed number of total outcomes; a generalization of the binomial distribution;
- *Multivariate* hyper geometric distribution, similar to the multinomial distribution, but using sampling without replacement; a generalization of the hyper geometric distribution;

3.9.7 Related to events in a Poisson process (events that occur independently with a given rate)

- *Poisson* distribution, for the number of occurrences of a Poisson-type event in a given period of time
- *Exponential* distribution, for the time before the next Poisson-type event occurs

3.9.8 Useful for hypothesis testing related to normally-distributed outcomes

- *Chi-square* distribution, the distribution of a sum of squared standard normal variables; useful e.g. for inference regarding the sample variance of normally-distributed samples
- *Student's* t distribution, the distribution of the ratio of a standard normal variable and the square root of a scaled chi squared variable; useful for inference regarding the mean of normally-distributed samples with unknown variance
- *F-distribution*, the distribution of the ratio of two scaled chi squared variables; useful e.g. for inferences that involve comparing variances or involving R-squared (the squared correlation coefficient)

3.9.9 Useful as conjugate prior distributions in Bayesian inference

- *Beta distribution*, for a single probability (real number between 0 and 1); conjugate to the Bernoulli distribution and binomial distribution
- *Gamma distribution*, for a non-negative scaling parameter; conjugate to the rate parameter of a Poisson distribution or exponential distribution, the precision (inverse variance) of a normal distribution, etc.
- *Dirichlet distribution*, for a vector of probabilities that must sum to 1; conjugate to the categorical distribution and multinomial distribution; generalization of the beta distribution
- *Wishart distribution*, for a symmetric non-negative definite matrix; conjugate to the inverse of the covariance matrix of a multivariate normal distribution; generalzation of the gamma distribution

4.0 Self-Assessment Exercise(s)

Answer the following questions:

1. Define probability distribution
2. What is the relationship between a random variable and probability distribution
3. List the distributions related to:
    a. Bernoulli trials
    b. Categorical outcomes
    c. Hypothesis testing
4. The student is expected to familiarize him/herself with these probability distributions and their applications.

5.0 Conclusion

The basis of simulation is randomness. Here we have discussed this fundamental basis which offers us the possibility to quantitatively represent uncertainties in simulations. With **Probabilities** in simulation we can explicitly represent uncertainties by specifying inputs as probability distributions.

6.0 Summary

In this unit we discussed the following:

- Defined Probability as
- Discussed the fundamental concepts of probability theory
- The limit theorem
- Random variables and Random processes
- Probability distributions

- Provided a listing of common probability distributions grouped by their related processes

7.0 Further Readings

- Devore, J. L. (2018). *Probability and statistics for engineering and the sciences*. Toronto, Ontario: Nelson.
- Georgii, H. (2013). *Stochastics: Introduction to probability and statistics*. Berlin: De Gruyter.
- Giri, N. C. (2019). *Introduction to probability and statistics*. London: Routledge.
- Johnson, R. A., Miller, I., & Freund, J. E. (2019). *Miller & Freunds probability and statistics for engineers*. Boston: Pearson Education.
- Laha, R. G., & Rohatgi, V. K. (2020). *Probability theory*. Mineola, NY: Dover Publications.
- Mathai, A. M., & Haubold, H. J. (2018). *Probability and statistics: A course for physicists and engineers*. Boston: De Gruyter.
- Pishro-Nik, H. (2014). *Introduction to probability, statistics, and random processes*. Blue Bell, PA: Kappa Research, LLC.
- Spiegel, M. R., Schiller, J. J., & Srinivasan, R. A. (2013). *Schaums outline of probability and statistics*. New York: McGraw-Hill.

Module 2: MODELLING AND SIMULATION CONCEPTS

Module Introduction

This module is divided into four (4) units

Unit 1: Simulation and Modelling
Unit 2: Modelling Methods
Unit 3: Physics-Based Finite Element Model
Unit 4: Statistics for Modelling and Simulation

Unit 1: Simulation and Modelling

Contents

1.0 Introduction

Real world phenomenon are very dynamic thus difficult to exactly predict. To make decisions in such circumstances, we need a tool or verifiable procedures that guide decision makers to an informed and provable decision and action. In this unit we will look at one such tool; simulation which has become cornerstone to many probabilistic projects.

2.0 Intended Learning Outcomes (ILOs)

At the end of this unit you should be to:

- Say what simulation is about
- State why we need simulation
- Describe how simulations are done
- Describe various types of Simulations

- Give examples of Simulation
- Show areas of applications of Simulation

3.0 Main Content

3.1 What is Simulation?

The term **simulation** is used in different ways by different people. As used here, simulation is defined as the process of creating a *model* (i.e., an abstract representation or exact copy) of an existing or proposed *system* (e.g., a project, a business, a mine, a forest, the organs in your body, etc.) in order to identify and understand those factors which control the system and/or to predict (forecast) the future behaviour of the system. Almost any system which can be quantitatively described using equations and/or rules can be simulated.

The underlying purpose of simulation is to shed light on the underlying mechanisms that control the behaviour of a system. More practically, simulation can be used to predict (forecast) the future behaviour of a system, and determine what you can do to influence that future behaviour. That is, simulation can be used to predict the way in which the system will evolve and respond to its surroundings, so that you can identify any necessary changes that will help make the system perform the way that you want it to.

For example, a fisheries biologist could dynamically simulate the salmon population in a river in order to predict changes to the population, and quantitatively understand the impacts on the salmon of possible actions (e.g., fishing, loss of habitat) to ensure that they do not go extinct at some point in the future.
Also flight simulator on PC is also a computer model of some aspect of the flight; it shows on the screen the controls and what the pilot is supposed to see from the ―cockpit‖ (his armchair).

Simulation therefore, is a technique (not a method) for representing a dynamic real world system by a model and experimenting with the model in order to gain information about the system and hence take appropriate decision. Simulation can be done by hand or by a computer.

Simulation is a powerful and important tool because it provides a way in which alternative designs, plans and/or policies can be evaluated without having to experiment on a real system, which may be prohibitively costly, time-consuming, or simply impractical to do. That is, it allows you to ask *"What if?" questions* about a system without having to experiment on the actual system itself (and hence incur the costs of field tests, prototypes, etc.).

3.2 When to Use simulation

Simulation is used in systems that change with time, such as a gas station, where cars come

and go (called dynamic systems) and involve randomness. In such a system nobody can guess at exactly which time the next car should arrive at the station. Modelling complex dynamic systems theoretically need too many simplifications and the emerging models may not therefore be valid. Simulation does not require many simplifying assumptions, making it the only tool even in absence of randomness.

Simulation is used to observe the dynamic behaviour of a model of real or imaginary system. Indeed, by simulating a complex system we are able to understand the behaviour at low cost. Otherwise we would have to carry out a complicated theoretical research or to build a device (an electric heater, a building or a plane), and observe how it changes to get hints for improvements in the design.

If you run a shop, an hospital or a bank, then computer simulation may show you bottlenecks, service time, flows, and queues of clients and provide important information on how to improve your business.

Note that often we describe a real world system by:

1. A physical model
2. A mathematical or analytic model
3. An analogue model.

What happens when a system is not amenable to treatment using the above model? Constructing a real physical system could be very expensive and what more testing it with live human beings and observing what happens could be fatal. Training a new pilot using an airplane is suicidal. This is why simulation is designed and utilized.

Thus simulation is the answer to our question. Many operations Research analysts consider simulation to be a method of last resort. This is because it may be useful when other approaches cannot be used, for example when a real world situation is complex. Note that nothing prevents you from using simulation approach to analytic problem. Results can at least be compared!

Thus, before designing and implementing a real life system, it is necessary to find out via simulation studies whether the system will work otherwise the whole exercise will be a wild goose chase. Inevitably huge sums of money might have been wasted.

Unlike the situation in mathematical programming, so far there are no clear cut underlying principle guiding the formulation of simulation models. Each application is ad-hoc to a large extent. In general there are three basic objectives of simulation studies:

1. To Describe a Current System – Suppose that a manufacturing company has suddenly observed a marked deterioration in meeting due-dates of customers order. It may be necessary to build a simulation model to see how the current procedures for estimating due dates, scheduling production and ordering raw materials are giving rise to observed delays.
2. To explore a Hypothetical System – such as installing a new system, which will cost a lot of money, it might be better to build a hypothetical model of the system and learn from its behaviour.
3. To Design an Improved System – for example consider a supermarket that has one payment counter. Due to increase in patronage, it is considering to increase the

number of pay points. A simulation experiment may identify if one, two or more additional points are needed or not needed.

3.3 Types of Simulations

Computer models can be classified according to several criteria including:

- Stochastic or deterministic (and as a special case of deterministic, chaotic)
- Steady-state or dynamic
- Continuous or discrete (and as an important special case of discrete, discrete event or DE models)
- Local or distributed.

For example:

- Steady-state models use equations defining the relationships between elements of the modelled system and attempt to find a state in which the system is in equilibrium. Such models are often used in simulating physical systems, as a simpler modelling case before dynamic simulation is attempted.
- Dynamic simulations model changes in a system in response to (usually changing) input signals.
- *Stochastic* models use *random number generators* to model the chance or random events; they are also called Monte Carlo simulations.

There are two basic types of simulation for which models are built, and the process of choosing the subset of characteristics or features is different for each. The distinction between the two types is based on how time is represented; either as a **continuous** variable or as a **discrete** variable.

3.3.1 Continuous simulation

Continuous simulations treat time as continuous and express changes in terms of a set of differential equations that reflect the relationships among the set of characteristics. Thus the characteristics or features chosen to model the system must be those whose behaviour is understood mathematically.

Continuous simulation is used in systems where the state changes all the time, not just at the time of some discrete events. For example, the water level in a reservoir due to in and outflow changes all the time. In such cases ***continuous simulation*** is more appropriate, although discrete events simulation can serve as an approximation.

A meteorological modelling falls is another example in this category. The characteristics of weather models are wind components, temperature, water vapour, cloud formation, precipitation, and so on. The interaction of these components over time can be modelled by a set of partial differential equations, which measure the rate of change of each component over some three-dimensional region.

A *continuous dynamic simulation* performs numerical solution of differential-algebraic equations or differential equations (either partial or ordinary). Periodically, the simulation program solves all the equations, and uses the numbers to change the state and output of the

simulation. Applications include flight simulators, racing-car games, chemical process modelling, and simulations of electrical circuits. Originally, these kinds of simulations were actually implemented on analog computers, where the differential equations could be represented directly by various electrical components such as operational amplifiers. By the late 1980s, however, most "analogue" simulations were run on conventional digital computers that emulate the behaviour of an analog computer.
A typical Continuous (stochastic) system has a large number of control parameters that can have a significant impact on the performance of the system. To establish a basic knowledge of the behaviour of a system under variation of input parameters, sensitivity analysis is usually performed, which applies small changes from one state to the nominal values of input parameters. For such simulation, variations of the input parameter cannot be made infinitely small. The sensitivity of the performance measure with respect to an input parameter is therefore defined as (partial) derivative.
Sensitivity analysis is concerned with evaluating sensitivities (gradient) of performance measures with respect to parameter of interest. It provides guidance for design and operational decisions and plays a pivotal role in identifying the most significant system parameters, as well as bottleneck of subsystems.
In designing, analysing and operating such complex systems, one is interested not only in performance evaluation but also in sensitivity analysis and optimisation.

3.3.2 Discrete-event simulation,

Discrete event models are made up of *entitie*s, *attributes*, and *events*. An entity represents some object in the real system that must be explicitly defined. That is, the characteristic or feature of the system or an object. For example, if we were modelling a manufacturing plant, the different machines, and the product being created, would be entities. An attribute is some characteristic of a particular entity. The identification number, the purchase date, and the maintenance history would be attributes of a particular machine. An event is an interaction between entities. For example, the sending of the output from one machine as input to the next machine would be an event.

Suppose we are interested in a gas station. We may describe the behaviour of this system graphically by plotting the number of cars in the station; the state of the system. Every time a car arrives the graph increases by one unit while a departing car causes the graph to drop by one unit. This graph (called sample path), could be obtained from observation of real station, but could also be artificially constructed. Such *artificial construction and analysis of the resulting sample path (or more sample paths in more complex cases) consists of the simulation.*

The path consists of only horizontal and vertical lines, as cars arrivals and departures occurred, at distinct points in time, what we refer to as events. Between two consecutive events, nothing happens – the graph is horizontal. When the number of events are finite, we call the simulation ***discrete event***.

Discrete event systems (DES) are dynamic systems, which evolve in time by the occurrence

of events at possible irregular time intervals. DES abound in real-world applications. Examples include traffic systems, flexible manufacturing systems, computer communication systems, production lines, flow networks etc. Most of these systems can be modelled in terms of discrete events whose occurrence causes the system to change from one state to another. Simulations may be performed manually. Most often, however, the system model is written either as a computer program or as some kind of input into simulator software.

A *discrete event simulation* (DE) manages events in time. Most computer, logic-test and fault-tree simulations are of this type. In this type of simulation, the simulator maintains a queue of events sorted by the simulated time they should occur. The simulator reads the queue and triggers new events as each event is processed. It is not important to execute the simulation in real time. It's often more important to be able to access the data produced by the simulation, to discover logic defects in the design, or the sequence of events.

A special type of discrete simulation which does not rely on a model with an underlying equation, but can nonetheless be represented formally, is *agent-based simulation*. In agent-based simulation, the individual entities (such as molecules, cells, trees or consumers) in the model are represented directly (rather than by their density or concentration) and possess an internal *state* and set of behaviours or *rules* which determine how the agent's state is updated from one time-step to the next.

3.4 Steps In Constructing A Simulation Model.

1. Formulate the model (see modelling)
2. Design the Experiment – Workout details of experimental procedures before running the model subsystems, parameters, relationships, data structures, etc.
3. Develop the Computer Programs – Each historical evolution of the model, including generation of random events and generation of objects, will take place within the computer. if a model has a simple structure, you can use **BASIC, FORTRAN, PASCAL or C** and so on to develop the computerized version. However, it is better to use a simulation language such as **SIMULATIONSCRIPT, GPSS, SIMULATIONULA (SIMULA), SIMULATIONNET (SIMNET) II, QMS**, etc.

3.4.1 To Extract the terms in Simulation

Let us consider building a simulation of gas station with a single pump served by a single service man. Assumptions: arrival of cars as well as their times are random.
At first identify the:
State: number of cars waiting for service and number of cars served at any moment.
Event: arrival of cars, start of service, end of service.
Entities: these are the cars.
Queue: the queue of cars in front of the pump waiting for service.
Random realization: interval times, service times.
Distribution: we shall assume exponential distributions for the interval time and service time.

Next, specify what to do at each event. The above example would look like this:

At event of entity arrival: Create next arrival. If the server is free, send entity for start of service. Otherwise it joins the queue. At event of service start: Server becomes occupied. Schedule end of service for this entity. At event of service end: Server becomes free. If any entity is waiting in the queue: remove the first entity from the queue; send it for start of service.

Some initiation is still required, for example, the creation of the first arrival. Lastly, the above is translated into code. This is easy with appropriate library function, which has subroutine for creation, scheduling, proper timing of events, queue manipulations, random variate generation and statistics collection.

3.4.2 Simulation terminologies:

State – A variable characterizing an attribute in the system such as level of stock in inventory or number of jobs in waiting for processing.
Event: - An occurrence at a point in time which may change the state of the system, such as arrival of a customer or start of work on a job.
Entity: An object that passes through the system, such as cars in an intersection or orders in a factory. Often an event (e.g., arrival) is associated with an entity (e.g., customer).
Queue: A queue is not only a physical queue of people, or cars, etc it can also be a task list, a buffer of finished goods waiting for transportation or any place where entities are waiting for something to happen for any reason.
Creating: Is causing an arrival of new entity into the system at some point in time.
Scheduling: is the act of assigning a new future event to an existing entity. **Random variable**: is a quantity that is uncertain, such as interval time between two incoming flights or number of defectives parts in a shipment.
Random Variate: is an artificially generated random variable.
Distribution: is the mathematical law, which governs the probabilistic features of a random variable.

3.5 Applications of Computer Simulation

Computer simulation has become a useful part of modelling many natural systems in physics, chemistry and biology, and human systems in economics and social science (the computational sociology) as well as in engineering to gain insight into the operation of those systems. A good example of the usefulness of using computers to simulate can be found in the field of network traffic simulation. In such simulations the model behaviour will change each simulation according to the set of initial parameters assumed for the environment. Computer simulations are often considered to be *human out of the loop* simulations.

Computer graphics can be used to display the results of a computer simulation. Animations can be used to experience a simulation in real-time e.g. in training simulations. In some cases animations may also be useful in faster than real-time or even slower than real-time modes. For example, faster than real-time animations can be useful in visualizing the build up of queues in the simulation of humans evacuating a building.

There are many different types of computer simulation; the common feature they all share is the attempt to generate a sample of representative scenarios for a model in which a complete enumeration of all possible states of the model would be prohibitive or impossible. Several software packages also exist for running computer-based simulation modelling that makes the modelling almost effortless and simple.

a. Simulation in computer science

In computer science, simulation has an even more specialized meaning: Alan Turing uses the term "simulation" to refer to what happens when a digital computer runs a state transition table (runs a program) that describes the state transitions, inputs and outputs of a subject discrete-state machine. The computer simulates the subject machine.

In computer programming, a simulator is often used to execute a program that has to run on some inconvenient type of computer, or in a tightly controlled testing environment For example, simulators are usually used to debug a micro program or sometimes commercial application programs. Since the operation of the computer is simulated, all of the information about the computer's operation is directly available to the programmer, and the speed and execution of the simulation can be varied at will.

Simulators may also be used to interpret fault trees, or test very large scale integration (VLSI) logic designs before they are constructed. In theoretical computer science the term *simulation* represents a relation between state transition systems.

b. Simulation in training

Simulation is often used in the training of civilian and military personnel. This usually occurs when it is prohibitively expensive or simply too dangerous to allow trainees to use the real equipment in the real world. In such situations they will spend time learning valuable lessons in a "safe" virtual environment. Often the convenience is to permit mistakes during training for a safety-critical system.

Training simulations typically come in one of three categories:

- **live** simulation (where real people use simulated (or "dummy") equipment in the real world);
- **virtual** simulation (where real people use simulated equipment in a simulated world (or "virtual environment")), or
- **constructive** simulation (where simulated people use simulated equipment in a simulated environment). Constructive simulation is often referred to as "wargaming" since it bears some resemblance to table-top war games in which players command armies of soldiers and equipment which move around a board.

c. Simulation in Education

Simulations in education are somewhat like training simulations. They focus on specific tasks. In the past, video has been used for teachers and students to observe, problem solve and role play; however, a more recent use of simulation in education include animated

narrative vignettes (ANV). ANVs are cartoon-like video narratives of hypothetical and reality-based stories involving classroom teaching and learning. ANVs have been used to assess knowledge, problem solving skills and dispositions of children, and pre-service and in-service teachers.

Another form of simulation has been finding favour in business education in recent years. Business simulations that incorporate a dynamic model enable experimentation with business strategies in a risk free environment and provide a useful extension to case study discussions.

d. Medical Simulators

Medical simulators are increasingly being developed and deployed to teach therapeutic and diagnostic procedures as well as medical concepts and decision making to personnel in the health professions. Simulators have been developed for training procedures ranging from the basics such as blood draw, to laparoscopic surgery and trauma care.

Many medical simulators involve a computer connected to a plastic simulation of the relevant anatomy. Sophisticated simulators of this type employ a life size mannequin which responds to injected drugs and can be programmed to create simulations of life- threatening emergencies. In others simulations, visual components of the procedure are reproduced by computer graphics techniques, while touch-based components are reproduced by haptic feedback devices combined with physical simulation routines computed in response to the user's actions. Medical simulations of this sort will often use 3D CT or MRI scans of patient data to enhance realism.

Another important medical application of a simulator -- although, perhaps, denoting a slightly different meaning of *simulator* -- is the use of a *placebo* drug, a formulation which simulates the active drug in trials of drug efficacy.

e. City Simulators / Urban Simulation

A City Simulator is a tool used by urban planners to understand how cities are likely to evolve in response to various policy decisions. UrbanSim. The City Simulator developed at the University of Washington and ILUTE developed at the University of Toronto are examples of modern, large-scale urban simulators designed for use by urban planners. City simulators are generally agent-based simulations with explicit representations for land use and transportation.

f. Flight simulators

A flight simulator is used to train pilots on the ground. It permits a pilot to crash his simulated "aircraft" without being hurt. Flight simulators are often used to train pilots to operate aircraft in extremely hazardous situations, such as landings with no engines, or complete electrical or hydraulic failures. The most advanced simulators have high-fidelity visual systems and hydraulic motion systems. The simulator is normally cheaper to operate than a real trainer aircraft.

g. Marine simulators

This bears resemblance to flight simulators. The marine simulators are used to train ship personnel. Simulators like these are mostly used to simulate large or complex vessels, such as

cruise ships and dredging ships. They often consist of a replication of a ships' bridge, with operating desk(s), and a number of screens on which the virtual surroundings are projected.

Simulation in Engineering (Technology) Process

Simulation is an important feature in engineering systems or any system that involves many processes. For example in electrical engineering, delay lines may be used to simulate propagation delay and phase shift caused by an actual transmission line. Similarly, dummy loads may be used to simulate impedance without simulating propagation, and is used in situations where propagation is unwanted. A simulator may imitate only a few of the operations and functions of the unit it simulates. *Contrast with*: emulate.

Most engineering simulations entail mathematical modelling and computer assisted investigation. There are many cases, however, where mathematical modelling is not reliable. Simulation of fluid dynamics problems often requires both mathematical and physical simulations. In these cases the physical models require dynamic similitude. Physical and chemical simulations have also direct realistic uses, rather than research uses; in chemical engineering, for example, process simulations are used to give the process parameters immediately used for operating chemical plants, such as oil refineries.

Discrete Event Simulation is often used in industrial engineering, operations management and operational research to model many systems (commerce, health, defence, manufacturing, logistics, etc.); for example, the value-adding transformation processes in businesses, to optimize business performance. Imagine a business, where each person could do 30 tasks, where thousands of products or services involved dozens of tasks in a sequence, where customer demand varied seasonally and forecasting was inaccurate- this is the domain where such simulation helps with business decisions across all functions.

h. Simulation and games

Strategy games - both traditional and modern - may be viewed as simulations of abstracted decision-making for the purpose of training military and political leaders. In a narrower sense, many video games are also simulators, implemented inexpensively. These are sometimes called "sim games". Such games can simulate various aspects of reality, from economics to piloting vehicles, such as flight simulators (described above).

i. The "classroom of the future"

The "classroom of the future" will probably contain several kinds of simulators, in addition to textual and visual learning tools. This will allow students to enter school better prepared, and with a higher skill level. The advanced student or postgraduate will have a more concise and comprehensive method of retraining -- or of incorporating new academic contents into their skill set -- and regulatory bodies and institution managers will find it easier to assess the proficiency and competence of individuals.

In classrooms of the future, the simulator will be more than a "living" textbook; it will become an integral a part of the practice of Education and training. The simulator

environment will also provide a standard platform for curriculum development in educational institutions.

3.6 Model Evaluation

An important part of the modelling process is the evaluation of an acquired model. *How do we know if a mathematical model describes the system well?* This is not an easy question to answer. Usually the engineer has a set of measurements from the system which are used in creating the model. Then, if the model was built well, the model will adequately show the relations between system variables for the measurements at hand. The question then becomes: How do we know that the measurement data are a representative set of possible values? Does the model describe well the properties of the system between the measurement data (interpolation)? Does the model describe well events outside the measurement data (extrapolation)?

A common approach is to split the measured data into two parts; training data and verification data. The training data are used to *train* the model, that is, to estimate the model parameters (see above). The verification data are used to evaluate model performance. Assuming that the training data and verification data are not the same, we can assume that if the model describes the verification data well, then the model describes the real system well.

However, this still leaves the *extrapolation question* open. How well does this model describe events outside the measured data? Consider again Newtonian classical mechanics-model. Newton made his measurements without advanced equipment, so he could not measure properties of particles travelling at speeds close to the speed of light. Likewise, he did not measure the movements of molecules and other small particles, but macro particles only. It is then not surprising that his model does not extrapolate well into these domains, even though his model is quite sufficient for ordinary life physics.

The reliability and the trust people put in computer simulations depends on the validity of the simulation model, therefore verification and validation are of crucial importance in the development of computer simulations. Another important aspect of computer simulations is that of reproducibility of the results, meaning that a simulation model should not provide a different answer for each execution. Although this might seem obvious, this is a special point of attention in stochastic simulations, where random numbers should actually be semi-random numbers. An exception to reproducibility are human in the loop simulations such as flight simulations and computer games. Here a human is part of the simulation and thus influences the outcome in a way that is hard if not impossible to reproduce exactly.

Case Studies

Operations study to add a new plane arrival at La Guardia southwest terminal

LaGuardia airport planned to add a new flight to the schedule of the southwest terminal. The airport administration wanted to understand how the introduction of a new flight would

influence terminal capacity.

Problem

In order to understand the scale of the problem, the developers conducted a preliminary static pedestrian flow analysis based on data of how long before the flight passengers arrived at the airport. In the picture, the solid line represented the number of seats in the waiting area, the red stacks represented the number of passengers in the terminal before introducing the new flight, and additional passengers from the new flight were represented by purple areas. The graph showed that if the new plane took off in the afternoon at 5:00 pm, the already crowded waiting area would have to bear an additional burden that could lead to a significant problem.

The developers used the AnyLogic Pedestrian Library to create a crowd simulation model of the terminal in order to examine the use of seats under different scenarios. The basic model displayed the operation of all terminal areas before the introduction of the new flight, and then various assumptions could be checked against this model. The best situation was when people were waiting for departure at their gates, but the consultants wanted to check how far they would have to move away from their gates to wait for their departure.

To set up the crowd simulation model, the developers used tables of passenger preference for waiting areas.

The model showed how far from their gate people would have to wait. The results of modeling the base scenario, without the new flight, showed that some of the peaks were reduced compared with the static analysis. This was due to passengers lining up 30 minutes before their flights. The model also showed where the people would actually wait. From this, it could be verified that there was no overflow and that the situation was stable.

In the afternoon, the waiting area was a lot more heavily utilized. There were a lot of passengers mixing in different areas and waiting for different gates. With the new flight at this peak time, some of these areas would get extremely overloaded. This pedestrian simulation was very useful in showing the operations of this terminal and how adding the new flight would affect the passengers in this area, including how far they would have to move to wait for their flights.

Solution

Designing large transport facilities requires careful consideration and agreement on every detail. That means that such projects must go through a great deal of decision making. The initial task of engineers usually produces alternatives and functional designs. These consider physical requirements and standards, but whether business or operating objectives will be met can be hard to determine accurately. It is here that AnyLogic based modeling helps by enabling faster decision-making and significantly improving insight into the various tasks that engineers face when planning large transport facilities.

4.0 Self-Assessment Exercise(s)

Answer the following questions:

- What are the objectives of simulation?
- In one sentence for each distinguish between different types of simulation

- Briefly describe simulation in five application areas.

5.0 Conclusion

Simulation is used to shed light on the underlying mechanisms that control the behaviour of a system. It can be used to predict (forecast) the future behaviour of a system, and determine what you can do to influence that future behaviour. We simulation when we require information to solve bottlenecks, service time, flows, and queues of clients and provide important information on how to improve your business.

We simulate when a system is not amenable to treatment using any of the physical model, mathematical or analogue models. Other reasons to resort to simulation include when it is very expensive to construct a real physical system and what more testing it with live human beings and observing what happens could be fatal. Training a new pilot using an airplane is suicidal. These are where and when simulations are designed and utilized.

6.0 Summary

In this unit we:

- defined simulation as the process of creating a *model* (i.e., an abstract representation or exact copy) of an existing or proposed *system* (e.g., a project, a business, a mine, a forest, the organs in your body, etc.) in order to identify and understand those factors which control the system and/or to predict (forecast) the future behaviour of the system.
- Stated that simulation is required when a system is not amenable to treatment using any existing model or when it is very expensive to construct a real physical system or when testing it with live human beings could be fatal.
- classified simulation into:
    - Stochastic or deterministic (and as a special case of deterministic, chaotic)
    - Steady-state or dynamic
    - Continuous or discrete (and as an important special case of discrete, discrete event or DE models)
    - Local or distributed
- Stated that simulations are done by: Formulating the model, Design the Experiment and Developing the Computer Programs.
- Listed areas of applications of Simulation to include: Computer science, Medicine, Education, City/Urban planning, Training etc.

7.0 Further Readings

- Gordon, S. I., & Guilfoos, B. (2017). *Introduction to Modeling and Simulation with*

MATLAB® and Python. Milton: CRC Press.

- Zeigler, B. P., Muzy, A., & Kofman, E. (2019). *Theory of modeling and simulation: Discrete event and iterative system computational foundations*. San Diego (Calif.): Academic Press.
- Kluever, C. A. (2020). *Dynamic systems modeling, simulation, and control*. Hoboken, N.J: John Wiley & Sons.
- Law, A. M. (2015). *Simulation modeling and analysis*. New York: McGraw-Hill.
- Verschuuren, G. M., & Travise, S. (2016). *100 Excel Simulations: Using Excel to Model Risk, Investments, Genetics, Growth, Gambling and Monte Carlo Analysis*. Holy Macro! Books.
- Grigoryev, I. (2015). *AnyLogic 6 in three days: A quick course in simulation modeling*. Hampton, NJ: AnyLogic North America.
- Dimotikalis, I., Skiadas, C., & Skiadas, C. H. (2011). *Chaos theory: Modeling, simulation and applications: Selected papers from the 3rd Cghaotic Modeling and Simulation International Conference (CHAOS2010), Chania, Crete, Greece, 1-4 June, 2010*. Singapore: World Scientific.
- Velten, K. (2010). *Mathematical modeling and simulation: Introduction for scientists and engineers*. Weinheim: Wiley-VCH.

Unit 2: Modelling Methods

Contents

1.0 Introduction

Modelling is an essential and inseparable part of all scientific activity, and many scientific disciplines have their own ideas about specific types of modelling. There is little general theory about scientific modelling, offered by the philosophy of science, systems theory, and new fields like knowledge visualization.

We create **models** for representation of the objects within a system together with the rules that govern the interactions of the objects. The representation may be concrete as in the case of the spaceship or flight simulators or abstract as in the case of the computer program that examines the number of checkout stations in service queue.

2.0 Intended Learning Outcomes (ILOs)

At the end of this unit, the student should be able to:

- Define Modelling
- Describe some basic modelling concepts
- Differentiate between Visual and Conceptual models
- Explain the Characteristics of Visual, models

3.0 Main Content

Definitions of Modelling

Modelling is the process of generating abstract, conceptual, graphical and/or mathematical models. Science offers a growing collection of methods, techniques and theory about all kinds of specialized scientific modelling. A scientific model can provide a way to read elements easily which have been broken down to a simpler form.

Model

A scientific model seeks to represent empirical objects, phenomena, and physical processes in a logical and objective way. All models are simplified reflections of reality, but despite their inherent falsity, they are nevertheless extremely useful. Building and disputing models is fundamental to the scientific enterprise. Complete and true representation may be impossible but scientific debate often concerns which is the better model for a given task, such as the most accurate climate model for seasonal forecasting.

For the scientist, a **model** is also a way in which the human thought processes can be amplified. For instance, models that are rendered in software allow scientists to leverage computational power to simulate, visualize, manipulate and gain intuition about the entity, phenomenon or process being represented.

3.1 Basic Modelling Concepts

Modelling as a substitute for direct measurement and experimentation

Models are typically used when it is either impossible or impractical to create experimental conditions in which scientists can directly measure outcomes. Direct measurement of outcomes under controlled conditions will always be more accurate than modelled estimates of outcomes. When predicting outcomes, models use *assumptions*, while measurements do not. As the number of assumptions in a model increases, the accuracy and relevance of the model diminishes.

Modelling language

A *modelling language* is any *artificial language* that can be used to express information or knowledge or systems in a structure that is defined by a consistent set of rules. The rules are used for interpretation of the meaning of components in the structure.

Simulation

A *simulation* is the implementation of a model. A steady state simulation provides information about the system at an instant in time (usually at equilibrium, if it exists). A dynamic simulation provides information over time. A simulation brings a model to life and shows how a particular object or phenomenon will behave. It is useful for testing, analysis or training where real-world systems or concepts can be represented by a model.

Structure

Structure is a fundamental and sometimes intangible notion covering the recognition, observation, nature, and stability of patterns and relationships of entities. From a child's verbal description of a *snow*, to the detailed *scientific analysis* of the properties of magnetic fields, the concept of structure is an essential foundation of nearly every mode of inquiry and discovery in science, philosophy, and art.

Systems

A *system* is a set of interacting or interdependent entities, real or abstract, forming an integrated whole. In general, a system is a construct or collection of different elements that together can produce results not obtainable by the elements alone. The concept of an 'integrated whole' can also be stated in terms of a system embodying a set of relationships which are differentiated from relationships of the set to other elements, and from relationships between an element of the set and elements not a part of the relational regime.

There are two types of systems:

1) Discrete, in which the variables change instantaneously at separate points in time and,
2) Continuous, where the state variables change continuously with respect to time.

3.1.1 The process of generating a model

Modelling refers to the process of generating a model as a conceptual representation of some phenomenon. Typically a model will refer only to some aspects of the phenomenon in question, and two models of the same phenomenon may be essentially different, that is in which the difference is more than just a simple renaming. This may be due to differing requirements of the model's end users or to conceptual or aesthetic differences by the modellers and decisions made during the modelling process. *Aesthetic* considerations that may influence the *structure* of a model might be the modeller's preferences regarding probabilistic models vis-a-vis deterministic ones, discrete vs continuous time etc. For this reason users of a model need to understand the model's original purpose and the assumptions of its validity.

3.1.2 Factors in evaluating a model

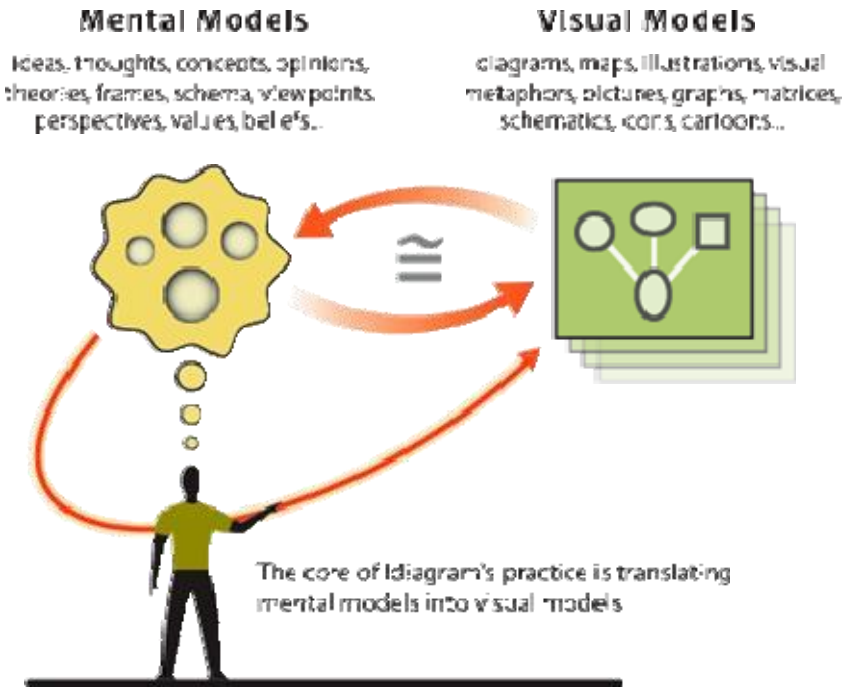
A model is evaluated first and foremost by its consistency to empirical data; any model inconsistent with reproducible observations must be modified or rejected. However, a fit to empirical data alone is not sufficient for a model to be accepted as valid. Other factors important in evaluating a model include:

- Ability to explain past observations
- Ability to predict future observations
- Cost of use, especially in combination with other models
- Refutability, enabling estimation of the degree of confidence in the model
- Simplicity, or even aesthetic appeal

3.2 Visual and Conceptual models

Visualization is any technique for creating images, diagrams, or animations to communicate a message. Visualization through visual imagery has been an effective way to communicate both abstract and concrete ideas since the dawn of man. Examples from history include cave paintings, Egyptian hieroglyphs, Greek geometry, and Leonardo da Vinci's revolutionary methods of technical drawing for engineering and scientific purposes. We should not hold a narrow definition of exactly what a **visual model** should *look like*. We should rather use whatever visual elements or styles such as diagrams, maps, graphs, charts, pictures, cartoons, etc. – that will most effectively represent the problem at hand.

We can however define visual models by what they strive to do, and list some of the important characteristics that distinguish 'visual models' from other kinds of graphic art.



Visual representation of data depends fundamentally on an appropriate **visual** scheme for mapping numbers into graphic patterns (Berlin 1983). One reason for the widespread use of graphical methods for quantitative data is the availability of a natural **visual** mapping: magnitude can be represented by length, as in a bar chart, or by position along a scale, as in dot charts and scatter plots. One reason for the relative paucity of graphical methods for categorical data may be that a natural **visual** mapping for frequency data is not so apparent.

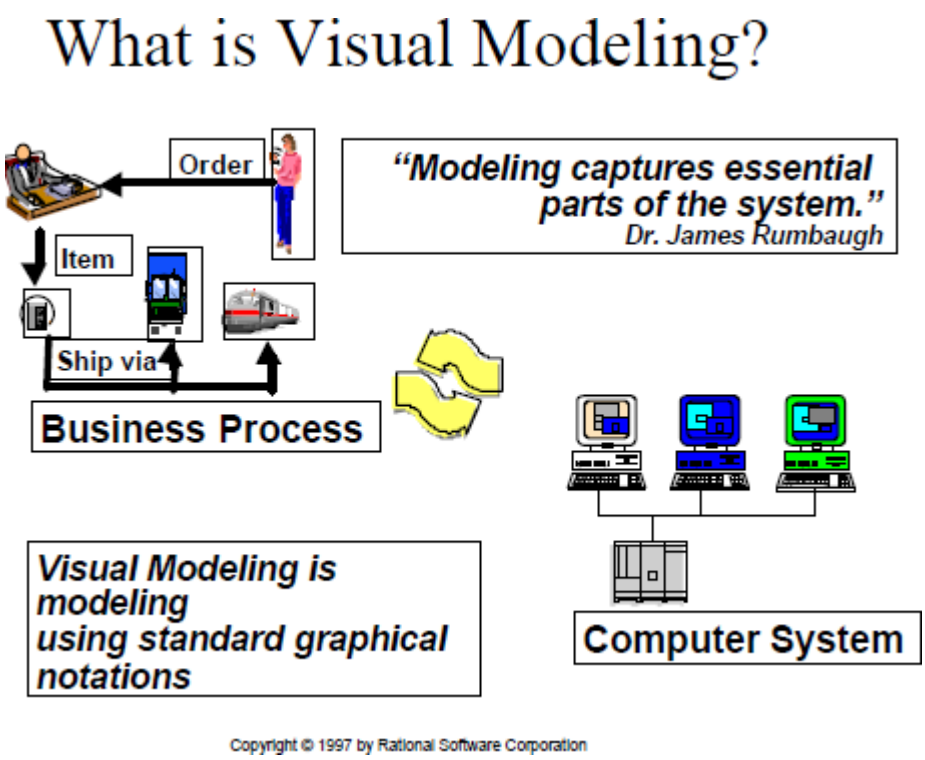
Conceptual model helps you interpret what is shown in a drawing or graph. A good **conceptual** model for a graphical display will have deeper connections with underlying statistical ideas as well.

For quantitative data, position along a scale can be related to mechanical models in which fitting data by least squares or least absolute deviations correspond directly to balancing forces or minimizing potential energy (Farebrother 1987).

The mechanical model for least squares regression, for example, likens each observation to a unit mass connected vertically to a rod by springs of unit modulus. Sall (1991a) shows how this mechanical model neatly describes the effects of sample size on power of a test, the leverage of outlying observations in regression, principal components, and collinearity among others.

Conceptual Modelling

- Is used for abstract (visual) representation of the problem domain
- It serves to enhance understanding of complex problem domains.
- It provides a basis for communication among project team members



3.3 Features of Visual and Conceptual Model

A visual model should:

- Render **conceptual knowledge** as opposed to quantitative data (information visualization) or physical things (technical illustration). We usually express conceptual knowledge with words alone, and yet the meaning behind those words is often *inherently visual*. Visual models seek to render directly the image-schematic (meaning that lies behind our words).
- Be **good models** - the images should accurately reflect the situation in the world and embody the characteristics of a useful model.
- **Integrate** the most salient aspects of the problem into a clear and coherent picture.
- **Fit the visual structure to the problem** – and not force the problem into a predefined visual structure.
- Use a **consistent visual grammar**.
- Should be **visually and cognitively *tractable***. Visual models exist to support robust qualitative thinking: they're software for 'human-simulation' (as opposed to computer-simulation) of the issue at hand. To serve as effective 'simulation software', visual models must be 'readable' and 'run able' by our visio-cognitive 'hardware' and should positively engage our prodigious visual intelligence.
- Tap into the power of **elegant design**. In other words, they shouldn't be ugly

Conceptual Modelling

- A good conceptual model should NOT reflect a solution bias.
- Should model the problem domain, not the solution domain.
- Initial conceptual model may be rough and general.
- May be refined during incremental development.

3.4 Cognitive Affordances of Visual Models

Due to the limited capacity of our working memory, 7 ± 2 _chunks' of information, we cannot hold in our minds concepts, arguments, or problems that consist of more than 5 to 9 objects or relationships. While this cognitive limitation severely restricts our ability to think about complex things, we can do what we often do: extend our intellectual abilities with external representations or 'models' of the problem.

The particular affordances diagrams – their ability to simultaneously show many objects and relationships – make them an ideal tool for thinking about conceptually-complex problems. Diagrams provide an external mnemonic aid that enables us to see complicated relationships and easily move between various mind-sized groupings of things.

4.0 Self-Assessment Exercise(s)

Answer the following questions:

1. Differentiate between modelling and a model
2. What factors are important in evaluating a model?
3. What are the desirable features of a visual model?

5.0 Conclusion

The essence of constructing a model is to identify a small subset of characteristics or features that are sufficient to describe the behaviour of the system under investigation. Since a model is an abstraction of a real system and not the system itself, there is therefore, a fine line between having too few characteristics to accurately describe the behaviour of the system and more than you need to accurately describe the system. The goal should be to build the simplest model that effectively describes the relevant behaviour of the system.

6.0 Summary

- We defined **modelling** as the process of generating abstract, conceptual, graphical and/or mathematical models. Science offers a growing collection of methods, techniques and theory about all kinds of specialized scientific modelling.
- We Listed and briefly explained some basic modelling concepts
- Differentiating between Visual and Conceptual models
- we discussed the important factors in evaluating a model to include:
    - Ability to explain past observations
    - Ability to predict future observations
    - Cost of use, especially in combination with other models
    - Refutability, enabling estimation of the degree of confidence in the model

    - Simplicity, or even aesthetic appeal
- We discussed the features of a good visual model which include:
    - Ability to render conceptual knowledge as opposed to quantitative data (information visualization) or physical things (technical illustration),
    - the images should accurately reflect the situation in the world,
    - the model should Integrate the most salient aspects of the problem into a clear and coherent picture,
    - Fit the visual structure to the problem,
    - It should Use a consistent visual grammar,
    - Should be visually and cognitively tractable.
- We also stated the Characteristics of Conceptual models

7.0 Further Readings

- Gordon, S. I., & Guilfoos, B. (2017). *Introduction to Modeling and Simulation with MATLAB® and Python*. Milton: CRC Press.
- Zeigler, B. P., Muzy, A., & Kofman, E. (2019). *Theory of modeling and simulation: Discrete event and iterative system computational foundations*. San Diego (Calif.): Academic Press.
- Kluever, C. A. (2020). *Dynamic systems modeling, simulation, and control*. Hoboken, N.J: John Wiley & Sons.
- Law, A. M. (2015). *Simulation modeling and analysis*. New York: McGraw-Hill.
- Verschuuren, G. M., & Travise, S. (2016). *100 Excel Simulations: Using Excel to Model Risk, Investments, Genetics, Growth, Gambling and Monte Carlo Analysis*. Holy Macro! Books.
- Grigoryev, I. (2015). *AnyLogic 6 in three days: A quick course in simulation modeling*. Hampton, NJ: AnyLogic North America.
- Dimotikalis, I., Skiadas, C., & Skiadas, C. H. (2011). *Chaos theory: Modeling, simulation and applications: Selected papers from the 3rd Cghaotic Modeling and Simulation International Conference (CHAOS2010), Chania, Crete, Greece, 1-4 June, 2010*. Singapore: World Scientific.
- Velten, K. (2010). *Mathematical modeling and simulation: Introduction for scientists and engineers*. Weinheim: Wiley-VCH.

Unit 3: Finite Element Model and Database Model

Contents

1.0 Introduction

The finite element method is one of the most powerful approaches for approximate solutions to a wide range of problems in mathematical physics. The method has achieved acceptance in nearly every branch of engineering and is the preferred approach in structural mechanics and heat transfer. Its application has extended to soil mechanics, heat transfer, fluid flow, magnetic field calculations, and other areas.

Managing large quantities of **structured** and **unstructured** data is a primary function of **information systems**. Data models describe structure of **data** for **storage** in data management systems such as relational databases. They typically do not describe unstructured data, such as documents, word processing, email messages, pictures, digital audio, and video

2.0 Intended Learning Outcomes (ILOs)

After studying this unit the reader should be able to:

- Define Finite Element Method (FEM)
- Describe the relationship between FEM and Finite element analysis
- State the origin and Applications of FEM
- Describe the Basics of FEM
- Define Data modeling

- Describe the different types and the three perspectives of data models
- Have an overview of database models

3.0 Main Content

3.1 Finite Element Model

Many physical phenomena in engineering and science can be described in terms of partial differential equations (PDE). In general, solving these equations by classical analytical methods for arbitrary shapes is almost impossible. The finite element method (FEM) is a numerical approach by which these PDE can be solved approximately.
The FEM is a function/basis-based approach to solve PDE. FEs are widely used in diverse fields to solve static and dynamic problems − Solid or fluid mechanics, electromagnetic, biomechanics, etc.

The **finite element method (FEM)** (its practical application often known as **finite element analysis** (FEA)) is a numerical technique for finding approximate solutions of partial differential equations (PDE) as well as of integral equations. The solution approach is based either on eliminating the differential equation completely (steady state problems), or rendering the PDE into an approximating system of ordinary differential equations, which are then numerically integrated using standard techniques such as Euler's method, Runge-Kutta, etc.

In solving partial differential equations, the primary challenge is to create an equation that approximates the equation to be studied, but is numerically stable, meaning that errors in the input and intermediate calculations do not accumulate and cause the resulting output to be meaningless. There are many ways of doing this, all with advantages and disadvantages. The steps may be broken down as follows:

1. Definition of the physical problem:- development of the model.
2. Formulation of the governing equations - Systems of PDE, ODE, algebraic equations, define initial conditions and/or boundary conditions to get a well-posed problem,
3. Discretization of the equations.
4. Solution of the discrete system of equations.
5. Interpretation of the obtained results.
6. Errors analysis.

The Finite Element Method is a good choice for solving partial differential equations over complicated domains (like cars and oil pipelines), when the domain changes (as during a solid state reaction with a moving boundary), when the desired precision varies over the entire domain, or when the solution lacks smoothness. For instance, in a frontal crash simulation it is possible to increase prediction accuracy in "important" areas like the front of the car and reduce it in its rear (thus reducing cost of the simulation); another example would be the simulation of the weather pattern on Earth, where it is more important to have accurate predictions over land than over the wide-open sea.

3.1.1 The Finite Element Analysis

Finite Element Analysis is a method to computationally model reality in a mathematical form to better understand a highly complex problem. In the real world *everything* that occurs is as a results of interactions between atoms (and sub-particles of those atoms), billions and billions of them. If we were to simulate the world in a computer, we would have to simulate this interaction based on the simple laws of physics. However, no computer can process the near infinite number of atoms in objects, so instead we model 'finite' groups of them.

For example, we might model a gallon of water by dividing it up into 1000 parts and measuring the interaction of these linked parts. If you divide into too few parts, your simulation will be too inaccurate. If you divide into too many, your computer will sit there for years calculating the result!

3.1.2 Why use FEA?

Simulation in general is always a good idea, as it lets you test designs and ideas without spending money or effort actually building anything. By using simulation, you can find fault points within your designs, simulate ideas as you think of them, and even quantitize and optimize them. One can even use simulation to verify theories - if the theoretical simulation matches what actually happens, then the theory is proven!

Sometimes you can hand calculate certain designs. But sometimes a design can be too complex, making FEA great for non-symmetric problems with ultra-complicated geometries.

3.1.3 History of FEM

The finite element method originated from the need for solving complex elasticity and structural analysis problems in civil and aeronautical engineering. The development can be traced back to the work by Alexander Hrennikoff (1941) and Richard Courant (1942). While the approaches used by these pioneers are dramatically different, they share one essential characteristic: mesh discretization of a continuous domain into a set of discrete sub-domains, usually called elements.

Hrennikoff's work discretizes the domain by using a lattice analogy while Courant's approach divides the domain into finite triangular subregions for solution of second order elliptic partial differential equations (PDEs) that arise from the problem of torsion of a cylinder. Courant's contribution was evolutionary, drawing on a large body of earlier results for PDEs developed by Rayleigh, Ritz, and Galerkin.

Development of the finite element method began in earnest in the middle to late 1950s for airframe and structural analysis and gathered momentum at the University of Stuttgart through the work of John Argyris and at Berkeley through the work of Ray W. Clough in the 1960s for use in civil engineering. By late 1950s, the key concepts of stiffness matrix and element assembly existed essentially in the form used today. NASA issued a request

for proposals for the development of the finite element software NASTRAN in 1965. The method was again provided with a rigorous mathematical foundation in 1973 with the publication of Strang and Fix's *An Analysis of The Finite Element Method* has since been generalized into a branch of applied mathematics for numerical modelling of physical systems in a wide variety of engineering disciplines, e.g., electromagnetism, thanks to Peter P. Silvester and fluid dynamics.

3.1.4 Applications of FEM

A variety of specializations under the umbrella of the mechanical engineering discipline (such as aeronautical, biomechanical, and automotive industries) commonly use integrated FEM in design and development of their products. Several modern FEM packages include specific components such as thermal, electromagnetic, fluid, and structural working environments. In a structural simulation, FEM helps tremendously in producing stiffness and strength visualizations and also in minimizing weight, materials, and costs.

3.2 Overview of Basic FEM

The basic steps of the finite element method are discussed next in more generality. Although attention is focused on structural problems, most of the steps translate to other applications problems as noted above. The role of FEM in numerical simulation is schematized in figure 1 below.

This diagram displays the three key simulation steps: idealization, discretization and solution. It also indicates the fact that each step introduces different of errors. For example the discretization error is the discrepancy obtained when the discrete solution is substituted in the mathematical model.

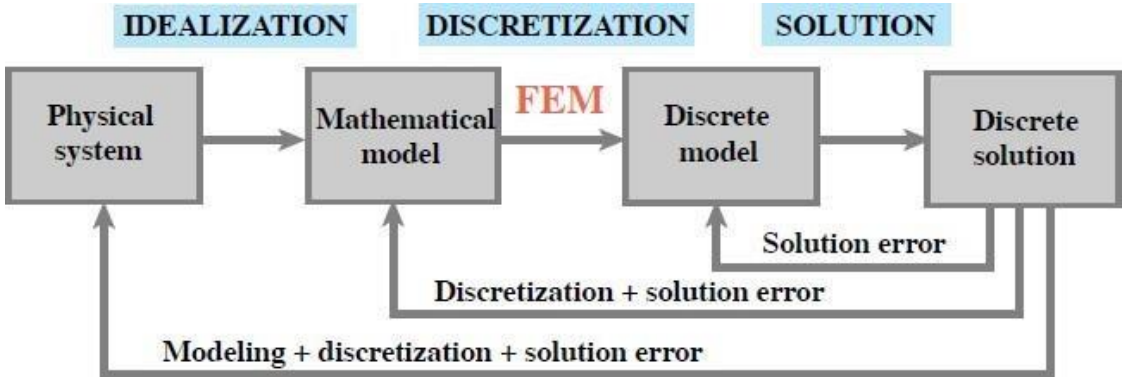


Figure1: Steps of the physical simulation process: idealization, discretization and solution.

3.2.1 Idealization

a. Models

The word ―model‖ has the traditional meaning of a scaled copy or representation of an object. And that is precisely how most dictionaries define it. We use here the term in a more modern sense, increasingly common since the advent of computers:

A model is a symbolic device built to simulate and predict aspects of behaviour of a system.

Note the careful distinction made between ―behaviour‖ and ―aspects of behaviour.‖ To predict everything, in all physical scales, you must deal with the actual system. A model

abstracts aspects of interest to the modeler. The term ―symbolic‖ means that a model represents a system in terms of the symbols and language of another science. For example, engineering systems may be (and are) modeled with the symbols of mathematics and/or computer sciences.

b. Mathematical Models

Mathematical modelling, or *idealization*, is a process by which the engineer passes from the actual physical system under study, to a *mathematical model* of the system, where the term *model* is understood in the wider sense defined above.

The process is called *idealization* because the mathematical model is necessarily an abstraction of the physical reality. (Note the phrase *aspects of behaviour* in the definition.) The analytical or numerical results obtained for the mathematical model are re-interpreted in physical terms only for those aspects.

Why is the mathematical model an abstraction of reality?

Engineering systems such as structures tend to be highly complex. To simulate its behaviour it is necessary to reduce that complexity to manageable proportions. Mathematical modelling is an abstraction tool by which complexity can be brought under control. This is achieved by ―filtering out‖ physical details that are not relevant to the analysis process. For example, a continuum material model necessarily filters out the aggregate, crystal, molecular and atomic levels of matter. If you are designing a bridge or building such levels are irrelevant. Consequently, choosing a mathematical model is equivalent to choosing an information filter.

3.2.2 Implicit vs. Explicit Modelling

Suppose that you have to analyze a structure and at your disposal is a ―black box‖ general-purpose finite element program. This is also known in the trade as a ―canned program.‖ Those programs usually offer a *catalog* of element types; for example, bars, beams, plates, shells, axisymmetric solids, general 3D solids, and so on. The moment you choose specific elements from the catalog you automatically accept the mathematical models on which the elements are based. This is *implicit modelling*, a process depicted in Figure 2.

Ideally you should be fully aware of the implications of your choice. Providing such ―finite element literacy‖ is one of the objectives of this course.

Unfortunately many users of commercial programs are unaware of the ―implied consent‖ aspect of implicit modelling.

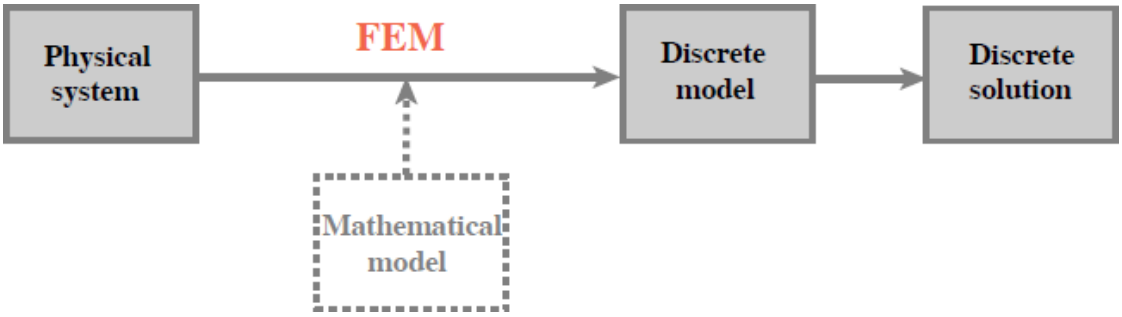


Figure 2: Implicit modelling: picking elements from an existing FEM code implicitly accepts an idealization. Read the fine print.

The other extreme occur when you select a mathematical model of the physical problem with your eyes wide open and *then* either shop around for finite element programs that implements that model, or write the program yourself. This is *explicit modelling*. It requires far more technical expertise, resources, experience and maturity than implicit modelling. But for problems that fall out of the ordinary it may be the right thing to do.

In practice a combination of implicit and explicit modelling is quite common. The physical problem to be solved is broken down into subproblems. Those subproblems that are conventional and fit existing programs may be treated with implicit modelling. Those subproblems that require special handling may yield only to explicit modelling treatment.

3.3 Discretization

3.3.1 Purpose

Mathematical modelling is a simplifying step. But models of physical systems are not necessarily easy to solve. They usually involve coupled partial differential equations in space and time subject to boundary and/or interface conditions. Such analytical models have an *infinite* number of degrees of freedom.

At this point one faces the choice of trying for analytical or numerical solutions. Analytical solutions, also called ―closed form solutions,‖ are more intellectually satisfying, particularly if they apply to a wide class of problems. Unfortunately they tend to be restricted to regular geometries and simple boundary conditions. Moreover a closed- form solution, expressed for example as the inverse of an integral transform, often has to be numerically evaluated to be useful.
Most problems faced by the engineer either do not yield to analytical treatment or doing so would require a disproportionate amount of effort. The practical way out is numerical simulation. Here is where finite element methods and the digital computer enter thescene.

To make numerical simulations practical it is necessary to reduce the number of degrees of freedom to a *finite* number. The reduction is called *discretization*. The end result of the discretization process is the *discrete model* depicted in Figures 1 and 2.

Discretization can proceed in space dimensions as well as in the time dimension. Because the present course deals only with static problems, we need not consider the time dimension and are free to concentrate on *spatial discretization*.

3.3.2 Error Sources and Approximation

Figure 1 tries to convey graphically that each simulation step introduces a source of error. In engineering practice modelling errors are by far the most important. But they are difficult and expensive to evaluate, because such *model validation* requires access to and comparison with experimental results.

Next in order of importance is the *discretization error*. Even if solution errors are ignored and usually they can, the computed solution of the discrete model is in general only an approximation in some sense to the exact solution of the mathematical model. A quantitative measurement of this discrepancy is called the *discretization error*.
The characterization and study of this error is addressed by a branch of numerical mathematics called approximation theory.

Intuitively one might suspect that the accuracy of the discrete model solution would improve as the number of degrees of freedom is increased, and that the discretization error goes to zero as that number goes to infinity. This loosely worded statement describes the *convergence* requirement of discrete approximations. One of the key goals of approximation theory is to make the statement as precise as it can be expected from a branch of mathematics.

3.3.3 Finite and Boundary Element Methods

The most popular discretization procedures in structural mechanics are finite element methods and boundary element methods. The finite element method (FEM) is by far the most widely used. The boundary element method (BEM) has gained in popularity for special types of problems, particularly those involving infinite domains, but remains a distant second.
In non-structural application areas such as fluid mechanics and thermal analysis, the finite element method is gradually making up ground but faces stiff competition from both the classical and energy-based *finite difference* methods. Finite difference and finite volume methods are particularly well entrenched in computational fluid dynamics.

3.4 Interpretation of FEM

The finite element method (FEM) is the dominant discretization technique in structural mechanics. FEM can be interpreted from either a physical or mathematical standpoint. The treatment has so far emphasized the former.

The basic concept in the physical interpretation of the FEM is the subdivision of the mathematical model into disjoint (non-overlapping) components of simple geometry called *finite elements* or *elements* for short. The response of each element is expressed in terms of a finite number of degrees of freedom characterized as the value of an unknown function,

or functions, at a set of nodal points.

The response of the mathematical model is then considered to be approximated by that of the discrete model obtained by connecting or assembling the collection of all elements.

The disconnection-assembly concept occurs naturally when examining many artificial and natural systems. For example, it is easy to visualize an engine, bridge, building, airplane, or skeleton as fabricated from simpler components.
Unlike finite difference models, finite elements *do not overlap* in space. In the mathematical interpretation of the FEM, this property goes by the name *disjoint support*.

FEM Element Attributes
Just like the members in the truss example, one can take finite elements of any kind one at a time. Their local properties can be developed by considering them in isolation, as individual entities. This is the key to the programming of element libraries.

In the Direct Stiffness Method, elements are isolated by disconnection and localization. This procedure involves the separation of elements from their neighbors by disconnecting the nodes, followed by the referral of the element to a convenient local coordinate system. After these two steps we can consider *generaic* elements: a bar element, a beam element, and so on. From the standpoint of computer implementation, it means that you can write one subroutine or module that constructs all elements of one type, instead of writing one for each element instance.

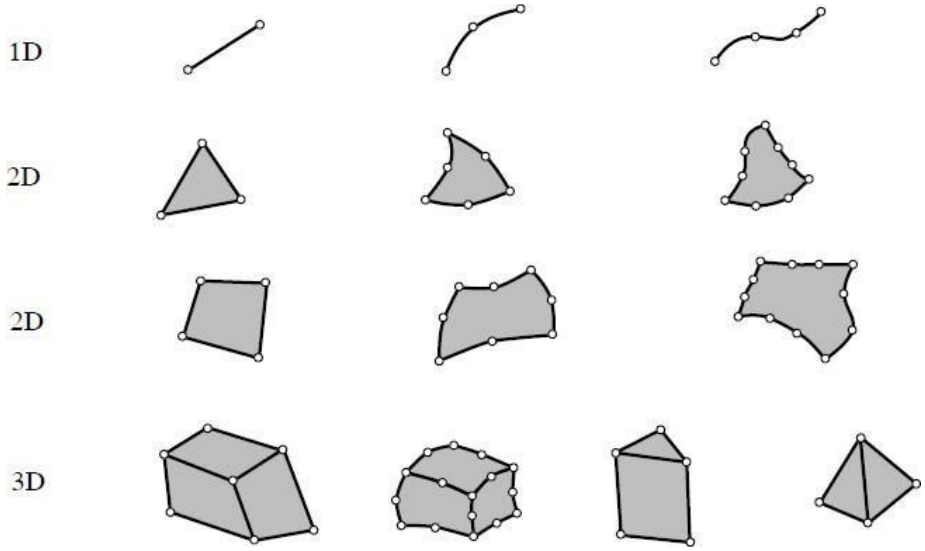


Figure 3: Typical finite element geometries in one through three dimensions

The following is a summary of the data associated with an individual finite element. This data is used in finite element programs to carry out element level calculations.

Dimensionality. Elements can have one, two or three space dimensions. (There are also

special elements with zero dimensionality, such as lumped springs.)

Nodal points. Each element possesses a set of distinguishing points called *nodal points* or *nodes* for short. Nodes serve two purposes: definition of element geometry, and home for degrees of freedom. They are located at the corners or end points of elements (see Figure 3); in the so-called refined or higher-order elements nodes are also placed on sides or faces.

Geometry. The geometry of the element is defined by the placement of the nodal points. Most elements used in practice have fairly simple geometries. In one-dimension, elements are usually straight lines or curved segments. In two dimensions they are of triangular or quadrilateral shape.
In three dimensions the three common shapes are tetrahedra, pentahedra (also called wedges or prisms), and hexahedra (also called cuboids or ―bricks‖). See Figure 3.

Degrees of freedom. The degrees of freedom (DOF) specify the *state* of the element. They also function as ―handles‖ through which adjacent elements are connected. DOFs are defined as the values (and possibly derivatives) of a primary field variable at nodal points.

The actual selection depends on criteria studied at length in Part II. Here we simply note that the key factor is the way in which the primary variable appears in the mathematical model. For mechanical elements, the primary variable is the displacement field and the DOF for many (but not all) elements are the displacement components at the nodes.
Nodal forces. There is always a set of nodal forces in a one-to-one correspondence with degrees of freedom. In mechanical elements the correspondence is established through energy arguments.

Constitutive properties. For a mechanical element these is the relation that specifies the material properties. For example, in a linear elastic bar element it is sufficient to specify the elastic modulus *E* and the thermal coefficient of expansion.

Fabrication properties. For a mechanical element these are fabrication properties which have been integrated out from the element dimensionality. Examples are cross sectional properties of MoM elements such as bars, beams and shafts, as well as the thickness of a plate or shell element.
This data is used by the element generation subroutines to compute element stiffness relations in the local system.

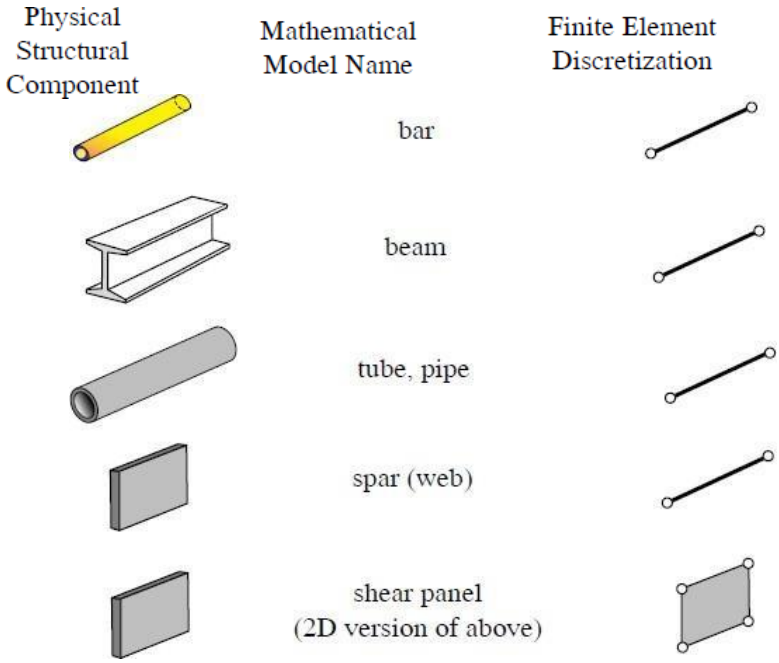


Figure 4. Examples of primitive structural elements

3.5 Assembly Procedure

The assembly procedure of the Direct Stiffness Method for a general finite element model follows rules identical in principle to those discussed for the truss example. As in that case the process involves two basic steps:

Globalization. The element equations are transformed to a common *global* coordinate system.

Merge. The element stiffness equations are merged into the master stiffness equations by appropriate indexing and entry addition.

The computer implementation of this process is not necessarily as simple as the hand calculations of the truss example suggest. The master stiffness relations in practical cases may involve thousands (or even millions) of degrees of freedom. To conserve storage and processing time the use of sparse matrix techniques as well as peripheral storage is required. But this inevitably increases the programming complexity.

3.6 Boundary Conditions

A key strength of the FEM is the ease and elegance with which it handles arbitrary boundary and interface conditions. This power, however, has a down side. One of the biggest hurdles a FEM newcomer faces is the understanding and proper handling of boundary conditions. Surprisingly, prior exposure to partial differential equations, without a balancing study of variational calculus, does not appear to be of much help in this regard.

In the present Section we summarize some basic rules for treating boundary conditions.

3.6.1 Essential and Natural B.C.

The important thing to remember is that boundary conditions (BCs) come in two basic flavors:
Essential BCs are those that directly affect the degrees of freedom, and are imposed on the left-hand side vector **u**.
Natural BCs are those that do not directly affect the degrees of freedom, and are imposed on the right-hand side vector **f**.

The mathematical justification for this distinction requires use of the variation calculus, and is consequently relegated to Part II of the course. For the moment, the basic recipe is:

1. If a boundary condition involves one or more degrees of freedom in a *direct* way, it is essential. An example is a prescribed node displacement.
2. Otherwise it is natural.

The term ―direct‖ is meant to exclude derivatives of the primary function, unless those derivatives also appear as degrees of freedom, such as rotations in beams and plates.

3.6.2 Boundary Conditions in Structural Problems

In mechanical problems, essential boundary conditions are those that involve *displacements* (but not strain-type displacement derivatives). The support conditions for the truss problem furnish a particularly simple example. But there are more general boundary conditions that occur in practice.

A structural engineer must be familiar with displacement B.C. of the following types.
Ground or support constraints. Directly restraint the structure against rigid body motions.

Symmetry conditions. To impose symmetry or antisymmetry restraints at certain points, lines or planes of structural symmetry. This allows the discretization to proceed only over part of the structure with a consequent savings in the number of equations to be solved.
Ignorable freedoms. To suppress displacements that are irrelevant to the problem. (In classical dynamics these are called *ignorable coordinates*.) Even experienced users of finite element programs are sometimes baffled by this kind.

Connection constraints. To provide connectivity to adjoining structures or substructures, or to specify relations between degrees of freedom. Many conditions of this type fall under the label.

multipoint constraints or *multifreedom constraints*, which can be notoriously difficult to handle from a numerical standpoint.

FEM allows detailed visualization of where structures bend or twist, and indicates the distribution of stresses and displacements. FEM software provides a wide range of simulation options for controlling the complexity of both modelling and analysis of a system. Similarly, the desired level of accuracy required and associated computational time

requirements can be managed simultaneously to address most engineering applications. FEM allows entire designs to be constructed, refined, and optimized before the design is manufactured.

This powerful design tool has significantly improved both the standard of engineering designs and the methodology of the design process in many industrial applications. The introduction of FEM has substantially decreased the time to take products from concept to the production line. It is primarily through improved initial prototype designs using FEM that testing and development have been accelerated. In summary, benefits of FEM include increased accuracy, enhanced design and better insight into critical design parameters, virtual prototyping, fewer hardware prototypes, a faster and less expensive design cycle, increased productivity, and increased revenue.

3.7 Data-based models

Data modelling is a method used to define and analyze data requirements needed to support the business processes of an organization. The data requirements are recorded as a conceptual data model with associated data definitions. Actual implementation of the conceptual model is called a logical data model. To implement one conceptual data model may require multiple logical data models.

Data modelling defines not just data elements, but their structures and relationships between them. Data modelling techniques and methodologies are used to model data in a standard, consistent, predictable manner in order to manage it as a resource. The use of data modelling standards is strongly recommended for all projects requiring a standard means of defining and analyzing data within an organization, e.g., using data modelling:

- to manage data as a resource;
- for the integration of information systems;
- for designing databases/data warehouses (aka data repositories)

Data modelling may be performed during various types of projects and in multiple phases of projects. Data models are progressive; there is no such thing as the final data model for a business or application. Instead a data model should be considered a living document that will change in response to a changing business. The data models should ideally be stored in a repository so that they can be retrieved, expanded, and edited over time.

Types of Data Models

Whitten (2004) determined two types of data modelling:

- Strategic data modelling: This is part of the creation of an information systems strategy, which defines an overall vision and architecture for information systems is defined. Information engineering is a methodology that embraces this approach.
- Data modelling during systems analysis: In systems analysis logical data models are created as part of the development of new databases.

Data modelling is also a technique for detailing business requirements for a database. It is sometimes called *database modelling* because a data model is eventually implemented in a database.

The main aim of data models is to support the development of data-base by providing the definition and format of data. According to West and Fowler (1999) "if this is done consistently across systems then compatibility of data can be achieved. If the same data structures are used to store and access data then different applications can share data.

However, systems and interfaces often cost more than they should, to build, operate, and maintain. They may also constrain the business rather than support it. A major cause is that the quality of the data models implemented in systems and interfaces is poor. As a consequence:

- Business rules, specific to how things are done in a particular place, are often fixed in the structure of a data model. This means that small changes in the way business is conducted lead to large changes in computer systems and interfaces
- Entity types are often not identified, or incorrectly identified. This can lead to replication of data, data structure, and functionality, together with the attendant costs of that duplication in development and maintenance
- Data models for different systems are arbitrarily different. The result of this is that complex interfaces are required between systems that share data. These interfaces can account for between 25-70% of the cost of current systems
- "Data cannot be shared electronically with customers and suppliers, because the structure and meaning of data has not been standardised. For example, engineering design data and drawings for process plant are still sometimes exchanged on paper

The reason for these problems is a lack of standards that will ensure that data models will both meet business needs and be consistent.

3.8 The three perspectives of Data model

The perspectives shows that a data model can be an external model (or view), a conceptual model, or a physical model. This is not the only way to look at data models, but it is a useful way, particularly when comparing models.

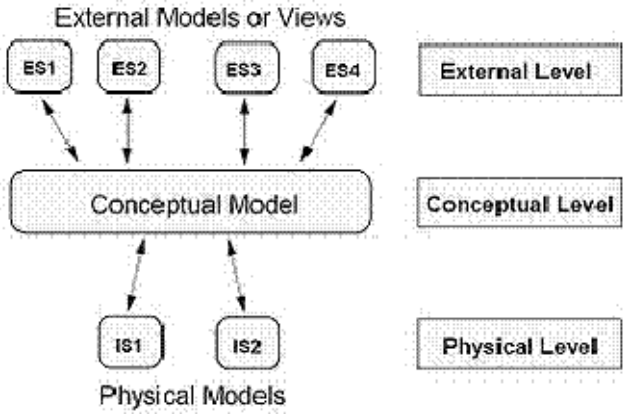


Fig. 6: ANSI/SPARC **three level architecture**.

A data model *instance* may be one of three kinds according to ANSI in 1975:

Conceptual schema - A conceptual schema specifies the kinds of facts or propositions that can be expressed using the model. In that sense, it defines the allowed expressions in an artificial 'language' with a scope that is limited by the scope of the model. It describes the semantics of a domain. For example, it may be a model of the interest area of an organization or industry. This consists of entity classes, representing kinds of things of significance in the domain, and relationships assertions about associations between pairs of entity classes The use of conceptual schema has evolved to become a powerful communication tool with business users. Often called a subject area model (SAM) or high-level data model (HDM), this model is used to communicate core data concepts, rules, and definitions to a business user as part of an overall application development or enterprise initiative. The number of objects should be very small and focused on key concepts. Try to limit this model to one page, although for extremely large organizations or complex projects, the model might span two or more pages

Logical schema - describes the semantics, as represented by a particular data manipulation technology. This consists of descriptions of tables and columns, object oriented classes, and XML tags, among other things.

Physical schema - describes the physical means by which data are stored. This is concerned with partitions, CPUs, tablespaces, and the like.
The significance of this approach, according to ANSI, is that it allows the three perspectives to be relatively independent of each other.
Storage technology can change without affecting either the logical or the conceptual model. The table/column structure can change without (necessarily) affecting the conceptual model. In each case, of course, the structures must remain consistent with the other model. The table/column structure may be different from a direct translation of the entity classes and attributes, but it must ultimately carry out the objectives of the conceptual entity class structure. Early phases of many software development projects

emphasize the design of a **conceptual data model** . Such a design can be detailed into a **logical data model** . In later stages, this model may be translated into **physical data model**. However, it is also possible to implement a conceptual model directly

3.9 Database Model

A **database model** is a theory or specification describing how a database is structured and used. Several such models have been suggested. Common models include:

Flat model: This may not strictly qualify as a data model. The flat (or table) model consists of a single, two-dimensional array of data elements, where all members of a given column are assumed to be similar values, and all members of a row are assumed to be related to one another.

Hierarchical model: In this model data is organized into a tree-like structure, implying a single upward link in each record to describe the nesting, and a sort field to keep the records in a particular order in each same-level list.

Network model: This model organizes data using two fundamental constructs, called records and sets. Records contain fields, and sets define one-to-many relationships between records: one owner, many members.

Relational model: is a database model based on first-order predicate logic. Its core idea is to describe a database as a collection of predicates over a finite set of predicate variables, describing constraints on the possible values and combinations of values.

Object-relational model: Similar to a relational database model, but objects, classes and inheritance are directly supported in database schemas and in the query language.

4.0 Self-Assessment Exercise(s)

Answer the following questions:

1. Draw the FEM physical simulation process
2. Briefly discuss each of the data assciated with FE used in programs for elementary calculations.
3. With the aid of diagrams differentiate between the common data model
4. Briefly discuss the basic rules for the treatment of boundary conditions
5. What is FEM and how/where can it be applied?
6. How does the ANSI three perspectives to data model allows for relatively independent of each.
7. What is the purpose of FEM discretization

5.0 Conclusion

The development of systems and interfaces often cost more than they should, to build, operate, and maintain. A major cause is that the quality of the data models implemented in systems and interfaces is poor. This usually is as a result of:

- Violation of business rules, as a result small changes in the way business is conducted lead to large changes in computer systems and interfaces.
- Unidentified or incorrect identification of entity which can lead to replication of data, data structure, and functionality, and increased costs of development and maintenance

Consequently data cannot be shared electronically with customers and suppliers due to unstructured and lack of standard data that can meet business needs.

6.0 Summary

In this unit, we have discussed elaborately,

- the Physics-based Finite Element Method (FEM) which is defined as a numerical technique for finding approximate solutions of partial differential equations (PDE) as well as of integral equations.
    - Here we stated the uses, traced its origin and discussed its aplications
    - Carried out the overview of the FEM basics including:
        - Idealization
        - Discretization; purpose and error sources
        - Finite and Boundary element methods
        - Interpretations of FEM and
        - Elementary attributes used in FEM
- Stated the three perspectives of data models; Conceptual, logical and Physical schemas.
- The different types of database models; Flat, hierarchical, network, Relational and Object oriented models.

7.0 Further Readings

- Gordon, S. I., & Guilfoos, B. (2017). *Introduction to Modeling and Simulation with MATLAB® and Python*. Milton: CRC Press.
- Zeigler, B. P., Muzy, A., & Kofman, E. (2019). *Theory of modeling and simulation: Discrete event and iterative system computational foundations*. San Diego (Calif.): Academic Press.
- Kluever, C. A. (2020). *Dynamic systems modeling, simulation, and control*. Hoboken, N.J: John Wiley & Sons.

- Law, A. M. (2015). *Simulation modeling and analysis*. New York: McGraw-Hill.
- Verschuuren, G. M., & Travise, S. (2016). *100 Excel Simulations: Using Excel to Model Risk, Investments, Genetics, Growth, Gambling and Monte Carlo Analysis*. Holy Macro! Books.
- Grigoryev, I. (2015). *AnyLogic 6 in three days: A quick course in simulation modeling*. Hampton, NJ: AnyLogic North America.
- Dimotikalis, I., Skiadas, C., & Skiadas, C. H. (2011). *Chaos theory: Modeling, simulation and applications: Selected papers from the 3rd Cghaotic Modeling and Simulation International Conference (CHAOS2010), Chania, Crete, Greece, 1-4 June, 2010.* Singapore: World Scientific.
- Velten, K. (2010). *Mathematical modeling and simulation: Introduction for scientists and engineers*. Weinheim: Wiley-VCH.

Unit 4: Statistics for Modelling and Simulation

Contents

1.0 Introduction

In this unit we will discuss two ways statistics are computed and applied in modelling and simulations these include: inference and descriptive processes. Statistical inference is generally distinguished from descriptive statistics. In simple terms, descriptive statistics can be thought of as being just a straightforward presentation of facts, in which modelling decisions made by a data analyst have had minimal influence. Statistical inference is the process of drawing conclusions from data that are subject to random variation, for example, observational errors or sampling variation. A complete statistical analysis will nearly always include both descriptive statistics and statistical inference, and will often progress in a series of steps where the emphasis moves gradually from description to inference.

2.0 Intended Learning Outcomes (ILOs)

By the end of this unit you should be able to:

- Differentiate between Descriptive and Inference statistics
- Describe the features of descriptive statistics
- Describe features of Inference statistics
- Compute the essential statistics for simulation

3.0 Main Content

3.1 Descriptive and Inference statistics

Descriptive statistics describe the main features of a collection of data quantitatively. Descriptive statistics are distinguished from inferential statistics (or inductive statistics), in

that descriptive statistics aim to summarize a data set quantitatively without employing a probabilistic formulation, rather than use the data to make inferences about the population that the data are thought to represent. Even when a data analysis draws its main conclusions using inferential statistics, descriptive statistics are generally also presented. For example in a paper reporting on a study involving human subjects, there typically appears a table giving the overall sample size, sample sizes in important subgroups (e.g., for each treatment or exposure group), and demographic or clinical characteristics such as the average age, the proportion of subjects of each sex, and the proportion of subjects with related co morbidities.

Inferential statistics tries to make inferences about a population from the sample data. We also use inferential statistics to make judgments of the probability that an observed difference between groups is a dependable one, or that it might have happened by chance in this study. Thus, we use inferential statistics to make inferences from our data to more general conditions; we use descriptive statistics simply to describe what's going on in our data.

3.1 Descriptive Statistics

Descriptive statistics provide simple summaries about the sample and the measures. Together with simple graphics analysis, they form the basis of quantitative analysis of data. Descriptive statistics summarize data. For example, the shooting percentage in basketball is a descriptive statistics that summarizes the performance of a player or a team. The percentage is the number of shots made divided by the number of shots taken. A player who shoots 33% is making approximately one shot in every three. One making 25% is hitting once in four. The percentage summarizes or describes multiple discrete events. Or, consider the score of many students, the grade point average. This single number describes the general performance of a student across the range of their course experiences.

One that describes a large set of observations with a single indicator risks distorting the original data or losing important detail. For example, the shooting percentage doesn't tell you whether the shots are three-pointers or lay-ups, and GPA doesn't tell you whether the student was in difficult or easy courses. Despite these limitations, descriptive statistics provide a powerful summary that may enable comparisons across people or other units.

3.1.1 Univariate Analysis

Univariate analysis involves the examination across cases of a single variable, focusing on three characteristics: the distribution; the central tendency; and the dispersion. It is common to compute all three for each study variable.

a. Distribution

The distribution is a summary of the frequency of individual or ranges of values for a variable. The simplest distribution would list every value of a variable and the number of cases that had that value. For instance, computing the distribution of gender in the study population means computing the percentages that are male and female. The gender variable has only two, making it possible and meaningful to list each one. However, this does not

work for a variable such as income that has many possible values. Typically, specific values are not particularly meaningful (income of 50,000 is typically not meaningfully different from 51,000). Grouping the raw scores using ranges of values reduces the number of categories to something more meaningful. For instance, we might group incomes into ranges of 0-10,000, 10,001-30,000, etc.

b. Central tendency

The central tendency of a distribution locates the "center" of a distribution of values. The three major types of estimates of central tendency are the *mean*, the *median*, and the *mode*.

The **mean** is the most commonly used method of describing central tendency. To compute the mean, take the sum of the values and divide by the count. For example, the mean quiz score is determined by summing all the scores and dividing by the number of students taking the exam. For example, consider the test score values:

15, 20, 21, 36, 15, 25, 15

The sum of these 7 values is 147, so the mean is 147/7 =21.

The mean is computed using the formular: $\square X_i / n$, where the sum is over i = 1 to n.

The **median** is the score found at the middle of the set of values, i.e., that has as many cases with a larger value as have a smaller value. One way to compute the median is to sort the values in numerical order, and then locate the value in the middle of the list. For example, if there are 500 values, the value in 250th position is the median. Sorting the 8 scores above produces:

15, 15, 15, 20, 21, 25, 36

There are 7 scores and score #4 represents the halfway point. The median is 20. If there is an even number of observations, then the median is the mean of the two middle scores. In the example, if there were an 8th observation, with a value of 25, the median becomes the average of the 4th and 5th scores, in this case 20.5:

15, 15, 15, 20, 21, 25, 25, 36

The **mode** is the most frequently occurring value in the set. To determine the mode, compute the distribution as above. The mode is the value with the greatest frequency. In the example, the modal value 15, occurs three times. In some distributions there is a "tie" for the highest frequency, i.e., there are multiple modal values. These are called **multi- modal** distributions.

Notice that the three measures typically produce different results. The term "average" obscures the difference between them and is better avoided. The three values are equal if the distribution is perfectly "**normal**" (i.e., bell-shaped).

c. Dispersion

Dispersion is the spread of values around the central tendency. There are two common measures of dispersion, the range and the standard deviation. The **range** is simply the highest value minus the lowest value. In our example distribution, the high value is 36 and the low is 15, so the range is 36 − 15 = 21.

The **standard deviation** is a more accurate and detailed estimate of dispersion because an outlier can greatly exaggerate the range (as was true in this example where the single outlier value of 36 stands apart from the rest of the values). The standard deviation shows the relation that set of scores has to the mean of the sample. Again let's take the set of scores:
15, 20, 21, 36, 15, 25, 15
to compute the standard deviation, we first find the distance between each value and the mean. We know from above that the mean is 21. So, the differences from the mean are:
15 − 21 = −6
20 − 21 = −1
21 − 21 = 0
36 − 21 = 15
15 − 21 = −6
25 − 21 = +4
15 − 21 = −6

Notice that values that are below the mean have negative differences and values above it have positive ones. Next, we square each difference:
$(6)^2 = 36$
$(-1)^2 = 1$
$(+0)^2 = 0$
$(15)^2 = 225$
$(-6)^2 = 36$
$(+4)^2 = 16$
$(-6)^2 = 36$
Now, we take these "squares" and sum them to get the **sum of squares** (SS) value. Here, the sum is 350. Next, we divide this sum by the number of scores minus 1. Here, the result is 350 / 6 = 58.3. This value is known as the **variance**. To get the standard deviation, we take the square root of the variance (remember that we squared the deviations earlier). This would be √58.3 = 7.63.

Although this computation may seem intricate, it's actually quite simple. In English, we can describe the standard deviation as:
the square root of the sum of the squared deviations from the mean divided by the number of scores minus one given as: $\sqrt{(\square (x_i - u)^2)/ n}$; where x = observed value and u = the mean

The standard deviation allows us to reach some conclusions about specific scores in our distribution. Assuming that the distribution of scores is close to "normal", the following

conclusions can be reached:

a. approximately 68% of the scores in the sample fall within one standard deviation of the mean (u - SD) and (u + SD)
b. approximately 95% of the scores in the sample fall within two standard deviations of the mean (u-2SD) and (u+2SD)
c. approximately 99% of the scores in the sample fall within three standard deviations of the mean (u - 3SD) and (u + 3SD)

For example, since the mean in our example is 21 and the standard deviation is 7.63, we can from the above statement estimate that approximately 95% of the scores will fall in the range of 21 − (2×7.63) to 21 + (2×7.63) or between 5.74 and 36.26. Values beyond two standard deviations from the mean can be considered "outliers". 36 is the only such value in our distribution.

Outliers help identify observations for further analysis or possible problems in the observations. Standard deviations also convert measures on very different scales, such as height and weight, into values that can be compared.

d. Other Statistics

In research involving comparisons between groups, emphasis is often placed on the **significance level** for the **hypothesis** that the groups being compared differ to a degree greater than would be expected by chance. This significance level is often represented as a **p-value**, or sometimes as the standard score of a test statistic. In contrast, an **effect size** conveys the estimated magnitude and direction of the difference between groups, without regard to whether the difference is statistically significant. Reporting significance levels without effect sizes is problematic, since for large sample sizes even small effects of little practical importance can be statistically significant.

3.1.2 Examples of descriptive statistics

Most statistics can be used either as a descriptive statistic, or in an inductive analysis. For example, we can report the average reading test score for the students in each classroom in a school, to give a descriptive sense of the typical scores and their variation. If we perform a formal *hypothesis test* on the scores, we are doing *inductive* rather than descriptive analysis.

The following is a list some statistical common in descriptive analyses:

- Measures of central tendency
- Measures of dispersion
- Measures of association
- Cross-tabulation, contingency table
- Histogram
- Quantile, Q-Q plot
- Scatter plot
- Box plot

3.3 Inference Statistics

The terms **statistical inference**, **statistical induction** and **inferential statistics** are used to describe systems of procedures that can be used to draw conclusions from datasets arising from systems affected by random variation. Initial requirements of such a system of procedures for *inference* and *induction* are that the system should produce reasonable answers when applied to well-defined situations and that it should be general enough to be applied across a range of situations.

The outcome of statistical inference may be an answer to the question "what should be done next?", where this might be a decision about making further experiments or surveys, or about drawing a conclusion before implementing some organizational or governmental policy.

For the most part, statistical inference makes propositions about populations, using data drawn from the population of interest via some form of random sampling. More generally, data about a random process is obtained from its observed behaviour during a finite period of time. Given a parameter or hypothesis about which one wishes to make inference, statistical inference most often uses:

- a statistical model of the random process that is supposed to generate the data, and
- a particular realization of the random process; i.e., a set of data.

The conclusion of a **statistical inference** is a statistical **proposition**.

3.3.1 Some common forms of statistical proposition

- An **estimate** - a particular value that best approximates some parameter of interest,
- A **confidence interval** (or set estimate) - an interval constructed from the data in such a way that, under repeated sampling of datasets, such intervals would contain the true parameter value with the probability at the stated confidence level,
- A **credible interval** - a set of values containing, for example, 95% of posterior belief,
- Rejection of an **hypothesis**
- **Clustering** or classification of data points into groups

3.3.2 Models/Assumptions

Any statistical inference requires some assumptions. A **statistical model** is a set of assumptions concerning the generation of the observed data and similar data. Descriptions of statistical models usually emphasize the role of population quantities of interest, about which we wish to draw inference.

3.3.3 Degree of models/assumptions

Statisticians distinguish between three levels of modelling assumptions;

- **Fully parametric**: The probability distributions describing the data-generation process are assumed to be fully described by a family of probability distributions involving only a finite number of unknown parameters. For example, one may assume that the distribution of population values is truly Normal, with unknown mean and variance, and that

datasets are generated by 'simple' random sampling. The family of *generalized linear models* is a widely-used and flexible class of parametric models.

- **Non-parametric**: The assumptions made about the process of generating the data are much less than in parametric statistics and may be minimal. For example, every continuous probability distribution has a median, which may be estimated using the sample median or the Hodges-Lehmann-Sen estimator, which has good properties when the data arise from simple random sampling.
- **Semi-parametric**: This term typically implies assumptions 'between' fully and non-parametric approaches. For example, one may assume that a population distribution have a finite mean. Furthermore, one may assume that the mean response level in the population depends in a truly linear manner on some covariate (a parametric assumption) but does not make any parametric assumption describing the variance around that mean. More generally, semi-parametric models can often be separated into 'structural' and 'random variation' components. One component is treated parametrically and the other non-parametrically.

3.3.4 Importance of valid models/assumptions

Whatever level of assumption is made, correctly-calibrated inference in general requires these assumptions to be correct; i.e., that the data-generating mechanisms really has been correctly specified.

- Incorrect assumptions of *'simple' random sampling* can invalidate statistical inference.
- More complex semi- and fully-parametric assumptions are also cause for concern. For example, incorrect assumptions of Normality in the population can invalidates some forms of regression-based inference.
- The use of **any** parametric model is viewed skeptically by most experts in sampling human populations: "most sampling statisticians, when they deal with confidence intervals at all, limit themselves to statements about [estimators] based on very large samples, where the central limit theorem ensures that these [estimators] will have distributions that are nearly normal." Here, the central limit theorem states that the distribution of the sample mean "for very large samples" is approximately normally distributed, if the distribution is not heavy tailed.

3.3.5 Approximate distributions

Given the difficulty in specifying exact distributions of sample statistics, many methods have been developed for approximating these.

With ***finite samples***, approximation results measure how close a limiting distribution approaches the statistic's sample distribution: For example, with 10,000 independent samples the normal distribution approximates (to two digits of accuracy) the distribution of the sample mean for many population distributions. Yet for many practical purposes, the normal approximation provides a good approximation to the sample-mean's distribution when there are 10 (or more) independent samples, according to simulation studies, and statisticians' experience. Following Kolmogorov's work in the 1950s, advanced statistics uses approximation theory and functional analysis to quantify the error of approximation: In this approach, the metric geometry of probability distributions is studied; this approach quantifies

approximation error.

With ***infinite samples***, limiting results like the central limit theorem describe the sample statistic's limiting distribution, if one exists. Limiting results are not statements about finite samples, and indeed are logically irrelevant to finite samples. However, the asymptotic theory of limiting distributions is often invoked for work in estimation and testing. For example, limiting results are often invoked to justify the generalized method of moments and the use of generalized estimating equations, which are popular in econometrics and biostatistics. The magnitude of the difference between the limiting distribution and the true distribution (formally, the 'error' of the approximation) can be assessed using simulation. The use of limiting results in this way works well in many applications, especially with low-dimensional models with log-concave likelihoods (such as with one-parameter exponential families).

3.3.6 Randomization-based models

For a given dataset that was produced by a randomization design, the randomization distribution of a statistic (under the null-hypothesis) is defined by evaluating the test statistic for all of the plans that could have been generated by the randomization design.

In frequentist inference, randomization allows inferences to be based on the randomization distribution rather than a subjective model, and this is important especially in survey sampling and design of experiments. Statistical inference from randomized studies is also more straightforward than many other situations.

In Bayesian inference, randomization is also of importance: In survey sampling – *sampling without replacement* ensures the *exchangeability* of the sample with the population; in randomized experiments, randomization warrants a *missing at random* assumption for covariate information.

Objective randomization allows properly inductive procedures. Many statisticians prefer randomization-based analysis of data that was generated by well-defined randomization procedures. However, it is has been observed that in fields of science with developed theoretical knowledge and experimental control, randomized experiments may increase the costs of experimentation without improving the quality of inferences. Similarly, results from randomized experiments are recommended by leading statistical authorities as allowing inferences with greater reliability than do observational studies of the same phenomena. However, a good observational study may be better than a bad randomized experiment.

The statistical analysis of a randomized experiment may be based on the randomization scheme stated in the experimental protocol and does not need a subjective model. However, not all hypotheses can be tested by randomized experiments or random samples, which often require a large budget, a lot of expertise and time, and may have ethical problems.

3.3.7 Modes of inference

Different schools of statistical inference have become established. These schools (or 'paradigms') are not mutually-exclusive, and methods which work well under one paradigm often have attractive interpretations under other paradigms. The two main paradigms in use are **frequentist** and **Bayesian** inference, which are both summarized below.

a. Frequentist inference

This paradigm regulates the production of propositions by considering (notional) repeated sampling of datasets similar to the one at hand. By considering its characteristics under repeated sample, the frequentist properties of any statistical inference procedure can be described - although in practice this quantification may be challenging. Examples of frequentist inference are: P-value and Confidence interval

The frequentist calibration of procedures can be done without regard to utility functions. However, some elements of frequentist statistics, such as statistical decision theory, do incorporate utility functions. Loss functions must be explicitly stated for statistical theorists to prove that a statistical procedure has an optimality property. For example, median-unbiased estimators are optimal under absolute value loss functions, and least squares estimators are optimal under squared error loss functions.

While statisticians using frequentist inference must choose for themselves the parameters of interest, and the estimators/test statistic to be used, the absence of obviously-explicit utilities and prior distributions has helped frequentist procedures to become widely- viewed as 'objective'.

b. Bayesian inference

The Bayesian calculus describes degrees of belief using the 'language' of probability; beliefs are positive, integrate to one, and obey probability axioms. Bayesian inference uses the available *posterior beliefs* as the basis for making statistical propositions. There are several different justifications for using the Bayesian approach. Examples of Bayesian inference are: *Credible intervals* for interval estimation and *Bayes factors* for model comparison

Many informal Bayesian inferences are based on "intuitively reasonable" summaries of the posterior. For example, the posterior mean, median and mode, highest posterior density intervals, and Bayes Factors can all be motivated in this way. While a user's utility function need not be stated for this sort of inference, these summaries do all depend (to some extent) on stated earlier beliefs, and are generally viewed as subjective conclusions.

Formally, Bayesian inference is calibrated with reference to an explicitly stated utility, or loss function; the 'Bayes rule' is the one which maximizes expected utility, averaged over the subsequent uncertainty. Formal Bayesian inference therefore automatically provides optimal decisions in a decision theoretic sense. Given assumptions, data and utility, Bayesian inference can be made for essentially any problem, although not every statistical inference need have a Bayesian interpretation. Some advocates of Bayesian inference assert that inference *must* take place in this decision-theoretic framework, and that Bayesian inference

should not conclude with the evaluation and summarization of posterior beliefs.

3.4 Other Essential Statistics for Simulations

3.4.1 Sample Size Determination

A common goal of survey research is to collect data representative of a population. The researcher uses information gathered from the survey to generalize findings from a drawn sample back to a population, within the limits of random error. However, when critiquing business education research, Wunsch (1986) stated that ―two of the most consistent flaws included:

1. disregard for sampling error when determining sample size, and
2. disregard for response and non-response bias‖.

Within a quantitative survey design, determining sample size and dealing with no response bias is essential. ―One of the real advantages of quantitative methods is their ability to use smaller groups of people to make inferences about larger groups that would be prohibitively expensive to study‖. The question then is, how large of a sample is required to infer research findings back to a population?

Standard textbook authors and researchers offer tested methods that allow studies to take full advantage of statistical measurements, which in turn give researchers the upper hand in determining the correct sample size. Sample size is one of the four inter-related features of a study design that can influence the detection of significant differences, relationships or interactions (Peers, 1996). Generally, these survey designs try to minimize both alpha error (finding a difference that does not actually exist in the population) and beta error (failing to find a difference that actually exists in the population) (Peers, 1996).

However, improvement is needed. Researchers are learning experimental statistics from highly competent statisticians and then doing their best to apply the formulas and approaches

Foundations for Sample Size Determination

Primary Variables of Measurement

The researcher must make decisions as to which variables will be incorporated into formula calculations. For example, if the researcher plans to use a seven-point scale to measure a continuous variable, e.g., job satisfaction, and also plans to determine if the respondents differ by certain categorical variables, e.g., gender, tenured, educational level, etc., which variable(s) should be used as the basis for sample size? This is important because the use of gender as the primary variable will result in a substantially larger sample size than if one used the seven-point scale as the primary variable of measure.

Cochran (1977) addressed this issue by stating that ―One method of determining sample size is to specify margins of error for the items that are regarded as most vital to the survey. An estimation of the sample size needed is first made separately for each of these important items‖. When these calculations are completed, researchers will have a range of n‘s, usually ranging from smaller n‘s for scaled, continuous variables, to larger n‘s for dichotomous or categorical variables.

The researcher should make sampling decisions based on these data. If the n‘s for the variables of interest are relatively close, the researcher can simply use the largest n as the sample size and be confident that the sample size will provide the desired results.

More commonly, there is a sufficient variation among the n‘s so that we are reluctant to choose the largest, either from budgetary considerations or because this will give an over- all standard of precision substantially higher than originally contemplated. In this event, the desired standard of precision may be relaxed for certain of the items, in order to permit the use of a smaller value of n. The researcher may also decide to use this information in deciding whether to keep all of the variables identified in the study. ―In some cases, the n‘s are so discordant that certain of them must be dropped from the inquiry; . . .‖.

Error Estimation

Cochran‘s (1977) formula uses two key factors:

(1) the risk the researcher is willing to accept in the study, commonly called the margin of error, or the error the researcher is willing to accept, and

(2) the alpha level, the level of acceptable risk the researcher is willing to accept that the true margin Alpha Level.

The alpha level used in determining sample size in most educational research studies is either .05 or .01 (Ary, Jacobs, & Razavieh, 1996). In Cochran‘s formula, the alpha level is incorporated into the formula by utilizing the t-value for the alpha level selected (e.g., t-value for alpha level of .05 is 1.96 for sample sizes above 120). Researchers should ensure they use the correct t- value when their research involves smaller populations, e.g., t-value for alpha of .05 and a population of 60 is 2.00.

In general, an alpha level of .05 is acceptable for most research. An alpha level of .10 or lower may be used if the researcher is more interested in identifying marginal relationships, differences or other statistical phenomena as a precursor to further studies.

An alpha level of .01 may be used in those cases where decisions based on the research are critical and errors may cause substantial financial or personal harm, e.g., major programmatic changes.

Acceptable Margin of Error

The general rule relative to acceptable margins of error in educational and social research is as follows: For categorical data, 5% margin of error is acceptable, and, for continuous data, 3% margin of error is acceptable (Krejcie & Morgan, 1970). For example, a 3% margin of error would result in the researcher being confident that the true mean of a seven point scale is within ±.21 (.03 times seven points on the scale) of the mean calculated from the research sample. For a dichotomous variable, a 5% margin of error would result in the researcher being confident that the proportion of respondents who were male was within ±5% of the proportion calculated from the research sample. Researchers may increase these values when a higher margin of error is acceptable or may decrease these values when a higher degree of precision is needed.

Variance Estimation

A critical component of sample size formulas is the estimation of variance in the primary variables of interest in the study. The researcher does not have direct control over variance and must incorporate variance estimates into research design. Cochran (1977) listed four

ways of estimating population variances for sample size determinations:

(1) take the sample in two steps, and use the results of the first step to determine how many additional responses are needed to attain an appropriate sample size based on the variance observed in the first step data;
(2) use pilot study results;
(3) use data from previous studies of the same or a similar population; or
(4) estimate or guess the structure of the population assisted by some logical mathematical results.

The first three ways are logical and produce valid estimates of variance; therefore, they do not need to be discussed further. However, in many educational and social research studies, it is not feasible to use any of the first three ways and the researcher must estimate variance using the fourth method.

A researcher typically needs to estimate the variance of scaled and categorical variables. To estimate the variance of a scaled variable, one must determine the inclusive range of the scale, and then divide by the number of standard deviations that would include all possible values in the range, and then square this number. For example, if a researcher used a seven-point scale and given that six standard deviations (three to each side of the mean) would capture 98% of all responses, the calculations would be as follows:

$$S = \frac{\text{7 (number of points on the scale)}}{\text{6 (number of standard deviations)}}$$

When estimating the variance of a dichotomous (proportional) variable such as gender, Krejcie and Morgan (1970) recommended that researchers should use .50 as an estimate of the population proportion. This proportion will result in the maximization of variance, which will also produce the maximum sample size. This proportion can be used to estimate variance in the population. For example, squaring .50 will result in a population variance estimate of .25 for a dichotomous variable.

Basic Sample Size Determination

a. Continuous Data

Before proceeding with sample size calculations, assuming continuous data, the researcher should determine if a categorical variable will play a primary role in data analysis. If so, the categorical sample size formulas should be used. If this is not the case, the sample size formulas for continuous data described in this section are appropriate.

Assume that a researcher has set the alpha level a priori at .05, plans to use a seven point scale, has set the level of acceptable error at 3%, and has estimated the standard deviation of the scale as 1.167. Cochran's sample size formula for continuous data and an example of its use is presented here along with the explanations as to how these decisions were made.

$$n_0 = \frac{(t)^2 * (s)^2}{(d)^2} = \frac{(1.96)^2(1.167)^2}{(7*.03)^2} = 118$$

Where t = value for selected alpha level of .025 in each tail = 1.96 (the alpha level of .05

indicates the level of risk the researcher is willing to take that true margin of error may exceed the acceptable margin of error.)
s = estimate of standard deviation in the population = 1.167 (estimate of variance deviation for 7 point scale calculated by using 7 [inclusive range of scale] divided by 6 [number of standard deviations that include almost all (approximately 98%) of the possible values in the range]).
d = acceptable margin of error for mean being estimated = .21 (number of points on primary scale * acceptable margin of error; points on primary scale = 7; acceptable margin of error = .03 [error researcher is willing to except]).

Therefore, for a population of 1,679, the required sample size is 118. However, since this sample size exceeds 5% of the population (1,679*.05=84), Cochran's (1977) correction formula should be used to calculate the final sample size. These calculations are as follows:

$$n = \frac{no}{(1 + no / Population)} = \frac{(118)}{(1 + 118/1679)} = 111$$

Where population size = 1,679.
n0 = required return sample size according to Cochran's formula= 118. n1 = required return sample size because sample > 5% of population.
These procedures result in the minimum returned sample size. If a researcher has a captive audience, this sample size may be attained easily.
However, since many educational and social research studies often use data collection methods such as surveys and other voluntary participation methods, the response rates are typically well below 100%. Salkind (1997) recommended over-sampling when he stated that ―If you are mailing out surveys or questionnaires . . . count on increasing your sample size by 40%-50% to account for lost mail and uncooperative subjects‖. But Over- sampling can add costs to the survey but is often necessary. A second consequence is, of course, that the variances of estimates are increased because the sample actually obtained is smaller than the target sample.
However, many researchers criticize the use of over-sampling to ensure that this minimum sample size is achieved and suggestions on how to secure the minimal sample size are scarce. If the researcher decides to use over-sampling, four methods may be used to determine the anticipated response rate:
(1) take the sample in two steps, and use the results of the first step to estimate how many additional responses may be expected from the second step;
(2) use pilot study results;
(3) use responses rates from previous studies of the same or a similar population; or
(4) estimate the response rate. The first three ways are logical and will produce valid estimates of response.

b. Categorical Data

The sample size formulas and procedures used for categorical data are very similar, but some

variations do exist. Assume a researcher has set the alpha level a priori at .05, plans to use a proportional variable, has set the level of acceptable error at 5%, and has estimated the standard deviation of the scale as .5. Cochran's sample size formula for categorical data and an example of its use is presented here along with explanations as to how these decisions were made.

$$no = \frac{(t)2 * (p)(q)}{(d)2}$$

$$no = \frac{(1.96)2(.5)(.5)}{(.05)2} = 384$$

Where t = value for selected alpha level of .025 in each tail = 1.96 (the alpha level of .05 indicates the level of risk the researcher is willing to take that true margin of error may exceed the acceptable margin of error).
Where (p)(q) = estimate of variance = .25 (maximum possible proportion (.5) * 1- maximum possible proportion (.5) produces maximum possible sample size).
Where d = acceptable margin of error for proportion being estimated = .05 (error researcher is willing to except).

Therefore, for a population of 1,679, the required sample size is 384. However, since this sample size exceeds 5% of the population (1,679*.05=84), Cochran's (1977) correction formula should be used to calculate the final sample size. These calculations are as follows:

$$n_1 = \frac{n_0}{(1 + n_o / \text{Population})}$$

$$n_1 = \frac{(384)}{(1 + 384/1679)} = 313$$

Where population size = 1,679,
n_0 = required return sample size according to Cochran's formula= 384,
n_1 = required return sample size because sample > 5% of population

These procedures result in a minimum returned sample size of 313. Using the same oversampling procedures as cited in the continuous data example, and again assuming a response rate of 65%, a minimum drawn sample size of 482 should be used. These calculations were based on the following:

Where anticipated return rate = 65%.
Where n2 = sample size adjusted for response rate. Where minimum sample size (corrected) = 313.

Therefore, n2 = 313/.65 = 482.

3.4.2 The Central Limit Theorem

The main idea of the central limit theorem (CLT) is that the average of a sample of observations drawn from some population with any shape-distribution is approximately distributed as a normal distribution if certain conditions are met. In theoretical statistics there are several versions of the central limit theorem depending on how these conditions are specified. These are concerned with the types of assumptions made about the distribution of the parent population (population from which the sample is drawn) and the actual sampling procedure.

One of the simplest versions of the theorem says that if is a random sample of size n (say, n larger than 30) from an infinite population, finite standard deviation, then the standardized sample mean converges to a standard normal distribution or, equivalently, the sample mean approaches a normal distribution with mean equal to the population mean and standard deviation equal to standard deviation of the population divided by the square root of sample size n. In applications of the central limit theorem to practical problems in statistical inference, however, statisticians are more interested in how closely the approximate distribution of the sample mean follows a normal distribution for finite sample sizes, than the limiting distribution itself. Sufficiently close agreement with a normal distribution allows statisticians to use normal theory for making inferences about population parameters (such as the mean) using the sample mean, irrespective of the actual form of the parent population.

It is well known that whatever the parent population is, the standardized variable will have a distribution with a mean 0 and standard deviation 1 under random sampling. Moreover, if the parent population is normal, then it is distributed exactly as a standard normal variable for any positive integer n. The central limit theorem states the remarkable result that, even when the parent population is non-normal, the standardized variable is approximately normal if the sample size is large enough (say > 30). It is generally not possible to state conditions under which the approximation given by the central limit theorem works and what sample sizes are needed before the approximation becomes good enough. As a general guideline, statisticians have used the prescription that if the *parent distribution is symmetric and relatively short-tailed*, then the sample mean reaches approximate normality for smaller samples than if the parent population is skewed or long-tailed.

Under certain conditions, in large samples, the sampling distribution of the sample mean can be approximated by a normal distribution. The sample size needed for the approximation to be adequate depends strongly on the shape of the parent distribution. Symmetry (or lack thereof) is particularly important. For a symmetric parent distribution, even if very different from the shape of a normal distribution, an adequate approximation can be obtained with small samples (e.g., 10 or 12 for the uniform distribution). For symmetric short-tailed parent distributions, the sample mean reaches approximate normality for smaller samples than if the parent population is skewed and long-tailed. In some extreme cases (e.g. binomial) samples sizes far exceeding the typical guidelines (e.g., 30) are needed for an adequate approximation.

For some distributions without first and second moments (e.g., Cauchy), the central limit theorem does not hold.

3.4.3 The Least Squares Model

Many problems in analyzing data involve describing how variables are related. The simplest of all models describing the relationship between two variables is a linear, or straight-line, model. The simplest method of fitting a linear model is to "eye-ball" a line through the data on a plot. A more elegant, and conventional method is that of "least squares", which finds the line minimizing the sum of distances between observed points and the fitted line. With this you will:

- Realize that fitting the "best" line by eye is difficult, especially when there is a lot of residual variability in the data.
- Know that there is a simple connection between the numerical coefficients in the regression equation and the slope and intercept of regression line.
- Know that a single summary statistic like a correlation coefficient does not tell the whole story. A scatter plot is an essential complement to examining the relationship between the two variables.

3.4.4 ANOVA: Analysis of Variance

Analysis of Variance or ANOVA enables us to test the difference between 2 or more means. ANOVA does this by examining the ratio of variability between two conditions and variability within each condition. For example, if we give a drug that we believe will improve memory to a group of people and give a placebo to another group of people, we might measure memory performance by the number of words recalled from a list we ask everyone to memorize. A **t-test** would compare the likelihood of observing the difference in the mean number of words recalled for each group. An ANOVA test, on the other hand, would compare the variability that we observe between the two conditions to the variability observed within each condition. We measure variability as the sum of the difference of each score from the mean. When we actually calculate an ANOVA we use a short-cut formula. Thus, when the variability that we predict (between the two groups) is much greater than the variability we don't predict (within each group) then we will conclude that our treatments produce different results.

3.4.5 Exponential Density Function (EDF)

EDF is use to take important class of decision problems under uncertainty such as the chance between events. For example, the chance of the length of time to next breakdown of a machine not exceeding a certain time, such as the photocopying machine in your office not to break during this week.

Exponential distribution gives distribution of time between independent events occurring at a constant rate. Its density function is:

$f(t) = \square \exp(-\square t)$,

where □ is the average number of events per unit of time, which is a positive number. The mean and the variance of the random variable t (time between events) are 1/ □, and 1/□^{2}, respectively

Applications include probabilistic assessment of the time between arrival of patients to the emergency room of a hospital, and arrival of ships to a particular port.

3.4.6 Poisson Process

An important class of decision problems under uncertainty is characterized by the small chance of the occurrence of a particular event, such as an accident. Poisson gives probability of exactly x independent occurrences during a given period of time if events take place independently and at a constant rate. It may also represent number of occurrences over constant areas or volumes. The following statements describe the *Poisson Process*:

1. The occurrences of the events are independent.
2. The occurrence of events from a set of assumptions in an interval of space or time has no effect on the probability of a second occurrence of the event in the same, or any other, interval.
3. Theoretically, an infinite number of occurrences of the event must be possible in the interval.
4. The probability of the single occurrence of the event in a given interval is proportional to the length of the interval.
5. In any infinitesimally small portion of the interval, the probability of more thanone occurrence of the event is negligible.

Poisson processes are often used, for example in quality control, reliability, insurance claim, incoming number of telephone calls, and queuing theory.

An Application: One of the most useful applications of the Poisson Process is in the field of queuing theory. In many situations where queues occur it has been shown that the number of people joining the queue in a given time period follows the Poisson model. For example, if the rate of arrivals to an emergency room is □ per unit of time period (say 1 hr), then:

P (n arrivals) = □^{n} e$^{-□□}$/ n!

The mean and variance of random variable n are both □□. However if the mean and variance of a random variable having equal numerical values, then it is not necessary that its distribution is a Poisson.

Applications:

P (0 arrival) = e$^{-□}$

P (1 arrival) = □ e$^{-□□}$/ 1!P (2 arrival) = □^{2}

e$^{-□□}$/ 2!

and so on. In general:

P (n+1 arrivals) = □□Pr (n arrivals) / n.

3.4.7 Uniform Density Function (UDF)

This function gives the probability that observation will occur within a particular interval when probability of occurrence within that interval is directly proportional to interval length.

For example, it is used to generate random numbers in sampling and Monte Carlo simulation.

The mass function of geometric mean of n independent uniforms [0,1]
is: $P(X = x) = n\, x^{(n-1)} (\text{Log}[1/x^n])^{(n-1)} / (n - 1)!$.
$z_L = [U^L-(1-U)^L] / L$ is said to have Tukey's symmetrical distribution.
You may like to use *Uniform Applet* to perform your computations, then visit also:
http://home.ubalt.edu/ntsbarsh/Business-stat/otherapplets/pvalues.htm

3.4.8 Test for Randomness

We need to test for both randomness as well as uniformity. The tests can be classified in 2 categories: Empirical or statistical tests, and theoretical tests.
Theoretical tests deal with the properties of the generator used to create the realization with desired distribution, and do not look at the number generated at all. For example, we would not use a generator with poor qualities to generate random numbers.
Statistical tests are based solely on the random observations produced.

A. Test for independence:

Plot the x_i realization vs x_{i+1}. If there is independence, the graph will not show any distinctive patterns at all, but will be perfectly scattered.

B. Runs tests.(run-ups, run-downs):

This is a direct test of the independence assumption. There are two test statistics to consider: one based on a normal approximation and another using numerical approximations.
Test based on Normal approximation:
Suppose you have N random realizations. Let K be the total number of runs in a sequence. If the number of positive and negative runs are greater than say 20, the distribution of K is reasonably approximated by a Normal distribution with mean (2N - 1) /3 and (16N - 29) / 90. Reject the hypothesis of independence or existence of runs if $|Zo| < Z(1\text{-alpha}/2)$ where Zo is the Z score.

C. Correlation tests:

Do the random numbers exhibit discernible correlation? Compute the sample Autcorrelation Function.

Frequency or Uniform Distribution Test:
Use Kolmogorov-Smirimov test to determine if the realizations follow a U(0,1).

3.4.9 Some Useful SPSS Commands

a. Test for Binomial:

NPAR TEST BINOMIAL(p)=GENDER(0, 1)

b. Gooness-of-fit for discrete r.v.:

NPAR TEST CHISQUARE=X (1,3)/EXPECTED=20 30 50

C. Two population t-test

T-TEST GROUPS=GENDER(1,2)/VARIABLES=X

4.0 Self-Assessment Exercise(s)

Answer the following questions:

1. State the Cochran's sample size formula for continuous and categorical data
2. Assume a researcher has set the alpha level a priori at 10%, plans to use a proportional variable, has set the level of acceptable error at 5%, and has estimated the standard deviation of the scale as .5. Find the sample size for a population of 2500
3. What are the Cochran's key factors for error estimation?
4. State the essential feature of Poisson process
5. List four ways of estimating population variances for sample size determinations according to Cochran.
6. What are the objectives of randomization, and state the importance of randomization in frequentist and Bayesian inferences

5.0 Conclusion

Statistics is the basis of simulation. In this unit we have simply introduced some basic statistics in modelling and simulations. We hope that the reader will broaden his/her understanding by consulting the referenced texts or othe statistics books.

6.0 Summary

In this unit we were able to

- Differentiate between the two broad components of statistics: descriptive and inference statistics
- Have concise discussions of descriptive statistics on; Univarite statistics measures: the distribution, central tendency, dispersion, etc. and gave some examples.
- Discuss Inference statistics under the following subheads: definition, Model/assumptions, approximate distributions, random-based models and modes of inference.
- Introduce some essential statistical measures in simulation such as;
    - sample size determination
    - central limit theorem

- least square model
- Analysis of variance
- Exponential distribution function
- Poisson distribution
- Uniform distribution
- Test for randomness
- Some commands of Special package for statistical analysis (SPSS)

7.0 Further Readings

- Gordon, S. I., & Guilfoos, B. (2017). *Introduction to Modeling and Simulation with MATLAB® and Python*. Milton: CRC Press.
- Zeigler, B. P., Muzy, A., & Kofman, E. (2019). *Theory of modeling and simulation: Discrete event and iterative system computational foundations*. San Diego (Calif.): Academic Press.
- Kluever, C. A. (2020). *Dynamic systems modeling, simulation, and control*. Hoboken, N.J: John Wiley & Sons.
- Law, A. M. (2015). *Simulation modeling and analysis*. New York: McGraw-Hill.
- Verschuuren, G. M., & Travise, S. (2016). *100 Excel Simulations: Using Excel to Model Risk, Investments, Genetics, Growth, Gambling and Monte Carlo Analysis*. Holy Macro! Books.
- Grigoryev, I. (2015). *AnyLogic 6 in three days: A quick course in simulation modeling*. Hampton, NJ: AnyLogic North America.
- Dimotikalis, I., Skiadas, C., & Skiadas, C. H. (2011). *Chaos theory: Modeling, simulation and applications: Selected papers from the 3rd Cghaotic Modeling and Simulation International Conference (CHAOS2010), Chania, Crete, Greece, 1-4 June, 2010*. Singapore: World Scientific.
- Velten, K. (2010). *Mathematical modeling and simulation: Introduction for scientists and engineers*. Weinheim: Wiley-VCH.

Module 3: QUEUES

Module Introduction

This module is divided into four (4) units

Unit 1: Simple Theories of Queues
Unit 2: Basic Probability Theories in Queuing
Unit 3: Queuing Models
Unit 4: Queuing Experiments

Unit 1: Simple Theories of Queues

Contents

1.0 Introduction

In this section we will look at a very useful type of simulations called ***queuing system***. We deal with queuing systems all the time in our daily lives. For examples, when you stand in line to cash a cheque at the bank, you are dealing with a queuing system. When you submit a ―batch job‖ (such as a compilation) on a mainframe computer, your job must wait in line until the CPU finishes the jobs scheduled ahead of it. When you make a phone call to reserve an airline ticket and get a recording that says, Thank you for calling. Your call will be answered by the next available operator. ―Please wait‖… you are dealing with a queuing system.

Please Wait Waiting is the critical element of queues.

2.0 Intended Learning Outcomes (ILOs)

By the end of this unit, you should be able to:

- Define queuing theory
- Describe queuing systems; parameters, question and examples of queuing systems
- Describe how Probability theories applied in queuing systems
- Describe some essential Queuing theories
- Describe queuing systems using Kendall Lee notations and Little‘s formula
- Explain the Queue discipline

3.0 Main Content

3.1 Over view of Queuing Systems

The word *queue* comes, via French, from the Latin *cauda*, meaning tail. A **queuing system** is a discrete-event model that uses random numbers to represent the arrival and duration of events. A queuing system is made up of servers and queues of objects to be served on usually *a first-in, first-out structure*. The objective of a queuing system is to utilize the servers (the tellers, checkers, CPU, operators, and so on) as fully as possible while keeping the wait time within a reasonable limit. These goals usually require a compromise between cost and customer satisfaction.

To put this on a personal level, no one likes to stand in line. If there were one checkout counter for each customer in a supermarket, the customers would be delighted. The supermarket, however, would not be in business very long. So a compromise is made: The number of cashiers is kept within the limits set by the store‘s budget, and the average customer is not kept waiting *too* long.

How does a company determine the optimal compromise between the number of servers and the wait time? One way is by experience; the company tries out different numbers of servers and sees how things work out. There are two problems with this approach: It takes too long and it is too expensive. Another way of examining this problem is by using a computer simulation.

Queuing theory is generally considered a branch of operations research because the results are often used when making business decisions about the resources needed to provide service. It is applicable in a wide variety of situations that may be encountered in business, commerce, industry, healthcare, public service and engineering. Applications are frequently encountered in customer service situations as well as transport and telecommunication. Queuing theory is directly applicable to intelligent transportation systems, call centers, PABXs, networks, telecommunications, server queuing , mainframe computer queuing of telecommunications terminals, advanced telecommunications systems, and traffic flow.

The use of queuing allows the systems to queue their customers' requests until free resources become available. This means that if traffic intensity levels exceed available

capacity, customer's calls are not lost; customers instead wait until they can be served. This method is used in queuing customers for the next available operator.

Let us look at how we can solve this problem as a time-driven simulation.
A *time-driven simulation* is one in which the model is viewed at uniform time intervals, say, every minute. To simulate the passing of a unit of time (a minute, for example), we increment a clock. We run the simulation for a predetermined amount of time, say, 100 minutes. (Of course, simulated time usually passes much more quickly than real time; 100 simulated minutes pass in a flash on the computer.)
Think of the simulation as a big loop that executes a set of rules for each value of the clock - from 1 to 100, in our example. Here are the rules that are processed in the loop body:

- *Rule 1*. If a customer arrives, he or she gets in line.
- *Rule 2*. If the teller is free and if there is anyone waiting, the first customer in line leaves the line and advances to the teller‘s window. The service time is set for that customer.
- *Rule 3*. If a customer is at the teller‘s window, the time remaining for that customer to be serviced is decremented.
- *Rule 4*. If there are customers in line, the additional minute that they have remained in the queue (their wait time) is recorded.

The output from the simulation is the average wait time. We calculate this value using the following formula:
Average wait time = total wait time for all customers / number of customers
Given this output, the bank can see whether their customers have an unreasonable wait in a one-teller system. If so, the bank can repeat the simulation with two tellers.

There are still two unanswered questions:

- How do we know if a customer arrived?
- How do we know when a customer has finished being serviced?

We must provide the simulation with information about the arrival times and the service times. These are the variables (parameters) in the simulation. We can never predict exactly when a customer arrives or how long each individual customer takes.
We can, however, make educated guesses, such as a customer arrives about every five minutes and most customers take about three minutes to service.

- How do we know whether or not a job has arrived in this particular clock unit?

The answer is a function of two factors: the number of minutes between arrivals (five in this case) and chance. *Chance?* Queuing models are based on chance? Well, not exactly. Let‘s express the number of minutes between arrivals another way- as the *probability* that a job arrives in any given clock unit. Probabilities range from 0.0 (no chance) to 1.0 (a sure thing). If on the average a new job arrives every five minutes, then the chance of a customer arriving in any given minute is 0.2 (1 chance in 5). Therefore, the probability of a new customer arriving in a particular minute is 1.0 divided by the number of minutes

between arrivals.

- Now what about luck?

In computer terms, luck can be represented by the use of a *random-number generator*. We simulate the arrival of a customer by writing a function that generates a random number between 0 and 1 and apply the following rules.

1. If the random number is between 0.0 and the arrival probability, a job has arrived.
2. If the random number is greater than the arrival probability, no job arrived in this clock unit.

By changing the rate of arrival, we simulate what happens with a one-teller system where each transaction takes about three minutes as more and more cars arrive. We can also have the duration of service time based on probability. For example, we could simulate a situation where 60% of the people require three minutes, 30% of the people require five minutes, and 10% of the people require ten minutes.

It is important at this point to note that simulation doesn‘t give us *the* answer or even *an* answer. Simulation is a technique for trying out ―what if‖ questions. We build the model and run the simulation many times, trying various combinations of the parameters and observing the average wait time. What happens if the cars arrive more quickly? What happens if the service time is reduced by 10%? What happens if we add a second teller?

3.1.1 Queuing Examples:

- waiting to pay in the supermarket
- waiting at the telephone for information
- planes circle before they can land

3.1.2 Queuing questions:

- What is the average waiting time of a customer?
- How many customers are waiting on average?
- How long is the average service time?
- What is the chance that one of the servers has nothing to do?

3.1.3 Parameters of queuing systems

The behaviour of a queuing system is dependent on:

- arrival process (**l** and distribution of interarrival times)
- service process (**m** and distribution of service times)
- number of servers
- capacity of the system

- size of the population for this system

3.2 Queuing theory

Queuing theory is the study of how systems with limited resources distribute those resources to elements waiting in line, and how those elements waiting in line respond.

Queuing theory is a mathematical discipline that studies systems intended for servicing a random flow of requests (the moments at which the requests appear as well as the time for servicing them is usually random).

Queuing theory is the study of the behaviour of queues (waiting lines) and their elements. Queuing theory is a tool for studying several performance parameters of computer systems and is particularly useful in locating the reasons for ―bottlenecks,‖ compromised computer performance caused by too much data waiting to be acted on at a particular phase.

Examples include the distribution of cars on highways (including traffic jams), data through computer networks and phone calls through voice networks. In these examples, Queue size and waiting time can be looked at, or items within queues can be studied and manipulated according to factors such as priority, size, or time of arrival.

The purpose of methods developed in queuing theory is to organize service reasonably so that a given quality is ensured. Queuing theory from this standpoint can be considered as part of operations research.

3.3 Queuing Discipline

A queuing discipline determines the manner in which the exchange handles calls from customers.[It defines the way they will be served, the order in which they are served, and the way in which resources are divided among the customers. Here are details of four queuing disciplines:

First in first out

This principle states that customers are served one at a time and that the customer that has been waiting the longest is served first.

Last in first out

This principle also serves customers one at a time, however the customer with the shortest waiting time will be served first.

Processor sharing

Customers are served equally. Network capacity is shared between customers and they all effectively experience the same delay.

Priority

Customers with high priority are served first.

The queuing strategy are:

– FIFO (first in first out)

– LIFO (last in first out = stack)
– SIRO (service in random order)
– SPT (shortest processing time first)
– PR (priority)

While the particular discipline chosen will likely greatly affect waiting times for particular customers (nobody wants to arrive early at an LCFS discipline), the discipline generally doesn‘t affect important outcomes of the queue itself, since arrivals are constantly receiving service regardless.

Queuing is handled by control processes within exchanges, which can be modelled using state equations. Queuing systems use a particular form of state equations known as a Markov chain that models the system in each state. Incoming traffic to these systems is modelled via a Poisson distribution and is subject to Erlang‘s queuing theory assumptions viz.

- *Pure-chance traffic* – Call arrivals and departures are random and independent events.
- *Statistical equilibrium* – Probabilities within the system do not change.
- *Full availability* – All incoming traffic can be routed to any other customer withinthe network.
- *Congestion is cleared as soon as servers are free.*

Classic queuing theory involves complex calculations to determine waiting time, service time, server utilization and other metrics that are used to measure queuing performance.

3.4 Queuing networks

Networks of queues are systems which contain an arbitrary, but finite, number *m* of queues. Customers, sometimes of different classes, travel through the network and are served at the nodes. The state of a network can be described by a vector , where k_i is the number of customers at queue *i*. In open networks, customers can join and leave the system, whereas in closed networks the total number of customers within the system remains fixed.

3.5 Role of Poisson process and Exponential distributions in Queues

A useful queuing model represents a real-life system with sufficient accuracy and is analytically tractable. A queuing model based on the Poisson process and its companion exponential probability distribution often meets these two requirements.

A Poisson process models random events (such as a customer arrival, a request for action from a web server, or the completion of the actions requested of a web server) as emanating from a memory less process. That is, the length of the time interval from the current time to the occurrence of the next event does not depend upon the time of occurrence of the last event. In the **Poisson** probability distribution, the observer records the number of events that occur in a time interval of fixed length. In the **(negative) exponential** probability distribution, the observer records the length of the time interval

between consecutive events. In both, the underlying physical process is memory less. Models based on the Poisson process often respond to inputs from the environment in a manner that mimics the response of the system being modelled to those same inputs. The analytically tractable models that result yield both information about the system being modeled and the form of their solution. Even a queuing model based on the Poisson process that does a relatively poor job of mimicking detailed system performance can be useful. The fact that such models often give "worst-case" scenario evaluations appeals to system designers who prefer to include a safety factor in their designs. Also, the form of the solution of models based on the Poisson process often provides insight into the form of the solution to a queuing problem whose detailed behaviour is poorly mimicked. As a result, queuing models are frequently modeled as Poisson processes through the use of the exponential distribution.

3.6 Limitations of queuing theory

The assumptions of classical queuing theory may be too restrictive to be able to model real-world situations exactly. The complexity of production lines with product-specific characteristics cannot be handled with those models. Therefore specialized tools have been developed to simulate, analyze, visualize and optimize time dynamic queuing line behaviour.

For example; the mathematical models often assume infinite numbers of customers, infinite queue capacity, or no bounds on inter-arrival or service times, when it is quite apparent that these bounds must exist in reality. Often, although the bounds do exist, they can be safely ignored because the differences between the real-world and theory is not statistically significant, as the probability that such boundary situations might occur is remote compared to the expected normal situation. Furthermore, several studies show the robustness of queuing models outside their assumptions. In other cases the theoretical solution may either prove intractable or insufficiently informative to be useful.

Alternative means of analysis have thus been devised in order to provide some insight into problems that do not fall under the scope of queuing theory, although they are often scenario-specific because they generally consist of computer simulations or analysis of experimental data.

3.7 Kendall-Lee Notations.

The notation for describing the characteristics of a queuing model was first suggested by David G. Kendall in 1953. Kendall's notation introduced an A/B/C queuing notation that can be found in all standard modern works on queuing theory.

The A/B/C notation designates a queuing system having A as interarrival time distribution, B as service time distribution and C as number of servers.

For example, "G/D/1" would indicate a General (may be anything) arrival process, a Deterministic (constant time) service process and a single server. More details on this

notation are given later below in the queuing models.
Since describing all of the characteristics of a queue inevitably becomes very wordy, a much simpler notation (known as Kendall-Lee notation) can be used to describe a system. Kendall-Lee notation gives us six abbreviations for characteristics listed in order separated by slashes. The first and second characteristics describe the arrival and service processes based on their respective probability distributions.
For the first and second characteristics,
M represents an exponential distribution, E
represents an Erlang distribution, and G
represents a general distribution.
The third characteristic gives the number of servers working together at the same time,
also known as the number of parallel servers.
The fourth describes the queue discipline by its given acronym.
The fifth gives the maximum number of number of customers allowed in the system.
The sixth gives the size of the pool of customers that the system can draw from.

For example, M/M/5/FCFS/20/inf could represent a bank with 5 tellers, exponential arrival times, exponential service times, an FCFS queue discipline, a total capacity of 20 customers, and an infinite population pool to draw from.
Kendall‘s notation:
A/B/c/N/K where:
A the interarrival distribution B
the service time distribution
C the number of parallel servers N
the system capacity
K the size of the target group.
Abbreviations for distribution
functions: M - Exponential
D - Constant or deterministic E
or Ek - Erlang
G - General Kendall‘s
notation Example:
***M*/*M*/1**/∞/∞ is a *single-server* system with unlimited queuing capacity and an infinite target group. The arrival intervals and the service times are distributed exponentially If *N* and *K* are infinite, they can be left out of the notation. ***M*/*M*/1**/∞/∞ is abbreviated to ***M*/*M*/1**

3.8 Little’s Queuing Formula.

In many queues, it is useful to determine various waiting times and queue sizes for Particular components of the system in order to make judgments about how the system should be run. Let us define L to be the average number of customers in the queue at any given moment of time assuming that the steady-state has been reached. We can break that down into Lq, the average number of customers waiting in the queue, and Ls, the average number of customers in service. Since customers in the system can only either be in the queue or in service, it goes to show that: L = Lq + Ls.

Likewise, we can define W as the average time a customer spends in the queuing system. Wq is the average amount of time spent in the queue itself and Ws is the average amount of time spent in service. As was the similar case before, W = Wq + Ws. It should be noted that all of the averages in the above definitions are the steady-state averages.
Defining λ as the arrival rate into the system, that is, the number of customers arriving the system per unit of time, it can be shown that:
$L = \lambda W$
$Lq = \lambda Wq$
$Ls = \lambda Ws$
This is known as Little‘s queuing formula.

In the mathematical theory of queues, **Little's result**, **theorem**, **lemma**, **law** or **formula** says:
The long-term average number of customers in a stable system L is equal to the long-term average arrival rate, λ, multiplied by the long-term average time a customer spends in the system, W; or expressed algebraically: $L = \lambda W$.

Although it looks intuitively reasonable, it's a quite remarkable result, as it implies that this behaviour is entirely independent of any of the detailed probability distributions involved, and hence requires no assumptions about the schedule according to which customers arrive or are serviced.

It is also a comparatively recent result; the first proof was published in 1961 by John Little, then at Case Western Reserve University. Handily his result applies to any system, and particularly, it applies to systems within systems. So in a bank, the customer line might be one subsystem, and each of the tellers another subsystem, and Little's result could be applied to each one, as well as the whole thing. The only requirements are that the system is stable and non-preemptive; this rules out transition states such as initial startup or shutdown.

In some cases, it is possible to mathematically relate not only the *average* number in the system to the *average* wait but relate the entire *probability distribution* (and moments) of the number in the system to the wait.

For example - Imagine a small shop with a single counter and an area for browsing, where only one person can be at the counter at a time, and no one leaves without buying something. So the system is roughly**:**
Entrance → Browsing → Counter → Exit

This is a stable system, so the rate at which people enter the store is the rate at which they arrive at the counter and the rate at which they exit as well. We call this the arrival rate. By contrast, an arrival rate exceeding an exit rate would represent an unstable system, and cause the store to overflow eventually.

Little's Law tells us that the average number of customers in the store, *L*, is the arrival rate, λ, times the average time that a customer spends in the store, *W*, or simply:

Assume customers arrive at the rate of 10 per hour and stay an average of 0.5 hour. This means we should find the average number of customers in the store at any time to be 5.

Now suppose the store is considering doing more advertising to raise the arrival rate to 20 per hour. The store must either be prepared to host an average of 10 occupants or must reduce the time each customer spends in the store to 0.25 hour. The store might achieve the latter by ringing up the bill faster or by walking up to customers who seem to be taking their time browsing and saying, "Can I help you?".

We can apply Little's Law to systems within the shop, for example the counter and its queue. Assume we notice that there are on average 2 customers in the queue and at the counter. We know the arrival rate is 10 per hour, so customers must be spending 0.2 hour on average checking out.

We can even apply Little's Law to the counter itself. The average number of people at the counter would be in the range (0, 1), since no more than one person can be at the counter at a time. In that case, the average number of people at the counter is also known as the counter's utilization.

3.9 Queuing Terminology and Notations

Usually, the following notations are used as standards:

- State of the system =number of customers in queuing system
- Queue length= number of customers waiting for service
  =state of the system minus number of customers being served.
- N(t)= number of customers in queuing system at time t (t >=0)
- Pn(t)=probability of exactly n customers in queuing system at time t.
- S= number of servers (parallel service channels) in queuing system.
- λ_n = mean arrival rate (expected number of arrivals per unit time) of new customer when n customers are in system
- μ_0 = mean service rate for overall system (expected number of customers completing service per unit time) when n customers are in system. Note μ_0 represents combined rate at which all busy servers (those serving customers) achieve service completion.

When λ_n is a constant for all n, this constant is denoted by λ. When

Certain notations also are required to describe steady-state result. When a queuing system has recently begun operation, the state of the system will be greatly affected by the initial state and by the time that has since elapsed. The system is said to be in a **transient**

condition. However, after sufficient time has elapsed, the state become independent of the state and the elapsed time (expected under unusual circumstances). The system has now reached **steady state condition**, where the probability distribution of the state of the system remains the same (steady-state or stationary distribution) over time. Queuing theory has tended to focus largely on the stead-state conditions, partially because the transient case is more difficult analytically.

The following notations assume that the system is in a steady-state:
Pn=probability of of exactly n customers in queuing system.
L = expected number of customers in in queuing system.

Lq = expected queue length (excluding customers being served).

W = waiting time in the system (including service time) for each individual customer. W = E(W)
Wq = waiting time in the queue (including service time) for each individual customer. Wq = E(Wq)

Relationships between L, W, Lq and Wq

Assume that λ_n is a constant λ for all n. it has been proved that in a steady-state queuing process,

$L = \lambda W$ (the little's formula).
$Lq = \lambda Wq$
Now assume that the mean service is a constant, $1/\mu$ for all $n \geq 1$. it then follows that: $W = Wq + 1/\mu$

4.0 Self-Assessment Exercise(s)

Give three examples of queuing in different service systems.
How do we know whether or not a job will arrive in a particular time unit?
Define queuing theory and explain the queuing disciplines
State the Little's formula and explain the Kendall's notations
Consider a barbing saloon. Demonstrate that it is a queuing system by describing its components.

5.0 Conclusion

Queuing systems are prevalent throughout society. The goals of queuing systems usually require a compromise between cost and customer satisfaction. The adequacy of these systems can have an important effect on the quality of life and productivity. In this section we looked at ways of utilizing queuing to optimize service delivery in order keep the waiting time, cost and customer satisfaction within reasonable limits.

6.0 Summary

In this section we:

- Define a **queuing system** as a discrete-event model that uses random numbers to represent the arrival and duration of events and **Queuing theory** as a mathematical discipline that studies systems intended for servicing a random flow of requests know how to construct queue systems; parameters, question and examples of queuing systems.
- We in the overview section look at:
    - The Parameters, for Construction of queuing system,
    - The questions and necessary questions in queuing construction
    - The network queues, the role of Poisson and exponential distributions
    - The limitations of queuing theory
- Described queuing systems using Kendall Lee notations and Little‘s formula

7.0 Further Readings

- Bhat, U. N. (2015). *An introduction to queueing theory: Modeling and analysis in applications*. Boston, MA: Birkhäuser.
- Medhi, J. (2019). *Introduction to queueing systems and applications*. London, UK: New Academic Science.
- Trivedi, K. S. (2016). *Probability and statistics with reliability, queuing, and computer science applications*. Hoboken, NJ: Wiley.
- Khinchin, A. I. (2013). *Mathematical methods in the theory of queuing*. Mineola, NY: Dover.
- Thomopoulos, N. T. (2012). *Fundamentals of queuing systems: Statistical methods for analyzing queuing models*. New York: Springer.
- Dudin, A., Klimenok, V. I., & Višnevskij, V. (2020). *The theory of queuing systems with correlated flows*. Cham, Switzerland: Springer.
- Dudin, A., Klimenok, V. I., & Višnevskij, V. (2020). *The theory of queuing systems with correlated flows*. Cham, Switzerland: Springer.

Unit 2: Basics Probability Theories in Queuing Systems

Contents

1.0 Introduction

To begin understanding queues, we must first have some knowledge of probability theory. In particular, we will review the exponential and Poisson probability distributions.

2.0 Intended Learning Outcomes (ILOs)

By the end of this unit, the reader should be able to

- State the relationship between Exponential and Poisson Probability Distributions
- Define the Input and Output parameters of a typical queuing system
- Describe the Steady-State probability for queues

3.0 Main Content

3.1 Exponential And Poisson Probability Distributions.

The exponential distribution with parameter λ is given by $\lambda e^{-\lambda t}$ for $t \geq 0$. If T is a random variable that represents interarrival times with the exponential distribution, then:

$$\mathbf{P(T \leq t) = 1 - e^{-\lambda t} \text{ and } P(T > t) = e^{-\lambda t}}$$

This distribution lends itself well to modelling customer interarrival times or service times for a number of reasons. The first is the fact that the exponential function is a strictly decreasing function of t. This means that after an arrival has occurred, the amount of waiting time until the next arrival is more likely to be small than large.

An important property of the exponential distribution is what is known as the no-memory property. The no-memory property suggests that the time until the next arrival will never depend on how much time has already passed. This makes intuitive sense for a model where we‘re measuring customer arrivals because the customers‘ actions are clearly independent of one another.

It‘s also useful to note the exponential distribution‘s relation to the Poisson distribution. *Poisson distribution* is a "discrete probability distribution. It expresses the probability of a number of events occurring in a fixed time if these events occur with a known average rate, and are independent of the time since the last event". Such events are said to be memoryless.

Most queuing systems' characteristics such as arrival and departure processes are described by a Poisson distribution. Assuming that arrivals and departures are random and independent i.e. they exhibit pure-chance property; arrivals are described by a Poisson random variable or Poisson random distribution as shown by equation below.

The probability that there are exactly k occurrences (k being a non-negative integer, k = 0, 1, 2, ...) is the Poisson distribution with parameter λ is given by:

$$f(k;\lambda) = \frac{e^{-\lambda}\lambda^{k}}{k!}, \text{……..(1)}$$

Where: e is the base of the natural logarithm (e = 2.71828...), k! is the factorial of k, and λ is a positive real number, equal to the expected number of occurrences that occur during the given interval. A number of textbooks e.g. Schaum‘s Outline statistics 3rd edition provides the e- λ for various values of λ or by using logarithms.

For instance, if the events occur on average every 4 minutes, and you are interested in the number of events occurring in a 10 minute interval, you would use as model a Poisson distribution with λ = 2.5.

Some properties of Poisson distribution Mean = □=λ
Variance = ζ^2 = □
Standard deviation σ = □λ
Moment coefficient of skewness □3= 1/□λ Moment coefficient of = □4= 3+1/□

Example 1

If the average rate of telephone calls received at an exchange of 8 lines is 6 per minute. Find the probability that a caller is unable to make a connection if this is defined to occur when all lines are engaged within a minute of the time of the call.

Solution

We first need to make an assumption that the overall rate of calls is constant, then we can use equation (1) as follows: Since our time unit is 1 minute, then λ = 6

Using the equation:

$$f(k;\lambda) = \frac{e^{-\lambda}\lambda^k}{k!},$$

Which leads after substitution to:

$$\frac{e^{-6}6^k}{k!}$$

The probability of not being able to make a call occurs only when there are at least 9 calls in any interval of a minute.

$$\sum_{k=9}^{\infty}\frac{e^{-6}6^k}{k!} = 1 - \sum_{k=0}^{8}\frac{6^k e^{-6}}{k!}$$

So, this leads to p(k+1) equals to:

$$\frac{\lambda}{k+1}p(k)$$

Solving this bit by bit with k from 0 to 8, we get p(k+1) = **0.8546.**

With these distributions in mind, we can begin defining the input and output processes of a basic queuing system, from which we can start developing the model further.

3.2 The Input Process.

To begin modelling the input process, we define ti as the time when the ith customer arrives. For all i ≥ 1, we define Ti = $t_{i+1} - t_i$ to be the ith inter-arrival time. We also assume that all T_i's are independent, continuous random variables, which we represent by the random variable A with probability density a(t).

Typically, A is chosen to have an exponential probability distribution with parameter λ defined as the arrival rate, that is to say, $a(t) = \lambda e^{-\lambda t}$.

It is easy to show that if A has an exponential distribution, then for all nonnegative values of t and h,

$$P(A > t + h | A \geq t) = P(A > h)$$

This is an important result because it reflects the no-memory property of the exponential distribution, which is an important property to take note of if we're modelling interarrival times.

Another distribution the can be used to model interarrival times (if the exponential distribution does not seem to be appropriate) is the Erlang distribution. An Erlang distribution is a continuous random variable whose density function relies on a rate parameter R and a shape parameter k. The Erlang probability density function is:

$$f(t) = \frac{R(Rt)^{k-1}e^{-Rt}}{(k-1)!}$$

3.3 The Output Process

Much like the input process, we start analysis of the output process by assuming that service times of different customers are independent random variables represented by the random variable S with probability density $s(t) = \mu e^{-\mu t}$. We also define μ as the service rate, with units of customers per hour. Ideally, the output process can also be modeled as an exponential random variable, as it makes calculation much simpler.

Imagine an example where four customers are at a bank with three tellers with exponentially distributed service times. Three of them receive service immediately, while the fourth has to wait for one position to clear.

What is the probability that the fourth customer will be the final one to complete service?

Due to the no-memory property of the exponential distribution, when the fourth customer finally steps up to a teller, all three remaining customers have an equal chance of finishing their service last, as the service time in this situation is not governed by how long they have already been served. Thus, the answer to the question is 1/3.

Unfortunately, the exponential distribution does not always represent service times accurately. For a service that requires many different phases of service (for example, scanning groceries, paying for groceries, and bagging the groceries), an Erlang distribution can be used with the parameter k equal to the number of different phases of service.

3.4 Output variables

The following are the definitions of the expected output variables

- Utilisation rate ρ (server utilization, percentage of the time that a server is busy, where c=the number of parallel servers)
- Probability of n customers in the system *Pn*
- Average number of customers in the system *L* (service and queue)
- Average number of customers in the queue *Lq*
- Average time spent by a customer in the system *w*(service and queue)
- Average time spent by a customer in the queue *wq*

3.5 Birth-Death Processes

We define the number of people located in a queuing system, either waiting in line or in service, to be the state of the system at time t. At t = 0, the state of the system is going to be equal to the number of people initially in the system. The initial state of the system is noteworthy because it clearly affects the state at some future t. Knowing this, we can define Pij(t) as the probability that the state at time t will be j, given that the state at t = 0 was i. For a large t, Pij(t) will actually become independent of i and approach a limit Πj. This limit is

known as the steady-state of state j.

Generally, if one is looking at the steady-state probability of j, it is incredibly difficult to determine the steps of arrivals and services that led up to the steady state. Likewise, starting from a small t, it is also very difficult to determine when exactly a system will reach its steady state, if it exists. Thus, for simplicity‘s sake, when we study a queuing system, we begin by assuming that the steady-state has already been reached.

A **birth-death process** is a process wherein the system‘s state at any t is a nonnegative integer. The variable λj is known as the birth rate at state j and symbolizes the probability of an arrival occurring over a period of time. The variable μj is known as the death rate at state j and symbolizes the probability that a completion of service occurs over a period of time.

Thus, *births* and *deaths* are synonymous with arrivals and service completions respectively. A birth increases the state by one while a death decreases the state by one. We note that μ0 = 0, since it must not be possible to enter a negative state. Also, in order to officially be considered a birth-death process, birth and deaths must be independent of each other.
The probability that a birth will occur between t and t + Δt is λjΔt, and such a birth will increase the state from j to j + 1. The probability that a death will occur between t and t +Δt is μjΔt, and such a birth will decrease the state from j to j − 1.

3.6 Steady-state Probabilities

In order to determine the steady-state probability Πj, we have to find a relation between Pij(t+Δt) and Pij(t) for a reasonably sized t. We begin by categorizing the potential states at time t from which a system could end up at state j at time t + Δt. In order to achieve this, the state at time t must be j, j − 1, j + 1, or some other value. Then, to calculate Πj, all we have to do is add up the probabilities of the system ending at state j for each of these beginning categories.

Note that:
To reach state j from state j − 1, we need one birth to occur between t and Δt. To reach j from j + 1, we need one death.
To remain at j, we need no births or deaths to occur.
To reach j from any other state we will need multiple births or deaths.

Since we will be eventually letting t approach zero, we find that it is impossible to reach state j from these other states because births and deaths are independent of each other, and won‘t occur simultaneously. Hence, we only need to sum the probabilities of these first three situations occurring.

That will give us:

$$P_{ij}(t+\Delta t) = [P_{i,j-1}(t)(\lambda_{j-1}\Delta t)] + [P_{i,j+1}(t)(\mu_{j+1}\Delta t)] + [P_{ij}(t)(1-\mu_j\Delta t - \lambda_j\Delta t)]$$

which can be rewritten as:

$$P'_{ij}(t) = \lim_{\Delta t \to \infty} \frac{P_{ij}(t+\Delta t) - P_{ij}}{\Delta t} = \lambda_{j-1} P_{i,j-1}(t) + \mu_{j+1} P_{i,j+1}(t) - P_{ij}(t)\mu_j - P_{ij}(t)\lambda_j$$

Since we're trying to calculate steady-state probabilities, it is appropriate to allow t to approach infinity, at which point Pij(t) can be thought of as a constant.

Then $P'_{ij}(t) = 0$ Defining the steady-state probability $\pi_j = \lim_{t \to \infty} P_{ij}(t)$,

we can substitute further.

$$\lambda_{j-1}\pi_{j-1} + \mu_{j+1}\pi_{j+1} - \pi_j\mu_j - \pi_j\lambda_j = 0$$

$$\lambda_{j-1}\pi_{j-1} + \mu_{j+1}\pi_{j+1} = \pi_j(\lambda_j + \mu_j) \text{ for } j = 1, 2, \ldots$$

$$\mu_1\pi_1 = \lambda_0\pi_0 \text{ for } j = 0$$

These results are known as the *flow balance equations*. You may notice that they suggest that the rate at which transitions occur into a particular state equal the rate at which transitions occur out of the same state. At this point, each steady-state probability can be determined by substituting in probabilities from lower states.

Starting with:

$$\pi_1 = \frac{\pi_0\lambda_0}{\mu_1},$$

we can get the general equation:

$$\pi_j = \pi_0 c_j$$

Where

$$c = \frac{\lambda_0\lambda_1 \ldots \lambda_{j-1}}{\mu_1\mu_2 \ldots \mu_j}$$

4.0 Self-Assessment Exercise(s)

A Poisson distribution is given by $p(x) = (0.72)^x e^{-0.72}/x!$ Find p(0) and p(2).

What do you understand by birth-death process in queues?

If 10% of the tools produced in a manufacturing process is defective. Find the Poisson approximation to the binomial distribution.

5.0 Conclusion

This unit reviewed an important function; Poisson distribution and gives an example of its application.

6.0 Summary

In this unit we discussed Probability theories applied in queuing systems:

- Looked at Poisson and Exponential distributions in queuing system

- Defined the Input, Output parameters of queues
- Discuss the Birth-Death process; *births* and *deaths* are synonymous with arrivals and service completions respectively. A birth increases the state byone while a death decreases the state by one.
- Defined the formula for calculating Steady- State probability

7.0 Further Readings

- Bhat, U. N. (2015). *An introduction to queueing theory: Modeling and analysis in applications*. Boston, MA: Birkhäuser.
- Medhi, J. (2019). *Introduction to queueing systems and applications*. London, UK: New Academic Science.
- Trivedi, K. S. (2016). *Probability and statistics with reliability, queuing, and computer science applications*. Hoboken, NJ: Wiley.
- Khinchin, A. I. (2013). *Mathematical methods in the theory of queuing*. Mineola, NY: Dover.
- Thomopoulos, N. T. (2012). *Fundamentals of queuing systems: Statistical methods for analyzing queuing models*. New York: Springer.
- Dudin, A., Klimenok, V. I., & Višnevskij, V. (2020). *The theory of queuing systems with correlated flows*. Cham, Switzerland: Springer.

- Dudin, A., Klimenok, V. I., & Višnevskij, V. (2020). *The theory of queuing systems with correlated flows*. Cham, Switzerland: Springer.

Unit 3: Queuing Models

Contents

1.0 Introduction

With the foundation laid for the study of important characteristics of queuing systems, we will in this unit begin to analyze particular systems themselves.

2.0 Intended Learning Outcomes (ILOs)

After reading this unit you should be able to:

- Define queuing model
- Describe the construction of models
- State basic characteristics of queues
- Describe the various queue models

3.0 Main Content

3.1 Queuing Model

In queuing theory, a **queuing model** is used to approximate a real queuing situation or system, so the queuing behaviour can be analysed mathematically. Queuing models allow a number of useful steady state performance measures to be determined, including:

- the average number in the queue, or the system,
- the average time spent in the queue, or the system,
- the statistical distribution of those numbers or times,
- the probability the queue is full, or empty, and
- the probability of finding the system in a particular state

These performance measures are important as issues or problems caused by queuing situations are often related to customer dissatisfaction with service or may be the root cause

of economic losses in a business. Analysis of the relevant queuing models allows the cause of queuing issues to be identified and the impact of proposed changes to be assessed.

3.1.1 Construction

Queuing models are generally constructed to represent the steady state of a queuing system, that is, the typical, long run or average state of the system. As a consequence, these are stochastic models that represent the probability that a queuing system will be found in a particular configuration or state.

A general procedure for constructing and analysing such queuing models is:

1. Identify the parameters of the system, such as the arrival rate, service time, queue capacity, and perhaps draw a diagram of the system.
2. Identify the system states. (A state will generally represent the integer number of customers, people, jobs, calls, messages, etc. in the system and may or may not be limited.)
3. Draw a state transition diagram that represents the possible system states and identify the rates to enter and leave each state. This diagram is a representation of a Markov chain.
4. Because the state transition diagram represents the steady state situation between state there is a balanced flow between states so the probabilities of being in adjacent states can be related mathematically in terms of the arrival and service rates and state probabilities.
5. Express all the state probabilities in terms of the empty state probability, using the inter-state transition relationships.
6. Determine the empty state probability by using the fact that all state probabilities always sum to 1.

Whereas specific problems that have small finite state models can often be analysed numerically, analysis of more general models, using calculus, yields useful formulae that can be applied to whole classes of problems.

3.1.2 Parameters

In constructing a queuing model, we must know the following four things:

1. The number of events and how they affect the system in order to determine the rules of entity interaction
2. The number of servers
3. The distribution of arrival times in order to determine if an entity enters the system
4. The expected service time in order to determine the duration of an event

Simulation uses these characteristics to predict the average wait time. The number of servers, the distribution of arrival times, and the duration of service can be changed. The average wait times are then examined to determine what a reasonable compromise would be.

Example
Consider the case of a drive-in bank with one teller. How long does the average car have to wait? If business gets better and cars start to arrive more frequently, what would be the effect on the average wait time? When would the bank need to open a second drive-in window?

This problem has the characteristics of a queuing model. The entities are a *server* (the teller), the *objects being served* (customers in cars), and a queue to hold the objects waiting to be served (customers in cars). The *average wait time* is what we are interested in observing. The events in this system are the arrivals and the departures of customers.

We first categorize queues into single and multiple servers and their service delineations.

3.2 Single-Server Queue

Single-server queues are, perhaps, the most commonly encountered queuing situation in real life. One encounters a queue with a single server in many situations, including business (e.g. sales clerk), industry (e.g. a production line) and transport (e.g. queues that the customer can select from.). Consequently, being able to model and analyse a single server queue's behaviour is a particularly useful thing to do.

3.2.1 The Poisson arrivals and service

M/M/1/∞/∞ represents a single server that has unlimited queue capacity and infinite calling population, both arrivals and service are Poisson (or random) processes, meaning the statistical distribution of both the inter-arrival times and the service times follow the exponential distribution. Because of the mathematical nature of the exponential distribution, a number of quite simple relationships are able to be derived for several performance measures based on knowing the arrival rate and service rate.

3.2.2 The Poisson arrivals and general service

M/G/1/∞/∞ represents a single server that has unlimited queue capacity and infinite calling population, while the arrival is still Poisson process, meaning the statistical distribution of the inter-arrival times still follow the exponential distribution, the distribution of the service time does not. The distribution of the service time may follow any general statistical distribution, not just exponential. Relationships are still able to be derived for a (limited) number of performance measures if one knows the arrival rate and the mean and variance of the service rate. However the derivations are generally more complex and difficult.

A number of special cases of M/G/1 provide specific solutions that give broad insights into the best model to choose for specific queuing situations because they permit the comparison of those solutions to the performance of an M/M/1 model.

3.2.3 The M/M/1/GD/∞/∞ Queuing System

An M/M/1/GD/∞/∞ system has exponential interarrival times, exponential service times, and one server. This system can be modelled as a birth-death process where

λj = λ for (j = 0, 1, 2...)

μ0 = 0

μj = μ for (j = 1, 2, 3...)

Substituting this in to the equation for the steady-state probability, we get

$$\pi_j = \frac{\lambda^j \pi_0}{\mu^j}$$

We will define p = λ/μ as the traffic intensity of the system, which is a ratio of the arrival and service rates. Knowing that the sum of all of the steady state probabilities is equal to one, we get:

$$\pi_0(1 + p + p^2 + ... + p^j) = 1$$

If we assume 0 ≤p ≤1 and let the sum S = (1 + p + p2 + ... + pj), then: S = 1/1-p and Π0 = 1 − p. This yield:

$$\pi_j = p^j(1 - p)$$

as the steady-state probability of state j.

Note that if p≥1, S approaches infinity, and thus no steady state can exist. Intuitively, if p ≥1, then it must be that λ ≥ μ, and if the arrival rate is greater than the service rate, then the state of the system will grow without end.

With the steady-state probability for this system calculated, we can now solve for L. If L is the average number of customers present in this system, we can represent it by the formula:

$$L = \sum_{j=0}^{\infty} j\pi_j = (1 - p)\sum_{j=0}^{\infty} jp^j$$

Let $S = \sum_{j=0}^{\infty} = p + 2p^2 + 3p^3 + ...$ Then $pS = p^2 + 2p^3 + 3p^4 + ...$ If we subtract, we get

$$S - pS = p + p^2 + p^3 ... = \frac{p}{1 - p}$$

And $S = \frac{p}{(1-p)^2}$. Substituting this into the equation for L will get us

$$L = (1 - p)\frac{p}{(1 - p)^2} = \frac{p}{1 - p} = \frac{\lambda}{\mu - \lambda}$$

To solve for Ls, we have to determine how many customers are in service at any given moment. In this particular system, there will always be one customer in service except for when there are no customers in the system. Thus, this can be calculated as

$$L_q = 0\pi_0 + 1(\pi_1 + \pi_2 + \pi_3 + ...) = 1 - \pi_0 = 1 - (1 - p) = p$$

From here, L_q is an easy calculation.

$$L_q = L - L_s = \frac{p}{1-p} - p = \frac{p^2}{1-p}$$

Using Little's queuing formula, we can also solve for W, Ws, and Wq by dividing each of the corresponding L values by λ.

3.2.4 The M/M/1/GD/c/∞ Queuing System

An M/M/1/GD/c/∞ queuing system has exponential interarrival and service times, with rates λ and μ respectively.

This system is very similar to the previous system, except that whenever c customers are present in the system, all additional arrivals are excluded from entering, and are thereafter no longer considered. For example, if a customer were to walk up to a fast food restaurant and see that the lines were too long for him to wait there, he would go to another restaurant instead.

A system like this can be modelled as a birth-death process with these parameters:
λj = λ for j = 0, 1, ..., c − 1
λc = 0
μ0 = 0
μj = μ for j = 1, 2, ..., c
The restriction λc = 0 is what sets this apart from the previous system. It makes it so that no state greater than c can ever be reached. Because of this restriction, a steady state will always exist. This is because even if λ>=μ, there will never be more than c customers in the system.

Looking at formulas derived from the study of birth-death processes and once again letting p =λ/μ, we can derive the following steady-state probabilities:

$$\pi_0 = \frac{1-p}{1-p^{c+1}}$$

$$\pi_j = p^j \pi_0 \text{ for } j = 1, 2, ..., c$$

$$\pi_j = 0 \text{ for } j = c+1, c+2, ..., \infty$$

A formula for L can be found in a similar fashion, but is omitted because of the messy calculations. The technique is similar to the one used in the previous section.
Calculating W is another issue. This is because in Little's queuing formula, λ represents the arrival rate, but in this system, not all of the customers who arrive will join the queue. In fact, λΠc arrivals will arrive, but leave the system. Thus, only λ− λΠc = λ(1 − Πc) arrivals will ever enter the system. Substituting this into Little's queuing formula gives us:

$$W = \frac{L}{\lambda(1 - \pi_c)}$$

3.3 The Multiple and Infinite Server Systems

Multiple-servers queue - Multiple (identical)-servers queue situations are frequently encountered in telecommunications or a customer service environment. When modelling these situations care is needed to ensure that it is a multiple servers queue, not a network of single server queues, because results may differ depending on how the queuing model behaves.

One observational insight provided by comparing queuing models is that a single queue with multiple servers performs better than each server having their own queue and that a single large pool of servers performs better than two or more smaller pools, even though there are the same total number of servers in the system.

Example 1

Consider a system having 8 input lines, single queue and 8 servers. The output line has a capacity of 64 kbit/s. Considering the arrival rate at each input as 2-packets/s. So, the total arrival rate is 16-packets/s. With an average of 2000 bits per packet, the service rate is 64 kbit/s/2000b = 32 packets/s. Hence, the average response time of the system is $1/(\mu - \lambda) = 1/(32 - 16) = 0.0625$ sec.

Example 2

Consider a second system with 8 queues, one for each server. Each of the 8 output lines has a capacity of 8 kbit/s. The calculation yields the response time as $1/(\mu - \lambda) = 1/(4 - 2)$ = 0.5 sec. And the average waiting time in the queue in the first case is $\rho/(1 - \rho)\mu = 0.03125$, while in the second case is 0.25.

Infinitely many servers - While never exactly encountered in reality, an *infinite-servers* (e.g. M/M/∞) model is a convenient theoretical model for situations that involve storage or delay, such as parking lots, warehouses and even atomic transitions. In these models there is no queue, as such; instead each arriving *customer* receives service. When viewed from the outside, the model appears to delay or store each *customer* for some time.

3.3.1 The M/M/s/GD/∞/∞ Queuing System.

An **M/M/s/GD/**∞/∞ queuing system, like the previous system we looked at, has exponential interarrival and service times, with rates λ and μ. What sets this system apart is that there are s servers willing to serve from a single line of customers, like perhaps one would find in a bank. If j<= s customers are present in the system, then every customer is being served. If j>s customers are in the system, then s customers are being served and the remaining $j - s$ customers are waiting in the line.

To model this as a birth-death system, we have to observe that the death rate is dependent on how many servers are actually being used. If each server completes service with a rate of μ, then the actual death rate is μ times the number of customers actually being served. Parameters for this system are as follows:

$$\lambda_j = \lambda \text{ for } j = 0, 1, ..., \infty$$

$$\mu_j = j\mu \text{ for } j = 0, 1, ..., s$$

$$\mu_j = s\mu \text{ for } j = s+1, s+2, ..., \infty$$

In solving the steady-state probabilities, we will define p = λ /su. Notice that this definition also applies to the other systems we looked at, since in the other two systems, s
= 1. The steady-state probabilities can be found in this system in the same manner as for other systems by using the flow balance equations.
I will also omit these particular steady-state equations because they are rather cumbersome.

3.3.2 The M/G/s/GD/s Queuing System.

Another reasonable model of a queue is one where if a customer arrives and sees all of the servers busy, then the customer exits the system completely without receiving service. In this case, no actual queue is ever formed, and we say that the blocked customers have been cleared. Since no queue is ever formed, Lq = Wq = 0. If λ is the arrival rate and 1/μ is the mean service time, then W = Ws = 1/μ.
In this system, arrivals are turned away whenever s customers are present, so
Πs is equal to the fraction of all arrivals who are turned away by the system. This means that an average of λΠs arrivals per unit of time will never enter the system, and thus, λ(1−Πs) arrivals per unit of time will actually enter the system. This leads us to the conclusion based on Little's queuing formula that L = Ls = λ(1−Πs)/μ .

4.0 Self-Assessment Exercise(s)

Answer the following questions:

1. What are the necessarily parameters of queue models
2. Differentiate between M/M/s/GD/∞/∞ **and** M/M/1/GD/c/∞
3. The jobs to be performed on a particular machine arrive according to a Poisson input process with mean rate of 2 per hour. Suppose that the machine breaks down and will require 1 hour to be repaired. What is the probability that the number of new jobs that will arrive during this time is a. 0, b. 2, c. 5 or more?

5.0 Conclusion

The goals of queuing systems and queuing models, usually require a compromise between cost and customer satisfaction. In this section we looked at ways of utilizing queuing to optimize service delivery in order keep the waiting time, cost and customer satisfaction within reasonable limits.
The applications of queuing theory extend well beyond waiting in line. It may take some creative thinking, but if there is any sort of scenario where time passes before a particular

event occurs, there is probably some way to develop it into a queuing model. Queues are so commonplace in society that it is highly worthwhile to study them, even if only to shave a few seconds off one‘s wait in the checkout line

6.0 Summary

We looked at:

- The procedure for construction of queue models
- some Basic Queuing models for:
- Single-server with Poisson arrivals and service and general service
- Multiple and Infinite Server Systems

7.0 Further Readings

- Bhat, U. N. (2015). *An introduction to queueing theory: Modeling and analysis in applications*. Boston, MA: Birkhäuser.
- Medhi, J. (2019). *Introduction to queueing systems and applications*. London, UK: New Academic Science.
- Trivedi, K. S. (2016). *Probability and statistics with reliability, queuing, and computer science applications*. Hoboken, NJ: Wiley.
- Khinchin, A. I. (2013). *Mathematical methods in the theory of queuing*. Mineola, NY: Dover.
- Thomopoulos, N. T. (2012). *Fundamentals of queuing systems: Statistical methods for analyzing queuing models*. New York: Springer.
- Dudin, A., Klimenok, V. I., & Višnevskij, V. (2020). *The theory of queuing systems with correlated flows*. Cham, Switzerland: Springer.

- Dudin, A., Klimenok, V. I., & Višnevskij, V. (2020). *The theory of queuing systems with correlated flows*. Cham, Switzerland: Springer.

Unit 4: Queuing Experiments

Contents

1.0 Introduction

This unit simply is about applications of queues in different daily activities and how the application enable us establish/use observed queuing systems relationships to make informed decisions to reduce costs.

2.0 Intended Learning Outcomes (ILOs)

By the end of this unit the reader should be able to:

- Perform experiments of queues as applied in:
- Car washing,
- Sales calls.
- Goods production and
- Translate the experiments into Simulation flowcharts and programs

3.1 Car Wash Experiment

A garage owner has installed an automatic car washing machine, which services cars one at a time. As his services get more popular, the owner is faced with the dilemma. As more customers use the car wash, the average waiting time tends to increase and service becomes less attractive. The owner may react in several ways: do nothing and let the customers flow stabilize at a lower level or build another car wash which will keep the present customers happy and probably attract more. Since building a car wash entails considerable expense, the later is not a decision to be taken lightly and demands some investigation.

Suppose the car wash services cars one at a time and each service takes 10 minutes. When a car arrives, it goes into the car wash if it is idle, otherwise, it must wait in the queue. As long as cars are waiting, the car wash is in continuous operation serving on the first come first

served principle. If the arrival times of cars have been recorded for one day, then a very simple model is capable of reproducing the essential aspects of the system on that day,. This model can generate data which describe the performance of the installation, such as use of equipments, average no of waiting cars, time spent by each car etc.

The operation of the model consists of two critical events:
Arrival of a Car – The arrival time is noted. If the car wash is idle a 10 minutes service starts at once otherwise the car goes to the queue.
End of Service – the elapse time for that car is noted and car leaves the system. If cars are waiting the first car in the queue is served and another 10 min service is started otherwise the machine becomes idle.
The car wash starts in the idle state and waits for the first arrival.
In other to understand the model better, we give a trace of the history of the model over a short period when car arrive at the time 6, 10, 13, 28, 42, 43, 48 (in minutes). The state of the model only changes at the two critical moments noted above – when a car arrives and when a service is finished. Accordingly we need only trace at these times.

CAR WASH TRACE

Time	Events
0	the car wash awaits the first arrival
6	Car 1 arrives and goes into the car wash
10	Car 2 arrives and must wait
13	Car 3 arrives and must wait
16	Car 1 leaves the system.
	Car 2 enters the car wash.
26	Car 2 leaves the system.
	Car 3 enters the car wash.
28	Car 4 arrives and must wait.
36	Car 3 leaves
	Car 4 enters the car wash.
42	Car 5 arrives and must wait.
43	Car 6 arrives and must wait
46	Car 4 leaves the system
	Car 5 enters the car wash
48	Car 7 arrives and must wait.
56	Car 5 leaves the system
	Car 6 enters
66	Car 6 leaves the system
	Car 7 enters the car wash
76	Car 7 leaves the system.

The data recorded from the trace for the cars are:

Car Number	1	2	3	4	5	6	7
Arrival time	6	10	13	28	42	43	48
Departed at	16	26	36	46	56	66	76
Elapse time	10	16	23	18	14	23	28

Average time spent = total-elapse-time / 7 = 76/7=18.8mins
In the real world the owner would compare the behaviour of the model with actual system behaviour. If the model does not approximate the actual system the model structure must be changed or refined. When a proper model has been developed, it may be extended to predict hypothetical system behaviour such as estimating the behaviour if another car wash is installed.

Let us demonstrate the trace using two car washes and run this model with two car washes operating upon the same data. The extra (unfair) decision rule is that car wash one is used if both car washes are idle.

Trace For Two Car Washes

TIME	**EVENT**
0	Car washes await first arrival.
6	Car 1 arrival and goes into wash 1
10	Car 2 arrival and goes into wash 2
13	Car 3 arrives and waits
16	Car 1 leaves wash 1 and Car 3 enter wash 1
20	Car 2 leaves wash 2 and Wash 2 idle
26	Car 3 leaves wash 1, Car wash 1 & 2 idle
28	Car 4 arrives into wash 1 and Wash 2 still idle
38	Car 4 leaves wash 1 and Car wash 1 & 2 now idle
42	Car 5 arrives into wash 1, wash 2 still idle.
43	Car 6 arrives into wash 2
48	Car 7 arrives and waits
52	Car 5 leaves wash 1, and Car 7 enters wash 1.
53	Car 6 leaves wash 2.
62	Car 7 leaves wash 1.

The data recorded for the 7 cars for the two car washes are:

Car Number	1	2	3	4	5	6	7
Arrival time	6	10	13	28	42	43	48
Departed at	16	20	26	38	52	53	62
Elapse time	10	10	13	10	10	10	14
Serviced by	1	2	1	1	1	2	1

Average time spent = 77/7 = 11mins Car wash 2 was idle for 22mins
Car 3 waited for 3mins while car waited for 4mins.

Thus the average waiting time prior to service is reduced from nearly 9mins to 1min by increasing the number of washes from 1 to 2.
Clearly no management would base its decisions on such a small sample. To arrive to proper estimates it is necessary to simulate the behaviour of the system over several days.

3.2 Salesman Calls

Example 2

A salesman averages one call each day and his results show that 50% of the call leads to a sale. Simulate two sequences of results for the salesman covering a period of 10 days. Use the following sequence of random numbers: 8, 4, 3, 7, 9, 0, 6, 1, 5, 6 and 3, 6, 6, 7, 1, 0, 0, 8, 2, 3.

Solution

The solution has two parts. The first part specifies the rules and the second applies them. The rules must link the random numbers so that success has probability of 0.5 and failure has probability of 0.5. A simple categorization of the digits (0,1,2,3,4,5,6,7,8,9) is to have two subsets{0,1,2,3,4} and {5,6,7,8,9} where each contains five elements. We associate success with first group and failure with the second group. Note that this categorization is far from unique. Any division of the ten digits into two groups of five would be good. So one might as well have used [0,2,4,6,8] and [1,3,5,7,9] for success and failure respectively.

It is usually very useful to draw a flowchart of the simulation process as shown below:

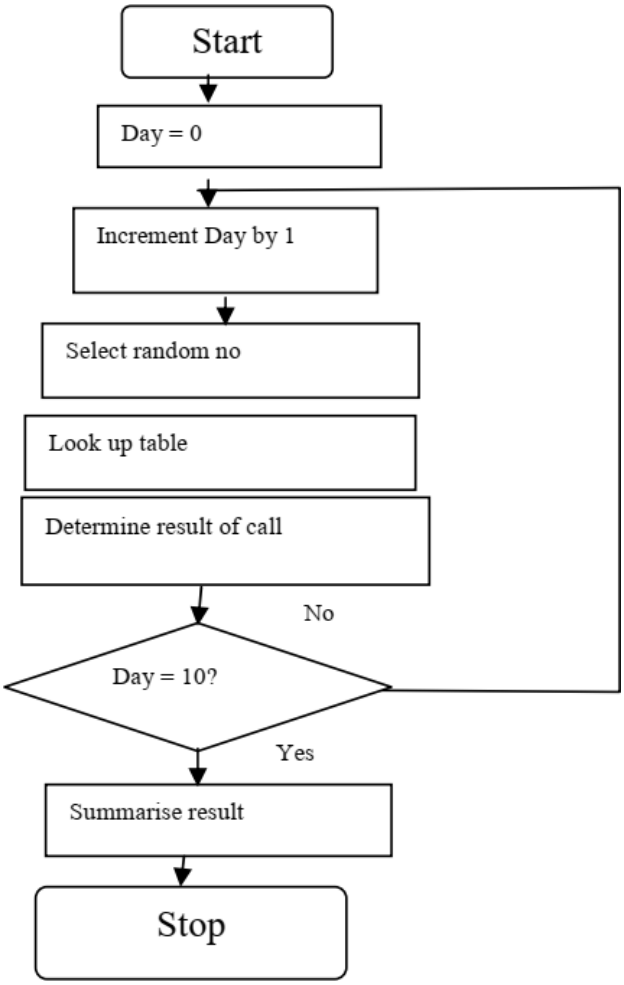


Fig. 1: Flowchart showing the activities of the salesman

It is convenient to summarise the data in what is referred to as lookup table:

Result of call	**Probability**	**Random numbers**
Success(s)	0.5	0,1,2,3,4
Failure(f)	0.5	5,6,7,8,9

The application of the rule for two 10-day periods are shown below.

First 10-day period

Day	1	2	3	4	5	6	7	8	9	10
Ran No.	8	4	3	7	9	0	6	1	5	6
R	f	s	s	f	f	s	f	S	f	f

esult of call

Second 10-day period

Day	1	2	3	4	5	6	7	8	9	10
Ran No.	3	6	6	7	1	0	0	8	2	3
Result of call	s	f	f	f	s	s	s	F	s	s

Having carried out a simulation, it is necessary to calculate appropriate statistics to summarize the situation being studied.

3.3 Sales Man BASIC Program

Example 3:- Here let us write a QBASIC program to simulate the result of 10 calls made by a salesman, given that each call has 50% chance of success and 50% chance of failure.

Solution

REM X is set to a random value between 0 and 1
REM For X between 0 and 0.5 the salesman has a successful call REM For X between 0 and 0.5 the salesman has a unsuccessful call REM A successful call is shown as ―s‖
REM An unsuccessful call is shown as ―f‖ Rem n is used to count the days
FOR N% = 1 TO N
X=RND
IF X<0.5 THEN A$=‖S+ IF X>0.5 THEN A$=‖F‖ PRINT A$
NEXT N% END
A typical outcome of running this program is: sssfffffsss

The program may be run repeatedly, sometimes over large number of N. in this case, the total number of S‘s and F‘s should be approximately equal.

Salesman Example 4

A salesman arranged to make a call each day for the next 10 working days. Previous experience showed that each arranged call had a 10% chance of cancellation. When a call was made the expected chances of success in making sales are as shown below.

Result	**%**
No sale	50
1 unit sold	10
2 unit sold	30
3 units sold	10

At the start of the 10-day period he assumed that 5 units were in stock and that a further 5 would be available for dispatch from day 6. it was the policy of the firm to dispatch orders on the same day they were placed. However, if no stock were available, orders would be held until the next delivery of stock.

Use a tabular simulation to cover the 10-days. Show whether each call was made and its result. Show also the level of stock held at the end of each day. Use the following random numbers: 5,4,5,6,2,9,3,0,3,9,3, 9, 4, 8, 4, 9, 8, 4.

Solution

Note that tabular simulation is a table constructed to produce a record of what has been

simulated to occur and to facilitate any analysis or monitoring of processes required. The first column would record an incremental number of events, other columns would record random numbers and the corresponding results, while others would monitor the implications of earlier columns. There is no fixed rule for doing this. Flexibility is therefore very necessary.

Two look-up tables are needed to answer this question.
First, we need to determine whether the call took place or cancelled.

State of call	**Chance**	**Random no.**
Took place	90%	0,1,2,3,4,5,6,7,8
Cancelled	10%	9

Second, if a call took place we need to know the result

No of units sold	**Chance**	**Random no.**
0	50%	0,1,2,3,4
1	10%	5
2	30%	6,7,8
3	10%	9

The allocation of random numbers in these two tables is based on the fact that each random number digit is assumed to have a 10% chance of occurring. Therefore chances of 10%, 30%, 50% and 90% require one, three, five and one digits respectively.

The following table gives the result of the simulation.

STATE OF CALL					RESULT OF CALL		
Day	Random no	Status	Random no	Stock level	Units sold	Opening stock	ding stock

0	- 5	-	- 9	5	- 3S		
1	4	*	3	2	0S	5	2
2	5	*	9	2	0S,2H,1W	2	2
3	6	*	4	-1	0S,2H,1W	2	-1
4	2	*	8	-1	0S,2H,3H	-1	-1
5	6	*	4	-3	5S	-1	-3
6	9	* x	- 9	5-	-	-3+5=2	2
7	3	*	8	3=2	-1	- 2	-
8	0	*	4	-	-3	-1	-1
9	3	*		-1	-3	-3	-3
10				-3			-3
				-3			

Note * means call took place and x means call cancelled.

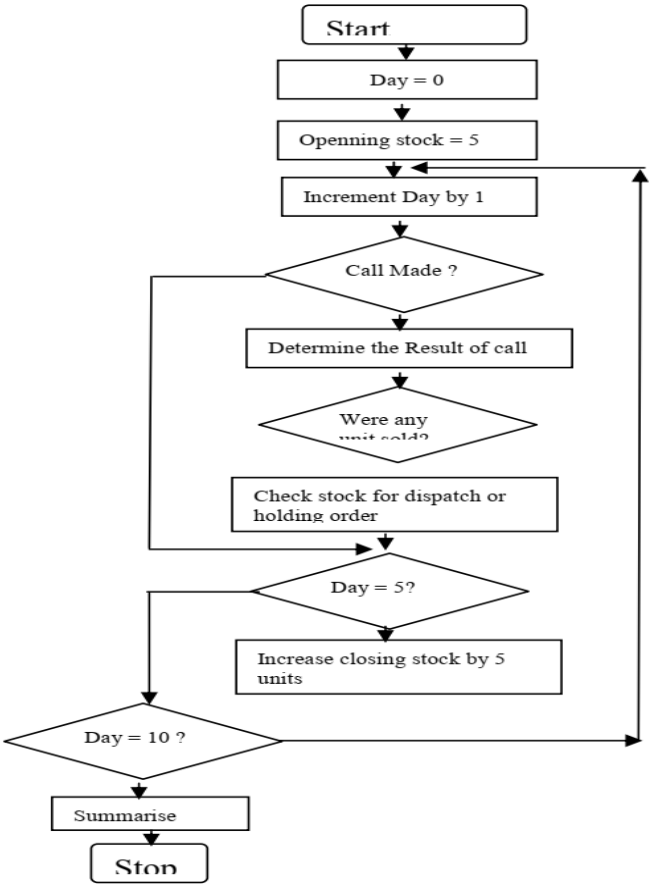


Fig. 2: The flow-chart of the problem solution

Note that the first random number in the table contains a 9 for day 7, so no call was made that day. Hence only 9 random numbers are needed to determine the result of call.

It will be noted that each delivery was followed 3 days later by stock-out. Thus there would have been no stock out if the initial stock level had been set to 8 units.

Notice that only two delivery periods have been simulated. Useful conclusions can be drawn after repeated simulations many more times.

3.4 Goods Production

Example 5 - An engineering company has two machines A and B to produce a product Z. the daily output of each has been specified in the table below.

MACHINE A		MACHINE B	
Daily output	Chance	Daily output	Chance
0	20	0	10
7	20	6	30
8	30	7	40
9	30	8	20

In addition, quality control tests give each day's output a probability 0.95 chance of being accepted and a probability 0.5 of rejection. Complete a tabular simulation to cover a period of 10 days, monitoring daily and cumulative output.

Solution

The three look-up tables needed are: one for each of the two machines and one for quality check.

The Quality control check

Decision	Chance %	Random No.
Accept	95	00, 01, …, 94
Reject	5	95,96,97,98,99

MACHINE A			**MACHINE B**		
Daily Output	**Chance%**	**Random No**	**Daily output**	**Chance%**	**Random No**
0	20	0,1	0	10	0
7	20	2.,3	6	30	1,2,3
8	30	4,5,6	7	40	4,5,6,7
9	30	7,8,9	8	20	8,9

As the chances in the first two tables are all multiples of 10%, only one digit from the set {0,1,2,3,4,5,6,7,8,9} is needed for each 10% chance.

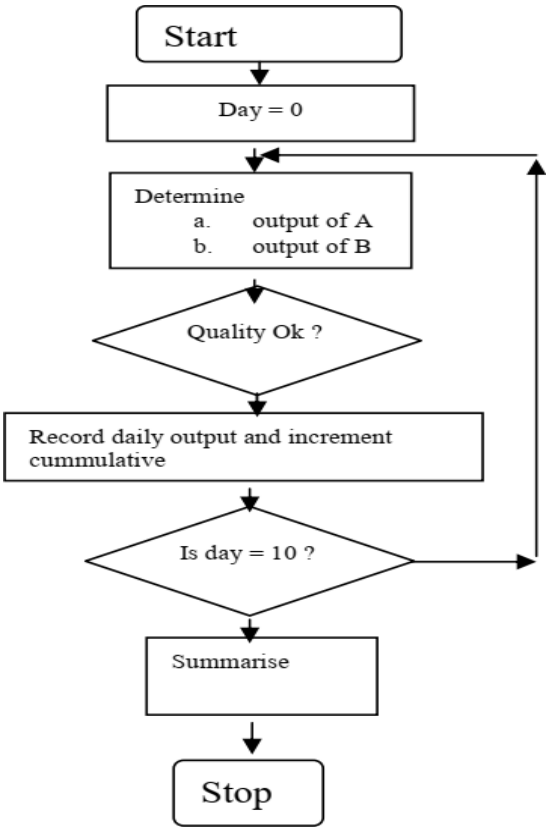
But in the quality table, we need probabilities of 0.95 and 0.05. In this case we must use the random digits in pairs and select from the set [00,01,02,…,99]. The set contains 100 pairs and we can assume that each pair has 1% percent chance of being selected. To simulate a probability of 0.95 we use 95 of the pairs and for 0.05 we use 5.

To carry out the simulation, random number blocks are used. A block is selected randomly and then the digits are used sequentially. No cheating is allowed. Random numbers should be used systematically. In the order in they appear.

The random number blocks used are shown below:

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

The flowchart of the simulation is:



The table is now produced as this:

Machine A **Machine B** **Quality Test**

Day	**Ran No**	**Output**	**Ran No**	**Output**	**Ran No**	**Result**	**Daily output**	**Cumm. output**
1	4	8	1	6	79	Accept	14	14
2	0	0	4	7	85	Accept	7	21
3	8	9	0	0	57	Accept	9	30
4	5	8	2	6	68	Accept	14	44
5	5	8	1	6	18	Accept	14	58
6	6	8	9	8	08	Accept	16	74
7	0	0	2	6	27	Accept	6	80
8	9	9	2	6	05	Accept	15	95

9	1	0	9	8	66	Accept	8	103
10	9	9	3	6	39	Accept	15	118

Considering the first block of random numbers, the first random number is 4, this is looked up in the table for machine A, the second in table 1 is for machine B, and then the next two random numbers 7,9 are looked up in table for quality check. For each subsequent day, four random digits are taken from the block of random numbers in that order.
We conclude that the cumulative output for 10 days was 118 units and that no output was rejected. Further simulation can be carried out and the result compared with the above to give clearer indication of reliability of the result.

4.0 Self-Assessment Exercise(s)

Answer the following questions:

1. Carry out the tabular simulation in example 5 using new set of random numbers. Compare your answer with that obtained in this example.
2. A company trading in motor vehicle parts wishes to determine the level of stock it should carry for the items in its range. Demand is not certain and there is a lead- time for stock replenishment. For item X, the following information is obtained:

Demand (unit per day)	Probability
3	0.1
4	0.2
5	0.3
6	0.3
7	0.1

Carrying cost per unit day	N2.00
Ordering cost per order	N50.00
Lead time for replenishment in days	3
Stock on hand at the beginning of simulation	20units

You are required to:
Carry out a simulation run over a 10-days period with the objective of evaluating the following inventory rule:
Order 15 units when present inventory plus any outstanding order falls below 15 units. The sequence of random numbers to be used is 0,9,1,1,5,1,8,6,3,5,7,1,2,9 using the first number for day 1. Your calculation should include the total cost of operating this inventory rule for 10days.

3. Carry out the simulation requested in example 3 using a six sided die to generate the sets of random numbers required. Use the subset {1, 2, 3) to mean success and the

subset {4,5,6} to mean failure for the 10-day period. Use the subsets [1,3,5] for success and [2,4,6] for failure for the next 10-day period.

5.0 Conclusion

In this unit we have manually simulated the application of queuing systems in different areas of daily activities. This reader is expected to extend his/her knowledge by trying some other areas of daily endeavours, and completing the exercises.

6.0 Summary

In this unit we have designed queuing systems in:

- Car-wash
- Sales calls
- Goods production, and
- Written computer programs in Basic languages to simulated those experiments

7.0 Further Readings

- Bhat, U. N. (2015). *An introduction to queueing theory: Modeling and analysis in applications*. Boston, MA: Birkhäuser.
- Medhi, J. (2019). *Introduction to queueing systems and applications*. London, UK: New Academic Science.
- Trivedi, K. S. (2016). *Probability and statistics with reliability, queuing, and computer science applications*. Hoboken, NJ: Wiley.
- Khinchin, A. I. (2013). *Mathematical methods in the theory of queuing*. Mineola, NY: Dover.
- Thomopoulos, N. T. (2012). *Fundamentals of queuing systems: Statistical methods for analyzing queuing models*. New York: Springer.
- Dudin, A., Klimenok, V. I., & Višnevskij, V. (2020). *The theory of queuing systems with correlated flows*. Cham, Switzerland: Springer.
- Dudin, A., Klimenok, V. I., & Višnevskij, V. (2020). *The theory of queuing systems with correlated flows*. Cham, Switzerland: Springer.

Module 4: SIMULATION LANGUAGES

Module Introduction

This module is divided into two (2) units

Unit 1: Examples of Simulation Languages

Unit 2: The SIMNET II Language

Unit 1: Examples of Simulation Languages

Contents

1.0 Introduction

Most conventional programming languages are not suitable for writing simulation programs. The programmer is usually confronted with a number of detailed decisions. He needs flexible tools for generating dynamic models, model formulation, programming verification, validation and experimental design and analysis. The basic purpose of most simulation studies is to compare alternatives. Therefore, the simulation program must be flexible enough to readily accommodate the alternatives that will be considered. Most of the instructions in a simulation program are logical operations whereas the relatively little actual arithmetic work required is usually of a very simple type. This should be reflected in the choice of computer programming language to be used.

2.0 Intended Learning Outcomes (ILOs)

By the end of this unit the reader should be able to:

- State the Purpose of simulation languages
- List Types and Examples of Simulation Languages
- State the approaches to model development

3.0 Main Content

3.1 The Purpose of Simulation Language

A computer **simulation language** describes the operation of a simulation on a computer. We have stated the two major types of simulation: continuous and discrete-event though more modern languages can handle combinations. Most languages also have a graphical interface and at least simple statistical gathering capability for the analysis of the results.
An important part of discrete-event languages is the ability to generate pseudo-random numbers and variates from different probability distributions.

The above considerations partly motivated the development of simulation languages in the early 1960's. These languages were designed especially to expedite the type of programming unique to simulation. Their specific purposes include the following:

1. To provide a convenient means of describing the elements that commonly appear in simulation models.
2. To expedite changing the design configuration of the system being simulated so that a large number of configurations can be considered easily.
3. To provide some form of internal timing and control mechanism with related commands to assist in the kind of book-keeping that is required when executing a simulation run.
4. To provide simple operational procedures, such as introducing changes into simulation models, initializing the state of the model, altering the kind of output data to be generated and stacking a series of simulation runs.

In simulation, computer models (literally) imitate the behaviour of the real situation as a function of time. As the simulation advances with time, pertinent statistics are gathered about the simulated system, in very much the same way it is carried out in real life.
But we pay attention to the system especially when changes in statistics take place. Such changes are associated with the occurrence of events, for example in a bank operation, arrival and departures from a facility, points in time at which the length of the queue and/or the idle/busy status may change.

3.2 Types and Examples of Simulation Languages

Discrete-event simulation languages, view the model as a sequence of random events each causing a change in state.

- GPSS - **General Purpose Simulation System** (originally **Gordon's Programmable Simulation System** after creator Geoffrey Gordon. The name was changed when it was decided to release it as a product) is a discrete time simulation language, where a simulation clock advances in discrete steps. A system is modelled as transactions enter the system and are passed from one service (represented by blocs) to another. This is particularly well suited for problems such as a production factory. It was popular in the late 1960s and early 1970s but is little used today.

- Siman, a language with a very good GUI (ARENA) of Rockwell company

- SimPy - is a process-based, object-oriented discrete-event simulation language. It is implemented in standard Python and released as open source software under the GNU Lesser General Public License (LGPL). It provides the modeller with components for building a simulation model including *Processes*, for active entities like customers, messages, and vehicles, and *Resources*, for passive components that form limited capacity congestion points like servers, checkout counters, and tunnels. There are two varieties of *Buffer* classes, *Level*s to hold stored quantities and *Store*s to hold sets of objects. It has commands to aid interaction between entities. It provides *Monitor* and *Tally* objects to aid in gathering statistics but the generation of random variates depends on the standard Python random module. Because it is implemented in Python, SimPy is platform- independent. SimPy simulates parallel processes by an efficient implementation of coroutines using Python's generators capability. It is based on ideas from Simula and SIMSCRIPT II.5. The first version was released in December 2002. Version 2.0, including an object-oriented but compatible interface and new documentation, was released in January 2009. Version 2.0.1, was released in April 2009.
- SIMSCRIPT II.5 - is the latest incarnation of SIMSCRIPT, one of the oldest computer simulation languages. Although military contractor CACI released it in 1971, it still enjoys wide use in large-scale military and air-traffic control simulations.
  SIMSCRIPT II.5 is a powerful, free-form, English-like, general-purpose simulation programming language. It supports the application of software engineering principles, such as structured programming and modularity, which impart orderliness and manageability to simulation models.

- Simula - **Simula** is a name for two programming languages, Simula I and Simula 67, developed in the 1960s at the Norwegian Computing Center in Oslo, by Ole- Johan Dahl and Kristen Nygaard. Syntactically, it is a fairly faithful superset of ALGOL 60. Simula 67 introduced objects, classes, subclasses,[1]:2.2.1 virtual methods, coroutines, discrete event simulation, and features garbage collection. Simula is considered the first object-oriented programming language. As its name implies, Simula was designed for doing simulations, and the needs of that domain provided the framework for many of the features of object-oriented languages today. Simula has been used in a wide range of applications such as simulating very large scale integration (VLSI) designs, process modelling, protocols, algorithms, and other applications such as typesetting, computer graphics, and education. Since Simula-type objects are reimplemented in C++, Java and C# the influence of Simula is often understated. The creator of C++, Bjarne Stroustrup, has acknowledged that Simula 67 was the greatest influence on him to develop C++, to bring the kind of productivity enhancements offered by Simula to the raw computational speed offered by lower level languages like BCPL. Simula is still used for various types of university courses, for instance, Jarek Sklenar teaches Simula to students at University of Malta.

Continuous simulation languages, the model essentially as a set of differential equations.

- Advanced Continuous Simulation Language (ACSL), (pronounced "axle"), is a computer language designed for modelling and evaluating the performance of continuous systems described by time-dependent, nonlinear differential equations. It is a dialect of the Continuous System Simulation Language (CSSL), originally designed by the Simulations Council Inc (SCI) in 1967 in an attempt to unify the continuous simulations field. ACSL is an equation-oriented language consisting of a set of arithmetic operators, standard functions, a set of special ACSL statements, and a MACRO capability which allows extension of the special ACSL statements.

ACSL is intended to provide a simple method of representing mathematical models on a digital computer. Working from an equation description of the problem or a block diagram, the user writes ACSL statements to describe the system under investigation.

An important feature of ACSL is its sorting of the continuous model equations, in contrast to general purpose programming languages such as Fortran where program execution depends critically on statement order Applications of ACSL in new areas are being developed constantly.

Typical areas in which ACSL is currently applied include control system design, aerospace simulation, chemical process dynamics, power plant dynamics, plant and animal growth, toxicology models, vehicle handling, microprocessor controllers, and robotics.

- Dynamo - DYNAMO was used for the system dynamics simulations of global resource-depletion reported in the Club of Rome's Limits to Growth. Originally designed for batch processing on mainframe computers, it was made available on minicomputers in the late 1970s, and became available as "micro-Dynamo" on personal computers in the early 1980s. The language went through several revisions from DYNAMO II up to DYNAMO IV in 1983, but has since fallen into disuse.

- **SLAM** - Simulation Language for Alternative Modelling

- **VisSim**, is a visual block diagram language for simulation of dynamical systems and model based design of embedded systems. It is developed by Visual Solutions of Westford, Massachusetts is widely used in control system design and digital signal processing for multidomain simulation and design. It includes blocks for arithmetic, Boolean, and transcendental functions, as well as digital filters, transfer functions, numerical integration and interactive plotting. The most commonly modeled systems are aeronautical, biological/medical, digital power, electric motor, electrical, hydraulic, mechanical, process, thermal/HVAC and econometric.

Hybrid, and other.

- Simulink - developed by MathWorks, is a commercial tool for modelling, simulating and analyzing multidomain dynamic systems. Its primary interface is a graphical

block diagramming tool and a customizable set of block libraries. It offers tight integration with the rest of the MATLAB environment and can either drive MATLAB or be scripted from it. Simulink is widely used in control theory and digital signal processing for multidomain simulation and Model-Based Design.

- SPICE - Analog circuit simulation

- **EICASLAB** is a software suite providing a laboratory for automatic control design and time-series forecasting developed as final output of the European ACODUASIS Project IPS-2001-42068 funded by the European Community within the Innovation Programme. The Project - during its lifetime - aimed at delivering in the robotic field the scientific breakthrough of a new methodology for the automatic control design.

  To facilitate such a knowledge transfer, EICASLAB was equipped with an ―automated algorithm and code generation‖ software engine, that allows to obtain a control algorithm algorithm even without a deep knowledge of the theory and the methodology that are otherwise normally required with traditional control design methodologies.

- Z simulation language - **Z** is a stack-based, complex arithmetic simulation language by ZOLA Technologies

3.3 Approaches to model development

There are two approaches in simulation model development – next-event scheduling and process operation. Both approaches are based on the concept of collecting statistics when an event occurs. Both differ in the amount of details that the user must provide.

We compare the two methods.

	NEXT-EVENT SCHEDULING	**PROCESS OPERATION**
1.	Requires extensive modelling effort	Most of modelling efforts are automated
2.	Very flexible	Not very flexible, restricted in scope
3.	Extensive coding is required	More compact and easier to implement

We can also on this basis classify simulation languages according to the type of simulation:

Next-event scheduling:	SIMSCRIPT, GASP,
Process Operation:	SIMULA, GPSS, SIMNET II, SIMAM, SLAM

4.0 Self-Assessment Exercise(s)

Answer the following questions:

- Find more information on the simulation languages list above or those you may find

on the net to enrich your understanding.

- Briefly discuss the important features of two simulation language; one from each category.
- Differentiate between the next event scheduling and process operation based simulations
- What are the Categorisations of simulation languages? Given two examples for each category.

5.0 Conclusion

For very serious work requiring simulation, you need a simulation software package. Most of them are not open source, this implies that you be ready to part with some money.

6.0 Summary

In this unit we looked at:

- The purpose of simulation languages
- Some types and examples of simulation languages, and categorised them according to: discrete, continuous, and hybrid
- The development strategies; the next event scheduling and process operation based simulations

7.0 Further Readings

- Gordon, S. I., & Guilfoos, B. (2017). *Introduction to Modeling and Simulation with MATLAB® and Python*. Milton: CRC Press.
- Zeigler, B. P., Muzy, A., & Kofman, E. (2019). *Theory of modeling and simulation: Discrete event and iterative system computational foundations*. San Diego (Calif.): Academic Press.
- Kluever, C. A. (2020). *Dynamic systems modeling, simulation, and control*. Hoboken, N.J: John Wiley & Sons.
- Law, A. M. (2015). *Simulation modeling and analysis*. New York: McGraw-Hill.
- Verschuuren, G. M., & Travise, S. (2016). *100 Excel Simulations: Using Excel to Model Risk, Investments, Genetics, Growth, Gambling and Monte Carlo Analysis*. Holy Macro! Books.
- Grigoryev, I. (2015). *AnyLogic 6 in three days: A quick course in simulation modeling*. Hampton, NJ: AnyLogic North America.
- Dimotikalis, I., Skiadas, C., & Skiadas, C. H. (2011). *Chaos theory: Modeling, simulation and applications: Selected papers from the 3rd Cghaotic Modeling and Simulation International Conference (CHAOS2010), Chania, Crete, Greece, 1-4 June, 2010*. Singapore: World Scientific.

- Velten, K. (2010). *Mathematical modeling and simulation: Introduction for scientists and engineers*. Weinheim: Wiley-VCH.

Unit 2: The SIMNET II Language

Contents

1.0 Introduction

SIMNET is a network-based discrete simulation language that differs from available process languages in that it utilizes exactly four nodes: a source for creating transactions, a queue where waiting may take place, a facility where service is performed, and an auxiliary that is introduced to enhance the modelling flexibility of the language. Each node is provided with sufficient information that defines the exact manner in which a transaction enters, resides in, and leaves the node.

2.0 Intended Learning Outcomes (ILOs)

After reading this unit, you should be able to:

- Design and develop simulation programs using SIMNET II language; including description of ;
  - The Nodes Statements
  - Rules for Operation of Nodes
  - The definition of Nodes
  - The Mathematical Expressions
  - Layout of SIMNET II model
  - The Output Report

3.0 Main Content

3.1 Design of SIMNET II Language

The design of the language provides automatic collection of global statistical summary based on either the *subinterval* or the *replication* method. It is also possible to execute independent runs with different initial data in a single simulation session. Execution in SIMNET can be carried out interactively or in batch mode. Although SIMNET does not make use of external (FORTRAN) subroutines, the language is capable of modelling complex situations rather conveniently.

Special routing of transactions among the four nodes is effected using seven types of branches and the so-called special assignments. SIMNET II offers flexible computational capabilities at a level equal to FORTRAN with access to all internal simulation data and files. The computational and modelling power of the language eliminates the need for the use of external FORTRAN or C inserts. The companion system ISES combines input, no-programming animation, debugging, and execution in a user-friendly interactive environment. An important feature of the system is that ISES generates the animation model without any special programming effort on the part of the user

The language is based on network approach that utilizes three main nodes and one auxiliary node which include:

1. **Source node -** from which transactions (customers) arrive;
2. **Queue node -** where waiting takes place if necessary;
3. **Facility node -** where service is performed;
4. **Auxiliary node -** which is added to enhance the modelling capabilities of SIMNET II.

Nodes in SIMNET II are connected by **branches**. As the transactions traverse the branches, they (nodes) perform important functions such as:

1. Controlling transaction flow anywhere in the network;
2. Collecting pertinent statistics
3. Performing arithmetic calculations

During the simulation execution, SIMNET II keeps track of the transactions by placing them in **files**. A file can be thought of as a two dimensional array with each row being used to store information about a unique transaction. The columns of the array represent the attributes that allow the modeler to keep track of the characteristics of each transaction. This means that attributes are local variables that move with their respective transactions wherever they go in the model network.

SIMNET II uses three types if files – Event Calendar, Queue and Facility.

The **event calendar** (or E. FILE) is the principal file that drives the simulation. It keeps track of an updated list of the model‘s events in their proper chronological order. The functions of

the queue and facility files are different from that of E. FILE. These files once defined by the model are automatically maintained by SIMNET II.

3.2 SIMNET II Nodes Statements

Statements in SIMNET II are specified node by node. The general format is:
Node identifier; field 1; field 2,….; field m:

The **node identifier** consists of user-defined names (12 characters maximum) followed by one of the codes *S, *Q, *F, or *A, which identify the name as either *a source, a queue, a facility or an auxiliary*. The node identifier is then followed by a number of fields separated by semicolons with the last terminating with a colon. Each field carries information that is needed for the operation of the node. The order of the fields must be followed for the processor to recognize their information content. If a field is not used, or defaulted, the position is indicated with a semicolon.
For example, the statement ARRIVE:

ARRIVE *S;10;;;;LIM = 500:

Identifies a source node named ARRIVE. The value 10 for the first field is the time between successive arrivals. Fields 2, 3 and 4 assume default values (see later) while field
5 indicates that the maximum number of creations from ARRIVE is limited to 500 transactions.

The above statement could be written also as:
ARRIVE *S; 10; /5/LIM =500:
Notice that the semicolon indicating fields 2, 3 and 4 are omitted. Instead, the next field number is indicated as shown.
Also, /5/ or more generally /n/ can be replaces by descriptive reserved words or single letters. E.g. /multiple/ or /m/, /limit/ or /1/, /source/ or /s/ and /resource/ or /l/:

SIMNET II programming is in free format and is upper/lower case insensitive. A statement can spill over one line provided each line is terminated by & (ampersand).
Example:

```
ARRIVE *S; 10; /5/LI &                  !Line 1 of ARRIVE
                        M = 500:        !Line 2 of ARRIVE
```

! Line 1 of ARRIVE is a comment and is ignored by SIMNET II processor. We now define the various fields of SIMNET II‘s four nodes.

3.2 The Definition the four Nodes

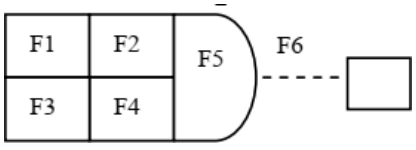
3.3.1 Source Node

The source node is used to create the arrival of transactions into the network. The definitions of its fields and its graphic symbol are given in table below.

The general format is SNAME *S; F1; F2; F3; MULT=F4; LIM=F5; F6; F7; *T:

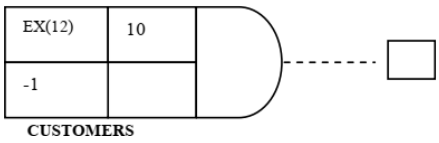
Field	Field	Field Identifier	Description	Type of Data	Default Values
F1	Inter-arrival time			Expression	0
F2	Occurrence time of first creation			Expression	0
F3	Mark attribute number with the attribute automatically carrying creation time if F3>0 or serial number if F3 < 0			Constant	none
F4	Simultaneous transaction per single creation			Constant or Variable	1
F5	Limit on number of creations if F5>0 or limit on time of creation If F5<0			Constant or Variable	infinity
F6	/s/	Output select rule (see later)			none
F7	Resources returned by source (see later)				none
*T	List of nodes reached from source by direct transfer (see later)				none

The Graphic Symbol and Representation:



We use examples below to look at the operation of the fields: F1 to F5 and *T. Example 1
Individual customers upon arrival at a car registration facility are assigned serial numbers that identify the order in which they will receive service. The inter-arrival time is exponential with mean 12 minutes. The first customer arrives about 10 minutes after the facility opens.

The graphic symbol is:



The source statement defining the solution is:

CUSTOMERS *S;EX(12);10;-1:

CUSTOMRS is the name of the source node.

- EX(12) in field 1 indicates the time between successive arrivals as a random sample based on exponential distribution with mean 12 (see table of random functions). Symbol EX(meaning exponential distribution) is reserved and is recognised by SIMNET II processor.
- 10 in field2 designates the arrival time of first customer as 10 starting from 0 (zero datum). Note that any mathematical expression may be used on the two field.
- -1 in field3 indicates that attribute 1 will automatically carry the serial numbers 1,2,3,… for the successive arrivals.

Since the third ends with a colon, all the remaining fields are defaulted. Note the following.

(a) Any mathematical expression may be used in the first two fields.
(b) In SIMNET II, attributes are designated by the reserved array A(.), so that – 1 in field3 signifies that A(1) = 1,2,3,… for successive arrivals.

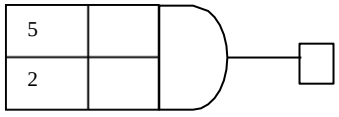
Example 2:

TV units arrive every 5 minutes for packaging. It is desired to keep track of the arrival time of each transaction in attribute 2

The following statements are equivalent.

TVS *S;5;; 2;* PACKAGING:

TVS *S;5;; 2; goto- PACKAGING:



Explanation

The creation time for the first TV unit is 0 because field F2 is defaulted. Because F3 = 2 (2>0), mark attribute A(2) for successive customers will assume the respective values 0,5,10,15,….

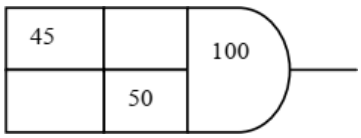
The transactions leaving source TVS will be transferred to a queue (buffer) node named PACKAGING as shown by the *T field.

Note: - *T always occupies the last field of the node, regardless the number of defaults fields that may proceed it.

- goto in the second statement has replaced the * in the first statement.

Example 3:

A mill is contracted to receive 100 truckloads of logs. Each truckload includes 50 logs. The mill processes the logs at a time. The arrival of trucks at the mill are spaced 45 minutes apart.

The required SIMNET II statement is: 

TRUCKS *S; 45; MULT = 50; LIM = 100; *MILL:

The LIM field indicates that the source TRUCKS will generate 100 successive transactions after which it will go dormant. As each transaction leaves TRUCKS, it will be replaced by 50 identical transactions representing the logs as shown in the MULT field (that is F4).

3.3.2 Queue Node

The purpose of a queue (buffer) is to house waiting transactions. An arriving transaction that cannot be serviced immediately must wait in a queue until the facility becomes available. In SIMNET II:

- A queue size may be finite or infinite.
- the initial number waiting at start of the simulation may be zero or greaterthan zero.
- transactions waiting in a queue may leave according to the selected queue discipline.
- the queue itself may act as an **accumulator** whereby a specified number of waiting transactions are replaced by one leaving transaction.

The descriptions of the various fields of a queue node are summarized in the tables below and tables 1 and 2.

The general format is QNAME *Q; F1(SUBF1), F2(SUBF2); F3; F4;F5; *T:

Field	Field Identifier Description	Type of Values	Default
F1	Maximum queue size	Constant or	Infinity
SUBF1	Initial number in queue	Variable	0
F2	Number of waiting transactions	Constant or Variable	
	required to create one leaving transaction	Constant or variable	1
SUBF2	Rule for computing the attributes of the leaving transactions when F2 >1: SUM, PROD, FIRST, LAST, H1(#),LO(#), where # is an attribute number (see table 1)		LAST

F3	/d/	Queue discipline: FIFO,LIFO, RAN, HI(#), LO(#), where # is an attribute number (see table 2)	FIFO
F4	/s/	Output select rule (see later) single creation	none
F5	/r/	Resources returned by queue (see later)	none
*T		List of nodes reached from queue by direct transfer (see later)	none

Graphic Symbol is:

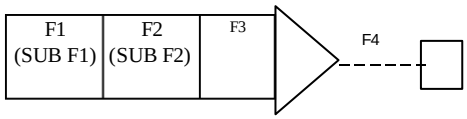


Table 1: Rules for Computing Exiting Transaction Attributes In an Accumulation Queue.

Rule	Description
SUM	Sum of the attributes of the accumulated transaction
PROD	Product of the attributes of the accumulated transaction
FIRST	Attributes of the first accumulated transactions
LAST	Attributes of the last of the accumulated transactions
HI(#)	Attributes of the transactions having the highest A(#) among all accumulated transactions
LO(#)	Attributes of the transactions having the lowest A(#) among all accumulated transactions.

Table 2 : Queue Discipline Codes

Discipline	Leaving Transaction
FIFO	First in, first out
LIFO	Last in, first out
RAN	RANdom selection
HI(#)	Transaction having the Highest attribute A(#), where # is an integer constant
LO(#)	Transaction having the Lowest Attribute (#).

3.4 Queue Node Examples

Example 1:

Rush and regular jobs arrive at a shop randomly with rush jobs taking priority for processing.

Solution

A direct way to represent this situation is to associate the job type with an attribute. Let A(1) = 0 and A(1) = 1 identify the regular and rush jobs, respectively. These jobs are then ordered in a queue named JOBQ as follows:

JOBQ *Q;;;HI(1):

The queue discipline HI(1) requires that all transactions be ordered in descending order of the value of A(1). This means that rush jobs with A(1) = 1 will be placed at the head of the queue. Notice that field 1 is defaulted, signifying that queue JOBQ has an infinite capacity.

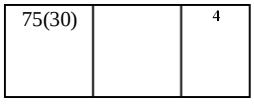
If the values assigned to A(1) are interchanged so that A(1)=0 represents the rush job, the queue discipline must be changed to LO(1) as follows:

JOBQ *Q;;;LO(1):

Example 2.

Units of a product are packaged four to a carton. The buffer area can hold a maximum of 75 units. Initially, the buffer is holding 30 units.

Solution



Let QUNIT represent the buffer, the associated statement is given as:

QUNIT *Q; 75(30); 4:

The first field sets the maximum queue capacity (=75) and the initial number in the system (=30). The queue discipline is FIFO because field 3 is defaulted. Field 2 indicates that four product units will be converted to a single carton. By default, the attributes of the carton transaction will equal those of the LAST of the four unit transactions forming the carton.

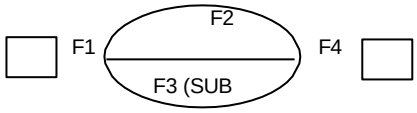
Facility Node

A *facility node* is where service is performed. In SIMNET II, a facility has a finite capacity representing the number of parallel servers. During the simulation, each server may be busy or idle. The description of its fields is given below.

The general format is: **FNAME** ***F;F1;F2;F3(SUBF3); F4; F5; *T:**

Field	Field Identifier	Field Description	Type of Values	Default
F1		Rule for selecting an input Queue (see later)		none
F2		Service Time	Expression	0
F3		Number of parallel servers	Constant or Variable	1
SUBF3		Initial number busy servers	Constant or Variable	0
F4	/s/	Output select rule (see later)		none
F5	/r/	Resource(s) acquired/released by facility (see later)		none
*T		List of nodes reached from facility by direct transfer (see later)		none

The Graphic Representation:



3.5 Facility Node Examples

Example 1.

A facility has one server who happens to be busy at the start of the simulation. The service time is 15 minutes. Units completing the service are removed (TERminated) from the system.

Using the name SRVR, the SIMNET II statement is given as:

SRVR *F;;15;1(1); *TERM:

Explanation

Field 1 is not needed in this situation because it normally deals with multiple queue input to the facility, which will be discussed later. In field 2 the value 15 provides the service time in the facility. Field 3 shows that the facility has one server that is initially busy. The *T field shows that the completed transaction will be Terminated by using *TERM (or goto-TERM). The symbol TERM is a reserved word of SIMNET II. It is not a node but simply a code that will cause the transaction to vanish from the system.

Example 2.
A facility has three parallel servers, two of which are initially busy. The service is exponential with mean 3 [EX(3)].

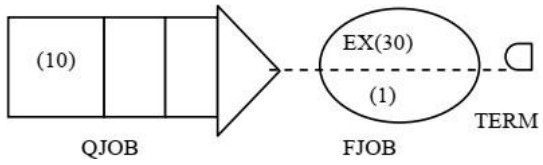
The associated statement is : **SRVR *F;;EX(3); 3(2):**

Example 3.
A small shop has one machine and 10 waiting jobs, in addition to the job that is currently being processed. The processing time is exponential with mean 30 minutes.

The network representing this situation is shown below, where the symbol following the facility represents TERM.

The associated SIMNET II statements are: QJOB *Q; (10):
FJOB *F;;EX(30); (1); *TERM:

Graphic Representation



Explanation

In the network above, since FJOB is initially busy, as indicated by the entry (1) in field 3, the facility will automatically process its resident job using a sample from EX(30) as its processing time. After the job leaves FJOB to be TERMinated, the facility will automatically look back and draw a new job from QJOB. This process is repeated until all 10 jobs are processed.

Auxiliary Node
An auxiliary is an infinite capacity node that will always accept all incoming transactions. The node is designed to enhance the modelling capability of the language. It is mostly suited for representing delays. Also auxiliary is the only node that can enter itself, a characteristic that is particularly useful in simulating repetitive actions (or loops). The table below gives a description of the fields of auxiliary node.

The general format is: **ANAME *A;F1;F2;F3; *T:**

Field	Field Identifier	Field Description	Type of Values	Default
F1	Delay Time		Expression	0

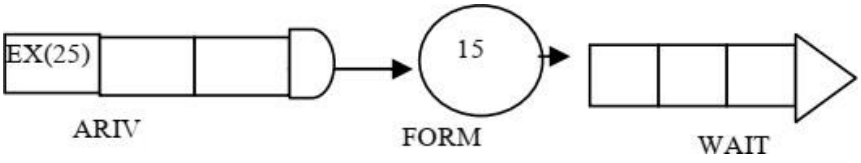
F2	/s/	Rule for selecting an output mode	none
F3	Resource(s) released by auxiliary (see later)		none
F4	List of nodes reached from auxiliary by direct transfer (see later)		

Its graphic representation is:

3.6 Example of Auxiliary Node

Job applicants arrive at an employment office every EX(25) minutes. Each applicant must fill out a form and then wait to be interviewed. It takes approximately 15 minutes to complete the form.

The network segment and statements describing the arrival, completing the form and waiting are given below.




```
ARIV *S;EX(25):
FORM *A;15;
WAIT *Q:
```

Explanation

The model assumes that the forms are immediately accessible to the arriving applicants. This is the reason for representing the process of filling out the form by the infinite capacity auxiliary FORM. If the forms were to be completed with the assistance of a clerk, the auxiliary FORM would have to be replaced with a single-server facility preceded by a queue.

3.7 Rules For The Operation of Nodes

The following are the rules for the operation of source, queue, facility and auxiliary nodes. Violations of these rules will be detected by the SIMNET II processor and an appropriate error message given.

1. A source may not be entered by any other node, including another source.
2. A queue may not feed **directly** into another queue, nor can it feed back into itself.
3. Facilities may follow one another without intervening queues. However, a facility may not feed directly into itself.
4. An auxiliary is the only node that can feed directly into itself, thus allowing the simulation of loops.

5. A transaction will skip a queue if the queue is not full and its successor node accepts the transaction, even if the queue happens to have waiting transactions when the skipping transaction arrives.
6. If a facility is preceded by a queue, the facility will automatically attempt to draw from the waiting transactions immediately upon the completion of service. If the queue happens to be empty, the facility will go dormant until it is **revived** by newly arriving transaction.
7. Movement of transactions in and out of the queue can only be caused by **other** nodes. The queue itself is not capable of initiating this movement.
8. When facilities follow one another in tandem or when the intervening buffers (queue) have limited capacities, a transaction completing service in one of the facilities will be **blocked** if its successor node is full a (finite) queue or a busy facility. The **unblocking** will take place automatically in a chain effect when the cause of blocking subsides.

3.8 SIMNET II Mathematical Expressions

Mathematical expressions are used in certain fields of nodes, such as the inter arrival time in a source. They may also be used with arithmetic assignments and conditions.

The rules for constructing and evaluating mathematical expressions in SIMNET II are the same as in FORTRAN. An expression may include any legitimate combination of the following elements:

1. User-defined non-subscripted or subscripted (array) variables.
2. All familiar algebraic and trigonometric functions (see table 3).
3. SIMNET II simulation variables that define the status of the simulation during execution (see table 4).
4. SIMNET II random samples from probabilistic distributions (see table 5).
5. SIMNET II special functions (see table 6).

Names of user-defined variables may be of any length, although only the first 12 characters are recognizable by the SIMNET II processor. The name may include intervening blanks but must exclude the following special symbols:

:,;( ) { } + - * / = < > $ & % ? !

These symbols are used to represent specific operations in SIMNET II. The following are typical example of SIMNET II‘s non-subscripted and array variables:

nbr_ of_ machines TIME BET ARVL

Sample(1*(J+K)**2)

SCORE (Sample(I+J), MAX(K, nbr_of_machines)

The algebraic and trigonometric functions accepted by SIMNET II are listed in table 3. The arguments of these functions may be any legitimate mathematical expressions. All given functions have the same properties as in FORTRAN.

SIMNET II simulation variables provide access to all simulation parameters and statistics

during execution. Simulations statistics are provided in the form of current, highest, lowest and average values. For example, LEN(QQ), HLEN(QQ), LLEN(QQ) and ALEN(QQ) define the current, highest, lowest and average LENgth of a queue named QQ. Table 4 describes the SIMNET II simulation variables that can be accessed during execution. These variables may be used directly within any mathematical expression.

Table 5 describes the random functions available in SIMNET II. All arguments can be represented by a SIMNET II mathematical expression. The default value of the random number stream, RS, is 1.

The last element of a mathematical expression includes SIMNET II special variables. These variables includes table look ups TL(argl, arg2) and mathematical functions FUN(arg1) and FFUN(arg1, arg2). See table 6.

Table 3: SIMNET II Intrinsic Functions

Algebraic	
Single argument:	INT,ABS,EXP,SORT,SIGN,LOG,LOG10
Double arguments:	MOD
Multiple arguments:	MAX, MIN
Trigonometric (single argument)	
Regular:	SIN,COS,TAN
Arc:	ASIN,ACOS,ATAN
Hyperbolic:	SINH,COSH,TANH

Table 4: SIMNET II Simulation Variables

Variable	***Definition***
LEN(auxiliary)	Current number of transactions residing In an auxiliary node.
LEN/HLEN/LLEN/AVAL	Current/highest/lowest/average (filename) LENgth of a queue or facility
VAL/HVAL/LVAL/AVAL (variable name)	Current/highest/lowest/average VALue of a Statiscal variable (see later)
LEV/HLEV/LLEV/ALEV	Current/highest/lowest/average (resource name) LEVel of a Resource.
COUNT(node, resource, 0r variable name)	Number of transactions that departed a node since the start of the simulation or the number of updates of a resource or a variable.

RUN.LEN	Length of current run
TR.PRD	Length of transient period
CUR.TIME	Current simulation time
OBS	Current statistic I observation number
NOBS	Total number of observations per run
RUN	Current statistical run number
NRUNS	Total number of runs
AQWA (queue name)	Average wait in queue for all customers including those who do not wait
AQWP (queue name)	Average wait in queue for those who must wait
AFBL (facility name)	Average blockage in a facility
AFTB (Facility name)	Average blockage time in a facility AFIT
(facility name)	Average time facility is idle
AFBT (Facility name)	Average time facility is busy ARTU (resource
name)	Average resource units in transit ARTT
(resource name)	Average time a resource is in transit
ARBT (resource name)	Average time a resource is busy (in use) ARIT
(resource name)	Average time a resource is idle
AFRQ (variable name, cell #)	Absolute histogram frequency of cell # of a
variable RFRQ (variable name, cell #)	Relative histogram frequency of cell # of a variable
NTERM (node name)	Number of transactions terminated from a node
NDEST (node name)	Number of transactions destroyed from a node.

Table: 5 SIMNET II Random Functions

Function	Definition
BE(arg 1, arg2, RS)	[0,1] Beta sample with shape parameters □ = arg1and □ = arg2
B(arg1, arg2, RS)	Binomial sample with parameter n = arg1 and ρ = arg2
DI(arg1,RS)	Discrete Probability sample if arg 1> 0 or linearly-interpolated sample if arg 1< 0
EX(arg1, RS)	Exponential sample with mean 1/□ = arg1
GA(arg1, arg2, RS)	GAmma sample with shape parameters □ = arg1 and 1/□ = arg 2; if □ is a positive integer, the sample is Erlang.
GE(arg1, RS)	GEometric sample with parameter ρ = arg1
LN (arg1, arg2, RS)	LogNormal sample corresponding to a normal distribution with mean = arg1 and standard deviation = arg2
NE(arg1, arg2, RS)	NEgative binomial with parameters c = arg1 and p = arg2
NO(arg1, arg2,RS)	NOrmal sample with mean = argl and standard deviation = arg2
PO(arg1,RS)	Poisson sample with mean = arg1
RND(RS)	[0,1] random sample
TR(arg1,arg2,arg3,RS)	Triangular sample in the interval [arg1,arg3] with mode arg2
UN(arg1,arg2,RS)	Uniform sample in the interval [arg1,arg2]
WE(arg1, arg2,RS)	Weibull sample with shape parameters = arg1 and = arg2

Table 6: SIMNET II special variables

Variable	Definition
TL(arg1,arg2)	Value of a dependent variable obtained from Table Look-up number arg1 given the value of the independent variable is arg2. If arg1 < 0, TL is automatically determined by linear interpolation.
FUN(arg1)	Mathematical expression number arg1 obtained from a predefined list of expressions.
FFUN(arg1,arg2)	Mathematical expression in location (arg1, arg2) obtained from a

predefined two-dimensional array list of expressions.

3.9 Layout of SIMNET II Language

Although SIMNET II statements are free formatted, the segments of the model must follow a specific organization as shown below:

$PROJECT; model name; date; analyst: $DIMENSION; ENTITY(m), array1, array2, ……. Arrayn: $ATTRIBUTES; (descriptive names of attributes):
Definitions Segment: $VARIABLES: (definitions of statistical variables): $SWITCHES: (definitions of logic switches): $RESOURCES: (definitions of scarce resources):
Model Logic Segment: $BEGIN: (model logic statements) $END:
Control Segment: $RUN-LENGTH =(run length): $TRACE=(limits of simulation period to be traced): $TRANSIENT-PERIOD=(transient period length): $RUNS =(number of runs in a single simulation session): $OBS/RUN=(number of statistical observations per run):
Initial Data Segment: $DISCRETE-PDFS: (definitions of discrete probability functions): $INITIAL-ENTRIES: (attributes of initial queue entries): $TABLE-LOOKUPS:(definitions of table look-up functions): $ARRAYS: (initial values of array variables): $CONSTANTS: (initial values of non-subscripted variables): $FUNCTIONS: (definitions of mathematical expressions): $PRE-RUN: (pre-execution arithmetic and READ/WRITE assignments):
$PLOT= (list of model elements to be plotted): $STOP:

$PROJECT and $DIMENSION are mandatory statements that always occupy the first and second statements of the model. The $PROJECT provides general information about the model. The $DIMENSION statement allocates memory dynamically to the model‘s files (queue, facilities and the E.FILE) and user-defined arrays. The dimension m of ENTITY is an estimate of the maximum number of transactions that can be in the system at any time. The only restriction on the use of $DIMENSION is that attributes be defined by the reserved array name A(.). For example, the statement:

$DIMENSION; ENTITY(50), A(5), sample(50,3):

indicates that the maximum number of transactions during execution is estimated not to exceed 50 and each transaction will have five attributes. The double-subscripted array sample (50,3) is defined to have 50 rows and three columns.

The optional $ATTRIBUTES statement is used when it is desired to assign descriptive names to the elements of the A(.) array. For example, suppose that the $DIMENSION statement specifies the attributes array as A(5), meaning that each transaction will have five attributes. The statement

$ATTRIBUTES; Type, ser_nbr(2),, Prod_time: signifies the following equivalences:
A(1)=Type A(2)=ser_nbr(1) A(3)=ser_nbr(2) A(5)=Prod_time.
Notice that the name of A(4) has been defaulted, which means that it has no descriptive name.
The definitions segment of the model defines the model‘s statistical variables, logic switches and resources. All three types of statements are optional.

The logic segment includes the code that describes the simulated system using the nodes and branches.

The control segment provides information related to how output results are gathered during execution. Finally, the initial data segment provides all the data needed to initialise the simulation

Example 1 (Multiserver Queuing Model)
Customers arrive randomly at a three-clerk post office. The interval time is exponential with mean 5 minutes. The service time is also exponential with mean 10 mins. All arriving customers form one waiting line and are served by free clerks on a FIFO basis.
The network representation of the model and the complete SIMNET II statements are given below.

```
$PROJECT; Post Office; 20 January, 2009, M. C. Okoronkwo:
$DIMENSION; ENTITY(30):
$BEGIN:
ARVL *S; EX(5):                    !Arrivals
LINE *Q:                           !Wait in line
$CLKS *F;;EX(10);3;goto-TERM:          !Served by one of 3 clerks
$END:
$RUN-LENGTH = 480                  !Run model for 480 minutes
$TRACE=30-35:                      !trace from 30-35
$RUNS=1:                           !for one run only
$STOP:
```

Explanation
The $DIMENSION statement estimates that at most 30 transactions(customers) will be in the

model at any one time. If during execution this estimate is exceeded, SIMNET II will give an error message. The $DIMENSION of ENTITY must be increased.

The model does not use $ATTRIBUTES, $VARIABLES, $SWITCHES OR
$RESOURCES as shown by the absence of these statements.
The logic of the model is represented by the statements enclosed between $BEGIN and $END. Transactions are automatically created by source ARVL by randomly sampling the inter-arrival time EX(5) with the first arrival taking place at time 0 (default of field 2). Arriving transactions will enter queue LINE if all three clerks are busy otherwise the queue is skipped. When a transaction completes a service, it will be terminated. At this point, facilty CLRKS will look back at QUEUE LINE and bring in the first in line transaction (queue discipline is FIFO by default) the control data of the model shows that it will be executed for one run of length 480 mins. The standard output of the model is shown below:

3.10 Sample Standard Output of the Model SIMNET OUTPUT REPORT

PROJECT: Post office RUN LENGTH = 480.00 NBR RUNS =1
DATE: 20 January, 2009 TRANSIENT PERSON =0.00 OBS/RUN = 1
ANALYST: M.C OKORONKWO TIME BASE/OBS= 480.00

INDEPENDENT RUNS DATA
RUN 1:

QUEUES

CAPACITY	IN:OUT RATIO	AV. LENGTH	MIN/MAX LAST LEN	AV. DELAY (Alt)	AV. DELAY	% zero wait Transaction
****	1:1	1.30	0/12/0	6.51	15.63	58

FACILITITIES

NBR SEVRS	MIN/MAX LAST UTILZ	AV. GROSS UTILZ	AV. BLOCKAGE	AV. BLKGE TIME	AV. IDLE TIME	BUSY TIME
3	3/3/1	1.9931	.0000	.00	8.48	16.78

TRANSACTION COUNT AT T=480.00 OF RUN 1:

NODE	IN	OUT	RESIDING (BLOCKED)	SKIPPING (DESTROYED)	UNLINKED/ LINKED	
*S:ARVL		96			(0)	0
*Q: LINE	40	40	0	56	0/	0
*F:CLRKS	96	95	1	0	0	95

The transaction count given at the end of the report gives a complete history of the flow of transaction during the run. The summary can be helpful on spotting irregularities in the model. In our example, during the 480 minutes run, 96 transactions were created by source ATVL, 40 of which experienced some waiting in queue LINE and the remaining 56 skipped the queue. Facility CLRKs received 96 customers and released 95 with 1 transaction remaining un processed at the end of the run. The remaining column on the count are all zeros. The UNLINKED/LINKED column is used only when the model exposes some file manipulations. The (Blocked) and (DESTROYED) column will show positive values whether a facility is blocked or when transaction are destroyed. Neither case applied in our case.
Queue LINE has an infinite capacity (*** shows), IN:OUT ration of 1:1 shows that each existing transaction corresponds to one waiting transaction. The Average length of 1.30 transactions represents the average run of waiting transactions over the entire length of the run(= 0,12,0) that occurred during the run. Average waiting time of all transactions (including those that do not wait) given by AV. DELAY ALL)=6051minutes. Next column AV. DELAY (+ve WAIT) show the average waiting for those that must wait as
15.63 minutes. Finally, last column indicates that 58% of transactions arriving from source AVRL skip LINE i.e they do not experience any waiting at all.

Facility CLRK has 3 parallel servers. Second column shows that CLRKs
The third column indicates that the average 1.9931 servers (out of 3) were busy throughtout the run, thus reflecting a gross percentage utilization of (1.9931/3)*100 = 66.4% the average BLOCKAGE records the average number of productive occupancy of the facility.

4.0 Self-Assessment Exercise(s)

1. In each of the following cases, what will be the value of attribute A(1) associated with the first three transactions that exited the source node?

a. AVIV *S;5;;1:
b. ARIV *S;5;31:
c. ARIV *S;5;3;-1:

2. How many transactions will be generated by each of the following source statement during the first 20 time units of the simulation?

a. ARIV *S;5;/L/LIM=3:
b. ARIV *S;/m/MULT=2:

3. The first five transactions arriving at queue QQ have the following attributes:

Transaction	A(1)	A(2)
1	4	9
2	7	-3
3	1	10
4	3	14
5	2	6

Show how these transactions will be ordered in QQ in each of the following cases:
a). QQ *Q: **b**) QQ *Q;/d/LIFO: **c**) QQ *Q;/d/HI(1): **d)** QQ *Q;/d/LO(2):

4. Consider the following model segment:
   QQ *Q;(3):
   FF *F;;2;(1);goto-TERM:

a. How many transactions will be processed by facility FF?

b. Determine the simulation time at which each transaction will leave FF.

5.0 Conclusion

The essence of this unit is to look at simulation languages designed to expedite simulation, through:

- Provision convenient means of describing the elements that commonly appear in simulation models.
- Expediting the change in the design configuration of the system being simulated so that a large number of configurations can be considered easily.
- Provision of simple operational procedures, such as introducing changes into simulation models, initializing the state of the model, altering the kind of output data to be generated and stacking a series of simulation runs

6.0 Summary

We looked at the following:

- Purpose of simulation language which is to describes the operation of a simulation on a computer
- Types and Examples of Simulation Languages which is broadly divided into two discrete and continuous events simulation languages.
- The approaches to model development divided into two: next-event scheduling and process operation with there various parameters
- The SIMNET II language and its development including:
  - Its Nodes Statements
  - The rules For The Operation of Nodes
  - Node Definition
  - The Mathematical Expressions
  - Layout of SIMNET II MODEL, and
  - SIMNET Output Report

7.0 Further Readings

- Gordon, S. I., & Guilfoos, B. (2017). *Introduction to Modeling and Simulation with MATLAB® and Python*. Milton: CRC Press.
- Zeigler, B. P., Muzy, A., & Kofman, E. (2019). *Theory of modeling and simulation: Discrete event and iterative system computational foundations*. San Diego (Calif.): Academic Press.
- Kluever, C. A. (2020). *Dynamic systems modeling, simulation, and control*. Hoboken, N.J: John Wiley & Sons.
- Law, A. M. (2015). *Simulation modeling and analysis*. New York: McGraw-Hill.
- Verschuuren, G. M., & Travise, S. (2016). *100 Excel Simulations: Using Excel to Model Risk, Investments, Genetics, Growth, Gambling and Monte Carlo Analysis*. Holy Macro! Books.
- Grigoryev, I. (2015). *AnyLogic 6 in three days: A quick course in simulation modeling*. Hampton, NJ: AnyLogic North America.
- Dimotikalis, I., Skiadas, C., & Skiadas, C. H. (2011). *Chaos theory: Modeling, simulation and applications: Selected papers from the 3rd Cghaotic Modeling and Simulation International Conference (CHAOS2010), Chania, Crete, Greece, 1-4 June, 2010*. Singapore: World Scientific.
- Velten, K. (2010). *Mathematical modeling and simulation: Introduction for scientists and engineers*. Weinheim: Wiley-VCH

Module 5: STOCHASTIC SIMULATIONS

Module Introduction

This module is divided into four (4) units

Unit 1: Stochastic Processes
Unit 2: Random Walks
Unit 3: Data Collection
Unit 4: Coding and Screening

Unit 1: Stochastic Processes

Contents

1.0 Introduction

So far in our work in probability and statistics, we have tended to deal with one (or at most two) random variables at a time. In many situations, we want to study the interaction of ―chance" with ―time" e.g. the behaviour of shares in a company on the stock market, the spread of an epidemic, the movement of a pollen grain in water (Brownian motion). To model this we, need a family of random variables (all defined on the same probability space), $(X(t); t \geq 0)$ where $X(t)$ represents e.g. the value of the share at time t. $(X(t); t \geq 0)$ is called a (continuous time) *stochastic process* or *random process*.

In this section, we will look at the theory of stochastic processes in an elementary manner. In probability theory, a **stochastic process**, or sometimes **random process**, is the counterpart to a deterministic process (or deterministic system). Instead of dealing with only one possible reality of how the process might evolve under time (as is the case, in solutions of an ordinary differential equation), in a stochastic or random process there is some indeterminacy in its future evolution described by probability distributions. This means that even if the initial condition (or starting point) is known, there are many possibilities the process might go to, but some paths

may be more probable and others less so.

2.0 Intended Learning Outcomes (ILOs)

By the end of this unit, the reader should be able to:

- Define Stochastic Process
- Classify Stochastic Processes
- Describe the Concepts in Stochastic Processes
- Show how stochastic processes are applied in different fields
- Describe the following processes: Ito, Levy, Wiener, Poisson, Point, Markov, Brownian, as stochastic process.

3.0 Main Content

3.1 Definition of Stochastic Process

A **stochastic process** is a probabilistic model of a system that evolves randomly in time and space. Formally, a stochastic process is a collection of random variables {X(t), t T} all defined on a common sample (probability) space. The X(t) is the state while (time) t is the index that is a member of set T.

Examples are the delay {D(i), i = 1, 2, ...} of the ith customer and number of customers {Q(t), T > 0} in the queue at time t in an M/M/1 queue. In the first example, we have a discrete-time, continuous state, while in the second example the state is discrete and time is continuous.

We can also define **stochastic process X** = {X(t), t ***T*** } as a collection of random variables. That is, for each t in the index set T, X(t) is a random variable. We often interpret t as time and call X(t) the state of the process at time t. when the index set ***T*** is countable set, we have a *discrete-time stochastic process*, or it is non-countable continuous set, we have a *continuous-time stochastic process*. Any realization of X is named a *sample path*, which can be discrete or continuous.

Although in most applications the index set is simply a set of time instants t_k, for the case of technical uncertainty this is not true.

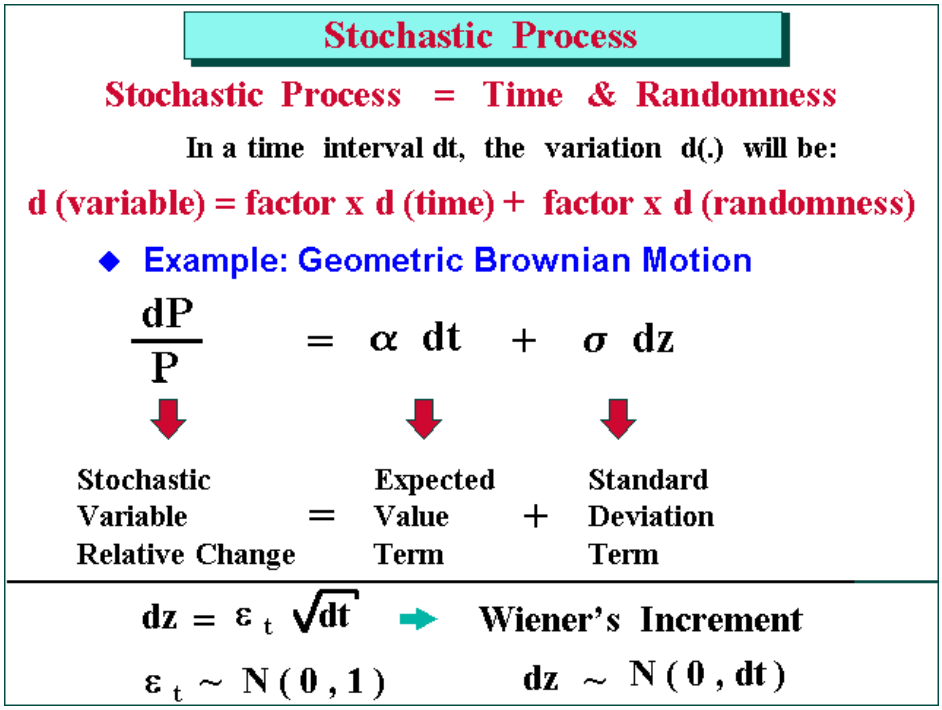
Another definition is that of Dixit & Pindyck's which state that "**Stochastic process** is a variable that evolves over time in a way that is at least in part random".

So, a stochastic process means time and randomness. In most cases, a stochastic variable has both a expected value term (drift term) and a random term (volatility term).

We can see the stochastic process forecasting for a random variable X, as a *forecast value (E[X]) plus a forecasting error*, where *error* follow some probability distribution. So:

$X(t) = E[X(t)] + error(t)$

The figure below presents the idea, a popular example, and brings the concept of ***increment*** (in this case the Wiener increment).



3.2 Classification of Stochastic Process

The following table is a classification of various stochastic processes. The man made systems have mostly discrete state. Monte Carlo simulation deals with discrete time while in discrete event system simulation, the time dimension is continuous.

A Classification of Stochastic Processes

		Change in the States of the System	
		Continuous	**Discrete**
Time	**Continuous**	Level of water behind a dam	Number of customers in a bank
	Discrete	Weekdays' range of temperature	Sales at the end of the day

3.3 General Stochastic Processes Concepts

To perform statistical analysis of the simulation output we need to establish some conditions, e.g. output data must be a covariance stationary process (e.g. the data collected over n simulation runs).

A **stationary** stochastic process is a stochastic process $\{X(t), t \varepsilon T\}$ with the property that the joint distribution of all vectors of h dimension remain the same for any fixed h.

First Order Stationary: A stochastic process is a first order stationary if the expectation of X(t) remains the same for all t.
For example in economic time series, a process is first order stationary when we remove any kinds of trend by some mechanisms such as differencing.

Second Order Stationary: A stochastic process is a second order stationary if it is first order stationary and covariance between X(t) and X(s) is a function of only t-s.
Again, in economic time series, a process is second order stationary when we stabilize also its variance by some kind of transformations such as taking square root.
Note: a stationary process is a second order stationary, however the reverse may not hold.
In simulation output statistical analysis we are satisfied if the output is *covariance stationary*.

Covariance Stationary: A covariance stationary process is a stochastic process $\{X(t), t \geq 0, T\}$ having finite second moments, i.e. expected of $[X(t)]^2$ be finite.
Clearly, any stationary process with finite second moment is covariance stationary. A stationary process may have no finite moment whatsoever.

Since a Gaussian process needs a mean and covariance matrix only, it is stationary (strictly) if it is covariance stationary.

Lévy processes are stochastic processes with stationary independent increments and continuous in probability.

Stationary increment property means that the probability distribution for the changes in the stochastic variable X, depends only on the time interval length.

Independent increments means that for all time instant t, the increments are independents.
The two most basic types of Lévy processes are Wiener processes and Poisson process.

Markov processes have the following property: given that its current state is known, the probability of any future event of the process is not altered by additional knowledge concerning its past behaviour.
In a more formal words, the probability distribution for $\mathbf{x_{t+1}}$ depends only on $\mathbf{x_t}$, and not additionally on what occurred before time t (doesn't depend of $\mathbf{x_s}$, where $\mathbf{s < t}$). that is suppose that: $\mathbf{P\{X_{n+1} = 1 | X_n = I, X_{n-1} = i_{n-1}, \ldots, X_1 = i_1, X_0 = i_0\} = P_{ij}}$
For all states $i_0, i_1, i_2, \ldots, i_{n-1}$, i, j and all n>=0, this stochastic process is a **Markov chain**.

Itô process is a generalized Wiener process. A ***Wiener*** process is also a special case of a ***strong diffusion process*** that is a particular class of a continuous time ***Markov*** process.
A continuous time ***Wiener Process*** (also called Brownian motion) is a stochastic process with three properties:

- It is a ***Markov process***. This means that all the past information is considered in the current value, so that futures values of the ***process depends only on its current value*** not on past values. The futures values are not affected by the past values history. In finance, this is

consistent with the efficient market hypothesis (that the current prices reflect all relevant information).

- It has ***independent increments***. Change in one time interval is independent of any other time interval (nonoverlapping).
- The changes of value over any finite time interval are ***normally distributed***. So, it has stationary increments, besides the property of independent increments. Therefore it is a particular ***Lévy process*** (Lévy process with normal distribution for the increments).

Point process is a stochastic process whose realizations are, instead continuous sample paths, counting measures.

Any counting process which is generated by an independent identically distributed (I.I.D) sum process (T_n) is called ***renewal counting process***.

The simplest and most fundamental point process is the ***Poisson process***, also referred as ***jump process*** in the financial literature.

The **Poisson process** is a counting process in which interarrival times of successive jumps are independently and identically distributed (i.i.d.) exponential random variables. The Poisson process is an example of renewal counting process.

Homogeneous Poisson Process has the following three properties:

1. It starts at zero. This means that at t = 0, there is no jump. By counting process point of view, N(0) = 0 (the number of jumps in the process N is zero at t = 0).
2. It has independent stationary increments. A Poisson process is also a special Lévy process.
3. For t > 0, the probability of n jumps occurrences until time t is:

$$\mathbf{P[N(t) = n] = (1/n!)\ (\lambda t)^n\ e^{-\lambda t}}\ ;\ n = 0, 1, 2, ...$$

which is a Poisson distribution with parameter t.

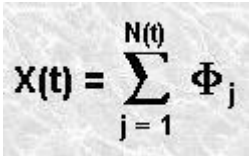
The Poisson distribution tends to the Normal distribution as the frequency tends to infinity. So, a Poisson distribution is *asymptotically Normal*.

A ***Non-homogeneous Poisson Process*** is more general than the homogeneous one: the stationary increment assumption is not required (remain the independent increment assumption), and the constant arrival rate of a Poisson process is replaced by a time- varying intensity function.

Compound Poisson Process:

Let $\mathbf{X_i}$ be a sequence of independent and identically distributed (I.I.D) random variables. These identical probability distributions can be interpreted as the jump-size distributions.

Let **N(t)** be a Poisson process, independent of X_i. The following process is called a Compound Poisson Process:



The sum of two independent compound Poisson processes is itself a compound Poisson process.

A ***jump is degenerate*** when the variable can only jump to a fixed value, and remains in this value. Example is the case of "sudden death" process which the variable drops to zero forever.

Interestingly the combination of Poisson processes with Brownian motions is related to Lévy process. According to Karlin & Taylor, "The general Lévy process can be represented as a sum of a Brownian motion, a uniform translation, and a limit (an integral) of a one-parameter family of compound Poisson processes, where all the contributing basic processes are mutually independent".

In the sample paths of Lévy processes, the large increments or "jumps" are called "**Lévy flights**".

3.4 Application of Stochastic Processes

Familiar examples of processes modeled as stochastic time series include stock market and exchange rate fluctuations, signals such as speech, audio and video, medical data such as a patient's EKG, EEG, blood pressure or temperature, and random movement such as Brownian motion or random walks.

Examples of random fields include static images, random terrain (landscapes).

3.4.1 Mathematical theory

The use of the term *stochastic* to mean *based on the theory of probability* has been traced back to Ladislaus Bortkiewicz, who meant the sense of *making conjectures* that the Greek term bears since ancient philosophers, and after the title of "Ars Conjectandi" that Bernoulli gave to his work on probability theory. In mathematics, specifically in probability theory, the field of stochastic processes has been a major area of research.

For example, a stochastic matrix is a matrix that has non-negative real entries that sum to one in each row.

3.4.2 Artificial intelligence

In artificial intelligence, stochastic programs work by using probabilistic methods to solve problems, as in stochastic neural networks, stochastic optimization, and genetic algorithms. A problem itself may be stochastic as well, as in planning under uncertainty. A deterministic environment is much simpler for an agent to deal with.

3.4.3 Natural science

An example of a stochastic process in the natural world is pressure in a gas as modelled by the Wiener process. Even though (classically speaking) each molecule is moving in a deterministic

path, the motion of a collection of them is computationally and practically unpredictable. A large set of molecules will exhibit stochastic characteristics, such as filling the container, exerting equal pressure, diffusing along concentration gradients, etc. These are emergent properties of the systems.

3.4.4 Physics

The name "Monte Carlo" for the stochastically Monte Carlo method was popularized by physics researchers.

Perhaps the most famous early use was by Enrico Fermi in 1930, when he used a random method to calculate the properties of the newly-discovered neutron. Monte Carlo methods were central to the simulations required for the Manhattan Project, though were severely limited by the computational tools at the time. Therefore, it was only after electronic computers were first built (from 1945 on) that Monte Carlo methods began to be studied in depth. In the 1950s they were used at Los Alamos for early work relating to the development of the hydrogen bomb, and became popularized in the fields of physics, physical chemistry, and operations research.

3.4.5 Biology

In biological systems, introducing stochastic 'noise' has been found to help improve the signal strength of the internal feedback loops for balance and other vestibular communication. It has been found to help diabetic and stroke patients with balance control.

3.4.6 Medicine

Stochastic effect or "chance effect" is one classification of radiation effects that refers to the random, statistical nature of the damage. In contrast to the deterministic effect, severity is independent of dose. Only the *probability* of an effect increases with dose. Cancer is a stochastic effect.

3.4.7 Creativity

Simonton (2003, Psych Bulletin) argues that creativity in science (of scientists) is a constrained stochastic behaviour such that new theories in all sciences are, at least in part, the product of stochastic processes.

3.4.8 Music

In music, **stochastic** elements are generated by strict mathematical processes.

Stochastic processes can be used in music to compose a fixed piece or can be produced in performance. Stochastic music was pioneered by Iannis Xenakis, who used probability, game theory, group theory, set theory, and Boolean algebra, and frequently used computers to produce his scores. Earlier, John Cage and others had composed *aleatoric* or indeterminate music, which is created by chance processes but does not have the strict mathematical basis.

3.4.9 Color reproduction

When color reproductions are made, the image is separated into its component colors by taking multiple photographs filtered for each color. One resultant film or plate represents each of the

cyan, magenta, yellow, and black data. Color printing is a binary system, where ink is either present or not present, so all color separations to be printed must be translated into dots at some stage of the work-flow. Traditional line screens which are amplitude modulated had problems until stochastic screening became available. A stochastic (or frequency modulated) dot pattern creates a sharper image.

3.4.10 Language and linguistics

Non-deterministic approaches in language studies are largely inspired by the work of Ferdinand de Saussure. In usage-based linguistic theories, for example, where it is argued that competence, or *langue*, is based on performance, or *parole*, in the sense that linguistic knowledge is based on frequency of experience, grammar is often said to be probabilistic and variable rather than fixed and absolute. This is so, because one's competence changes in accordance with one's experience with linguistic units. This way, the frequency of usage-events determines one's knowledge of the language in question.

3.4.11 Social Sciences

Stochastic social science theory is similar to systems theory in that events are interactions of systems, although with a marked emphasis on unconscious processes. The event creates its own conditions of possibility, rendering it unpredictable if simply for the amount of variables involved. Stochastic social science theory can be seen as an elaboration of a kind of 'third axis' in which to situate human behaviour alongside the traditional 'nature vs. nurture' opposition.

3.4.12 Business

- **Manufacturing -** Manufacturing processes are assumed to be stochastic processes. This assumption is largely valid for either continuous or batch manufacturing processes. Testing and monitoring of the process is recorded using a process control chart which plots a given process control parameter over time. Typically a dozen or many more parameters will be tracked simultaneously. Statistical models are used to define limit lines which define when corrective actions must be taken to bring the process back to its intended operational window.
- **Finance -** The financial markets use stochastic models to represent the seemingly random behaviour of assets such as stocks, commodities and interest rates. These models are then used by quantitative analysts to value options on stock prices, bond prices, and on interest rates, see Markov models. Moreover, it is at the heart of the insurance industry.

4.0 Self-Assessment Exercise(s)

Answer the following questions:

1. Briefly discuss the application of stochastic process in three fields
2. What are the properties of Wiener and Homogeneous Poisson processes?

5.0 Conclusion

In the simplest possible case (discrete time), a stochastic process amounts to a sequence of random variables known as a time series. Another basic type of a stochastic process is a random field, whose domain is a region of space, in other words, a random function whose arguments are drawn from a range of continuously changing values. One approach to stochastic processes treats them as functions of one or several deterministic arguments (inputs, in most cases regarded as time) whose values (outputs) are random variables: non-deterministic (single) quantities which have certain probability distributions. Random variables corresponding to various times (or points, in the case of random fields) may be completely different. The main requirement is that these different random quantities all have the same type. Although the random values of a stochastic process at different times may be independent random variables, in most commonly considered situations they exhibit complicated statistical correlations.

6.0 Summary

In this unit we:

- Defined Stochastic Process as a probabilistic model of a system that evolves randomly in time and space. Or a variable that evolves over time in a way that is at least in part random,
- Classify Stochastic Processes into Continuous or Discrete in; Time or Change in the State of the system.
- Discussed the following Concepts in Stochastic Processes: 1st, 2nd, and Covariance stationary, Levy, Ito, Point and Poisson processes.
- Highlighted the application of stochastic processes in different fields such as: Mathematics, Artificial intelligence, Natural science, Physics, Medicine, Business, etc.

7.0 Further Readings

- Gamerman, D., & Lopes, H. F. (2006). *Markov chain Monte Carlo: Stochastic simulation for Bayesian inference*. Boca Raton: Chapman & Hall/CRC.
- Graham, C., & Talay, D. (2015). *Stochastic Simulation and Monte Carlo Methods Mathematical Foundations of Stochastic Simulation*. Berlin: Springer Berlin Heidelberg.
- Mai, J., Scherer, M., & Czado, C. (2017). *Simulating copulas: Stochastic models, sampling algorithms, and applications*. New Jersey: World Scientific Publishing.
- Nelson, B. (2015). *Foundations and methods of stochastic simulation: A first course*. Place of publication not identified: Springer.
- Kozachenko, Y., Pogorilyak, O., Rozora, I., & Tegza, A. (2016). *Simulation of stochastic processes with given accuracy and reliability*. London: ISTE Press.
- Cochard, G. (2019). *Introduction to stochastic processes and simulation*. London, UK: ISTE.
- Rao, C. R., & Shanbhag, D. N. (2007). *Stochastic processes: Modelling and simulation.*

Amsterdam: Elsevier.
- Gallager, R. G. (2017). *Stochastic Processes: Theory for Applications*. Cambridge, United Kingdom: Cambridge University Press.
- Jones, P. W., & Smith, P. (2018). *Stochastic processes: An introduction*. Boca Raton (Fla.): Chapman & Hall/CRC.

Unit 2: Random Walks (RW)

Contents

1.0 Introduction

Random walk is a special case of stochastic process. This means that in this unit we are simply extending our knowledge of stochastic processes further.

2.0 Intended Learning Outcomes (ILOs)

By the end of this unit the reader should be able to:

- Define Random walks
- Explain various types of RW
- Illustrate different dimensions of RW with probabilities occurrence
- Relate RW to Wiener, Markov and Brownian processes, and
- List the applications of RW

3.0 Main Content

3.1 Definition of Random Walk

A **random walk**, sometimes denoted by **RW**, is a mathematical formalisation of a trajectory that consists of taking successive random steps.

The results of random walk analysis have been applied to computer science, physics, ecology, economics, psychology and a number of other fields as a fundamental model for random processes in time. For example, the path traced by a molecule as it travels in a liquid or a gas, the search path of a foraging animal, the price of a fluctuating stock and the financial status of a gambler can all be modeled as random walks. The term *random walk* was first introduced by Karl Pearson in 1905.

3.2 Types of Random Walks

Various different types of random walks are of interest. Often, random walks are assumed to be Markov chains or Markov processes, but other, more complicated walks are also of interest. Some random walks are on graphs, others on the line, in the plane, or in higher dimensions, while some random walks are on groups.

Random walks also vary with regard to the time parameter. Often, the walk is in discrete time, and indexed by the natural numbers, as in X_0, X_1, X_2, … However, some walks take their steps at random times, and in that case the position X_t is defined for the continuum of times $t \in [0,+\infty)$.

Specific cases or limits of random walks include the **drunkard's walk** and **Lévy flight**. Random walks are related to the diffusion models and are a fundamental topic in discussions of Markov processes. Several properties of random walks, including dispersal distributions, first-passage times and encounter rates, have been extensively studied

3.2.1 Lattice Random Walk

A popular random walk model is that of a random walk on a regular lattice (i.e. Network or Web), where at each step the walk jumps to another site according to some probability distribution.

In **simple random walk**, the walk can only jump to neighbouring sites of the lattice.

In **simple symmetric random walk** on a locally finite lattice, the probabilities of walk jumping to any one of its neighbours are the same. The most well-studied example is of random walk on the d-dimensional integer lattice (sometimes called the hypercubic lattice) $\mathbb{Z}^d$.

3.2.2 One-dimensional random walk

Imagine a one-dimensional length of something, a 'line' for example. Now imagine this line has numbers on it, spaced apart equally. A particularly elementary and concrete random walk is the random walk on the integer number line, Z, which starts at $S_o = 0$

and at each step moves by ±1 with equal probability. To define this walk formally, take

independent random variables Z1, Z2,….., where each variable is either 1 or −1, with a 50% probability for either value, and set

$$S_n := \sum_{j=1}^{n} Z_j.$$

The series {Sn} is called the **simple random walk on** Z. This series (the sum of the sequence of -1's and 1's) gives you the length you have 'walked', if each part of the walk is of length one.

This walk can be illustrated as follows. Say you flip a fair coin. If it lands on heads, you move one to the right on the number line. If it lands on tails, you move one to the left. So after five flips, you have the possibility of landing on 1, −1, 3, −3, 5, or −5. You can land on 1 by flipping three heads and two tails in any order. There are 10 possible ways of landing on 1. Similarly, there are 10 ways of landing on −1 (by flipping three tails and two heads), 5 ways of landing on 3 (by flipping four heads and one tail), 5 ways of landing on −3 (by flipping four tails and one head), 1 way of landing on 5 (by flipping five heads), and 1 way of landing on −5 (by flipping five tails). See the figure below for an illustration of this example.

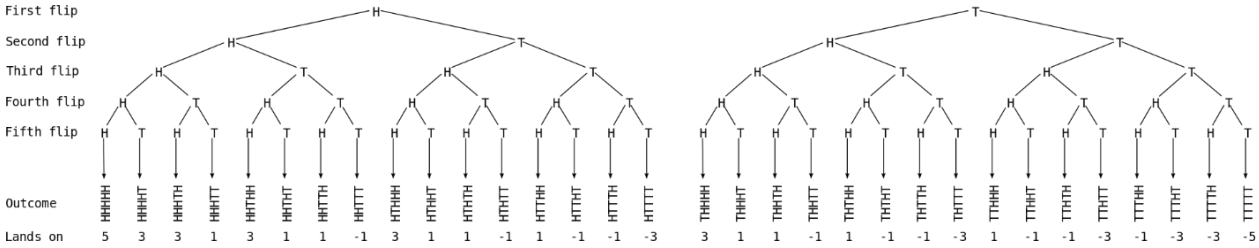


Figure 1: Five flips of a fair coin

This is of course random, so we cannot calculate it. But we may say quite a bit about its distribution. It is not hard to see that the expectation $E(S_n)$ of S_n is zero. That is, the more you flip the coin, the closer the mean of all your -1's and 1's will be to zero.

For example, this follows by the finite additivity property of expectation:

$$E(S_n) = \sum_{j=1}^{n} E(Z_j) = 0.$$

A similar calculation, using the independence of the random variables and the fact that $E(Z_n^2) = 1$, shows that:

$$E(S_n^2) = \sum_{i=1}^{n} E(Z_i^2) + 2 \sum_{1 \le i < j \le n} E(Z_i Z_j) = n.$$

This hints that $E(|S_n|)$, the expected translation distance after *n* steps, should be of the order of $\sqrt{n}$. In fact,

$$\lim_{n \to \infty} \frac{E(|S_n|)}{\sqrt{n}} = \sqrt{\frac{2}{\pi}}.$$

Suppose that you are some distance from the origin of the walk. How many times will the

random walk cross the line if permitted to continue walking forever?
The following, perhaps surprising theorem is the answer: A simple random walk on Z will cross every point an infinite number of times. This result has many names: the *level-crossing phenomenon*, *recurrence* or the *gambler's ruin*.
The reason for the last name is as follows: if you are a gambler with a finite amount of money playing *a fair game* against a bank with an infinite amount of money, you will surely lose. The amount of money you have will perform a random walk, and it will almost surely, at some time, reach 0 and the game will be over.
If a and b are positive integers, then the expected number of steps until a one-dimensional simple random walk starting at 0 first hits b or $-a$ is ab. The probability that this walk will hit b before $-a$ steps is $(a / (a + b))$, which can be derived from the fact that simple random walk is a martingale.

Some of the results mentioned above can be derived from properties of Pascal's triangle. The number of different walks of n steps where each step is +1 or −1 is clearly 2^n. For the simple random walk, each of these walks are equally likely. In order for S_n to be equal to a number k it is necessary and sufficient that the number of +1 in the walk exceeds those of −1 by k. Thus, the number of walks which satisfy $S_n = k$ is precisely the number of ways of choosing $(n + k)/2$ elements from an n element set (for this to be non-zero, it is necessary that $n + k$ be an even number), which is an entry in Pascal's triangle denoted by:

$$\binom{n}{(n+k)/2}.$$

Therefore, the probability that $S_n = k$ is equal to: $2^{-n}\binom{n}{(n+k)/2}$.

By representing entries of Pascal's triangle in terms of factorials and using Stirling's formula, one can obtain good estimates for these probabilities for large values of n.

This relation with Pascal's triangle is easily demonstrated for small values of n. At zero turns, the only possibility will be to remain at zero. However, at one turn, you can move either to the left or the right of zero, meaning there is one chance of landing on −1 or one chance of landing on 1. At two turns, you examine the turns from before. If you had been at 1, you could move to 2 or back to zero. If you had been at −1, you could move to −2 or back to zero. So there is one chance of landing on −2, two chances of landing on zero, and one chance of landing on 2.

n	**-5**	**-4**	**-3**	**-2**	**-1**	**0**	**1**	**2**	**3**	**4**	**5**
$P[S_0 = k]$						1					
$2P[S_1 = k]$					1		1				
$2^2P[S_2 = k]$				1		2		1			
$2^3P[S_3 = k]$			1		3		3		1		
$2^4P[S_4 = k]$		1		4		6		4		1	

$2^5 P[S_5 = k]$ 1 5 10 10 5 1

The central limit theorem and the law of the iterated logarithm describe important aspects of the behaviour of simple random walk on Z.

3.2.3 Gaussian random walk

Gaussian random walk is a random walk having a step size that varies according to a normal distribution. It is used as a model for real-world time series data such as financial markets. The Black-Scholes formula for modelling equity option prices, for example, uses a gaussian random walk as an underlying assumption.

A Stochastic process {X(t), t>=0} is called a Gaussian, or a normal process if X(tn) has a multivariate normal distribution for all t1, t2, …, tn.

If {X(t), t>=0} is a Brownian motion process , then because each of X(t1), X(t2), …, X(tn) can be expressed as a linear combination of independent normal random variables X(t1), X(t2)-X(t1), X(t3)-X(t2), …, X(tn)-X(tn-1) it follows that Brownian motion is Gaussian process.

Here, the step size is the inverse cumulative normal distribution $\Phi^{-1}(z,\mu,\zeta)$ where $0 \leq z \leq 1$ is a uniformly distributed random number, and μ and ζ are the mean and standard deviations of the normal distribution, respectively.

For steps distributed according to any distribution with zero mean and a finite variance (not necessarily just a normal distribution), the root mean squared translation distance after *n* steps is:

$$\sqrt{E|S_n^2|} = \sigma\sqrt{n}.$$

3.3 Random walk in two dimensions.

Random walk in two dimensions with more, and smaller, steps. In the limit, for very small steps, one obtains Brownian motion.
A Stochastic process {X(t), t>=0} is said to be a Brownian motion process if:

- X(0) = 0;
- {X(t), t>=0} has stationary and independent increments
- For every t>0, X(t) is normally distributed with mean 0 and variance $\zeta^2 t$.

Imagine now a drunkard walking randomly in an idealized city. The city is effectively infinite and arranged in a square grid, and at every intersection, the drunkard chooses one of the four possible routes (including the one he came from) with equal probability. Formally, this is a random walk on the set of all points in the plane with integer coordinates. Will the drunkard ever get back to his home from the bar? It turns out that he will. This is the high dimensional equivalent of the level crossing problem discussed above. The probability of returning to the

origin decreases as the number of dimensions increases. In three dimensions, the probability decreases to roughly 34%. A derivation, along with values of $p(d)$ are discussed in:
The **trajectory** of a random walk is the collection of sites it visited, considered as a set with disregard to *when* the walk arrived at the point. In one dimension, the trajectory is simply all points between the minimum height the walk achieved and the maximum (both are, on average, on the order of $\sqrt{n}$). In higher dimensions the set has interesting geometric properties. In fact, one gets a discrete fractal, that is a set which exhibits stochastic self- similarity on large scales, but on small scales one can observe "jaggedness" resulting from the grid on which the walk is performed.

3.4 Random walk on graphs

Assume now that our city is no longer a perfect square grid. When our drunkard reaches a certain junction he picks between the various available roads with equal probability. Thus, if the junction has seven exits the drunkard will go to each one with probability of one seventh. This is a random walk on a graph.

Will our drunkard reach his home? It turns out that under rather mild conditions, the answer is still yes. For example, if the lengths of all the blocks are between a and b (where a and b are any two finite positive numbers), then the drunkard will, almost surely, reach his home. Notice that we do not assume that the graph is planar, i.e. the city may contain tunnels and bridges. One way to prove this result is using the connection to electrical networks. Take a map of the city and place a one ohm resistor on every block. Now measure the "resistance between a point and infinity". In other words, choose some number R and take all the points in the electrical network with distance bigger than R from our point and wire them together. This is now a finite electrical network and we may measure the resistance from our point to the wired points. Take R to infinity. The limit is called the *resistance between a point and infinity*. It turns out that the following is true (an elementary proof can be found in the book by Doyle and Snell):

Theorem: *a graph is transient if and only if the resistance between a point and infinity is finite. It is not important which point is chosen if the graph is connected.*

In other words, in a transient system, one only needs to overcome a finite resistance to get to infinity from any point. In a recurrent system, the resistance from any point to infinity is infinite.

This characterization of recurrence and transience is very useful, and specifically it allows us to analyze the case of a city drawn in the plane with the distances bounded.

A random walk on a graph is a very special case of a Markov chain. Unlike a general Markov chain, random walk on a graph enjoys a property called *time symmetry* or *reversibility*. Roughly speaking, this property, also called the principle of detailed balance, means that the probabilities to traverse a given path in one direction or in the other have a very simple connection between them (if the graph is regular, they are just equal). This property has important consequences.

A good reference for random walk on graphs is the online book by Aldous and Fill. For groups see the book of Woess. If the transition kernel $p(x,y)$ is itself random (based on an environment ω) then the random walk is called a "random walk in random environment". When the law of the random walk includes the randomness of ω, the law is called the ***annealed law***; on the other hand, if ω is seen as fixed, the law is called a ***quenched law***.

3.5 Wiener Process

A **Wiener process** is a stochastic process with similar behaviour to Brownian motion, the physical phenomenon of a minute particle diffusing in a fluid. (Sometimes the Wiener process is called "Brownian motion", although this is strictly speaking a confusion of a model with the phenomenon being modeled.)

A Wiener process is the scaling limit of random walk in dimension 1. This means that if you take a random walk with very small steps you get an approximation to a Wiener process (and, less accurately, to Brownian motion).

To be more precise, if the step size is ε, one needs to take a walk of length L/ε^2 to approximate a Wiener process walk of length L. As the step size tends to 0 (and the number of steps increases proportionally) random walk converges to a Wiener process in an appropriate sense.

Formally, if B is the space of all paths of length L with the maximum topology, and if M is the space of measure over B with the norm topology, then the convergence is in the space M. Similarly, a Wiener process in several dimensions is the scaling limit of random walk in the same number of dimensions.

A random walk is a discrete fractal (a function with integer dimensions; 1, 2, ...), but a Wiener process trajectory is a true fractal, and there is a connection between the two. For example, take a random walk until it hits a circle of radius r times the step length. The average number of steps it performs is r^2. This fact is the *discrete version* of the fact that a Wiener process walk is a fractal of Hausdorff dimension. In two dimensions, the average number of points the same random walk has on the *boundary* of its trajectory is $r^{4/3}$. This corresponds to the fact that the boundary of the trajectory of a Wiener process is a fractal of dimension 4/3, a fact predicted by Mandelbrot using simulations but proved only in 2000 (Science, 2000).

A Wiener process enjoys many symmetries which random walk does not. For example, a Wiener process walk is invariant to rotations, but random walk is not, since the underlying grid is not. This means that in many cases, problems on random walk are easier to solve by translating them to a Wiener process, solving the problem there, and then translating back. On the other hand, some problems are easier to solve with random walks due to its discrete nature.

Random walk and Wiener process can be *coupled*, namely manifested on the same

probability space in a dependent way that forces them to be quite close. The simplest such coupling is the Skorokhod embedding, but other, more precise couplings exist as well.
The ***convergence*** of a random walk toward the Wiener process is controlled by the central limit theorem. For a particle in a known fixed position at $t = 0$, the theorem tells us that after a large number of independent steps in the random walk, the walker's position is distributed according to a normal distribution of total variance:

$$\sigma^2 = \frac{t}{\delta t}\varepsilon^2,$$

where t is the time elapsed since the start of the random walk , ε is the size of a step of the random walk, and δt is the time elapsed between two successive steps.

This corresponds to the Green function of the diffusion equation that controls the Wiener process, which demonstrates that, after a large number of steps, the random walk converges toward a Wiener process.

In 3D, the variance corresponding to the Green's function of the diffusion equation is:

$$\sigma^2 = 6\,D\,t.$$

By equalizing this quantity with the variance associated to the position of the random walker, one obtains the equivalent diffusion coefficient to be considered for the asymptotic Wiener process toward which the random walk converges after a large number of steps:

$$D = \frac{\varepsilon^2}{6\delta t} \text{ (valid only in 3D).}$$

Remark: the two expressions of the variance above correspond to the distribution associated to the vector ***R*** that links the two ends of the random walk, in 3D. The variance associated to each component R_x, R_y or R_z is only one third of this value (still in 3D).

3.6 Applications of Random Walks

The following are the applications of random walks:

- In *economics*, the "random walk hypothesis" is used to model shares prices and other factors. Empirical studies found some deviations from this theoretical model, especially in short term and long term correlations.
- In *population genetics*, random walk describes the statistical properties of genetic drift
- In *physics*, random walks are used as simplified models of physical Brownian motion and the random movement of molecules in liquids and gases. See for example diffusion-limited aggregation. Also in physics, random walks and some of the self interacting walks play a role in quantum field theory.
- In *mathematical ecology*, random walks are used to describe individual animal movements, to empirically support processes of biodiffusion, and occasionally to model population dynamics.

- In *polymer physics*, random walk describes an ideal chain. It is the simplest model to study polymers.
- In other fields of mathematics, random walk is used to calculate solutions to Laplace's equation, to estimate the harmonic measure, and for various constructions in analysis and combinatorics.
- In *computer science*, random walks are used to estimate the size of the Web. In the World Wide Web conference-2006, bar-yossef et al. published their findings and algorithms for the same. (This was awarded the best paper for the year 2006).
- In *image segmentation*, random walks are used to determine the labels (i.e., "object" or "background") to associate with each pixel[3]. This algorithm is typically referred to as the random walker segmentation algorithm.

- In *brain research*, random walks and reinforced random walks are used to model cascades of neuron firing in the brain.
- In vision science, fixational eye movements are well described by a random walk.
- In psychology, random walks explain accurately the relation between the time needed to make a decision and the probability that a certain decision will be made.
- Random walk can be used to sample from a state space which is unknown or very large, for example to pick a random page off the internet or, for research of working conditions, a random worker in a given country.

- When this last approach is used in computer science it is known as Markov Chain Monte Carlo or MCMC for short. Often, sampling from some complicated state space also allows one to get a probabilistic estimate of the space's size. The estimate of the permanent of a large matrix of zeros and ones was the first major problem tackled using this approach.
- In *wireless networking*, random walk is used to model node movement.
- Motile bacteria engage in a biased random walk.
- Random walk is used to model gambling.
- In *physics*, random walks underlying the method of Fermi estimation.
- During *World War II* a random walk was used to model the distance that an escaped prisoner of war would travel in a given time.

3.7 Probabilistic interpretation

A one-dimensional **random walk** can also be looked at as a Markov chain whose state space is given by the integers $i = 0, \pm 1, \pm 2, \ldots$, for some number p satisfying $0 < p < 1$. We can call it a **random walk** because we may think of it asbeing a model for an individual walking on a straight line who at each point of time either takes one step to the right with probability p or one step to the left with probability $1 - p$.

 **4.0 Self-Assessment Exercise(s)**

Answer the following questions:

1. What is skorokhod embedding in the coupling of Random walk and wiener process?
2. Define Random Walks and state the relationships between Random walk and Wiener process.
3. Differentiate between Annealed and Quenched laws in Random walks

5.0 Conclusion

We presented in this unit, the theory of stochastic processes in an elementary manner. A reader interested in a more rigorous approach could consult some of the referenced texts and many other sources. This unit tried to limit the discussion to a more physical description of stochastic processes.

6.0 Summary

In this unit we:

- Defined Random Walk as a mathematical formalisation of a trajectory that consists of taking successive random steps, which is often assumed to be Markov chains or Markov processes.
- Discussed different types of Random Walks especially, in: the One-Dimensional, Gaussian, two dimensions and Graphs
- Look at the applications of random walk and
- Explained the probability interpretations and
- Describedthe relationship of RW to Weiner process.

7.0 Further Readings

- Gamerman, D., & Lopes, H. F. (2006). *Markov chain Monte Carlo: Stochastic simulation for Bayesian inference*. Boca Raton: Chapman & Hall/CRC.
- Graham, C., & Talay, D. (2015). *Stochastic Simulation and Monte Carlo Methods Mathematical Foundations of Stochastic Simulation*. Berlin: Springer Berlin Heidelberg.
- Mai, J., Scherer, M., & Czado, C. (2017). *Simulating copulas: Stochastic models, sampling algorithms, and applications*. New Jersey: World Scientific Publishing.
- Nelson, B. (2015). *Foundations and methods of stochastic simulation: A first course*. Place of publication not identified: Springer.
- Kozachenko, Y., Pogorilyak, O., Rozora, I., & Tegza, A. (2016). *Simulation of stochastic processes with given accuracy and reliability*. London: ISTE Press.
- Cochard, G. (2019). *Introduction to stochastic processes and simulation*. London, UK: ISTE.

- Rao, C. R., & Shanbhag, D. N. (2007). *Stochastic processes: Modelling and simulation*. Amsterdam: Elsevier.
- Gallager, R. G. (2017). *Stochastic Processes: Theory for Applications*. Cambridge, United Kingdom: Cambridge University Press.
- Jones, P. W., & Smith, P. (2018). *Stochastic processes: An introduction*. Boca Raton (Fla.): Chapman & Hall/CRC.

Unit 3: Data Collection

Contents

1.0 Introduction

The purpose of most simulation models is to collect data and to analyse the data, in order to gain insights into the system being simulated. Thus we literally _play' with data in the conduct of _what if scenarios' to enable conclusions and decisions to be made. To derive good conclusions from data, we do not just use any data, care and efforts are usually made to collect usable data from select at times _restricted' source(s). Therefore before decisions are made base on statistics from data, we need to know how the data were collected; that is, we need to know the method(s) of data collection, and the necessary screening on the data that has taken place.

2.0 Intended Learning Outcomes (ILOs)

The reader should be able by the end of this unit to:

- Discuss the different methods of data collection: Census, Sample Survey, Experiments and Observations
- Describe the various methods determining sample size
- Describe data coding with respect to: What, Why, uses and determination of codes

3.0 Main Content

3.1 Methods of Data Collection

There are four main methods of data collection.

- **Census**: A census is a study that obtains data from every member of a population. In most studies, a census is not practical, because of the cost and/or time required.
- **Sample survey**: A sample survey is a study that obtains data from a subset of a population, in order to estimate population attributes.
- **Experiment**: An experiment is a controlled study in which the researcher attempts to understand cause-and-effect relationships. The study is "controlled" in the sense that the researcher controls (1) how subjects are assigned to groups and (2) which treatments each group receives.

In the analysis phase, the researcher compares group scores on some dependent variable. Based on the analysis, the researcher draws a conclusion about whether the treatment (independent variable) had a causal effect on the dependent variable.

- **Observational study**: Like experiments, observational studies attempt to understand cause-and-effect relationships. However, unlike experiments, the researcher is not able to control (1) how subjects are assigned to groups and/or (2) which treatments each group receives.

3.2 Pros and Cons of Data Collection Methods

Each method of data collection has advantages and disadvantages.

- **Resources**. When the population is large, a sample survey has a big resource advantage over a census. A well-designed sample survey can provide very precise estimates of population parameters - quicker, cheaper, and with less manpower than a census.
- **Generalizability**. Generalizability refers to the appropriateness of applying findings from a study to a larger population. Generalizability requires random selection. If participants in a study are randomly selected from a larger population, it is appropriate to generalize study results to the larger population; if not, it is not appropriate to generalize. Observational studies do not feature random selection; so it is not appropriate to generalize from the results of an observational study to a larger population.
- **Causal inference**. Cause-and-effect relationships can be teased out when subjects are randomly assigned to groups. Therefore, experiments, which allow the researcher to control assignment of subjects to treatment groups, are the best method for investigating causal relationships.

3.3 Sampling Survey Methods

Sampling method refers to the way that observations are selected from a population to be in the sample for a sample survey.

The reason for conducting a sample survey is to estimate the value of some attribute of a population.

- **Population parameter**. A population parameter is the true value of a population attribute
- **Sample statistic**. A sample statistic is an estimate, based on sample data, of a population

parameter.

Consider this example. A public opinion pollster wants to know the percentage of voters that favor a flat-rate income tax. The *actual* percentage of all the voters is a population parameter. The *estimate* of that percentage, based on sample data, is a sample statistic.

The quality of a sample statistic (i.e., accuracy, precision, representativeness) is strongly affected by the way those sample observations are chosen; that is., by the sampling method.

3.4 Categorization of Sampling Methods

As a group, sampling methods fall into one of two categories.

- **Probability samples**. With probability sampling methods, each population element has a known (non-zero) chance of being chosen for the sample.
- **Non-probability samples**. With non-probability sampling methods, we do not know the probability that each population element will be chosen, and/or we cannot be sure that each population element has a non-zero chance of being chosen.

Non-probability sampling methods offer two potential advantages - convenience and cost. The main disadvantage is that non-probability sampling methods do not allow you to estimate the extent to which sample statistics are likely to differ from population parameters. Only probability sampling methods permit that kind of analysis.

3.5 Non-Probability Sampling Methods

Two of the main types of non-probability sampling methods are voluntary samples and convenience samples.

- **Voluntary sample**. A voluntary sample is made up of people who self-select into the survey. Often, these folks have a strong interest in the main topic of the survey. For example, that a news show that asks viewers to participate in an on-line poll. This is a volunteer sample because the sample is chosen by the viewers, not by the survey administrator.

- **Convenience sample**. A convenience sample is made up of people who are easy to reach.

Consider the following example. A pollster interviews shoppers at a local mall. If the mall was chosen because it was a convenient site from which to solicit survey participants and/or because it was close to the pollster's home or business, this would be a convenience sample.

3.6 Probability Sampling Methods

The main types of probability sampling methods are *simple random sampling, stratified sampling, cluster sampling, multistage sampling, and systematic random sampling*. The key benefit of probability sampling methods is that they guarantee that the sample chosen is representative of the population. This ensures that the statistical conclusions will be valid.

Simple random sampling.

Simple random sampling refers to any sampling method that has the following properties.

- The population consists of N objects.
- The sample consists of n objects.
- If all possible samples of n objects are equally likely to occur, the sampling method is called simple random sampling.

There are many ways to obtain a simple random sample. One way would be the lottery method. Each of the N population members is assigned a unique number. The numbers are placed in a bowl and thoroughly mixed. Then, a blind-folded researcher selects n numbers. Population members having the selected numbers are included in the sample.

a. Stratified sampling Method (SSM)

With stratified sampling, the population is divided into groups, based on some characteristic. Then, within each group, a probability sample (often a simple random sample) is selected. In stratified sampling, the groups are called **strata**. As a example, suppose we conduct a national survey. We might divide the population into groups or strata, based on geography - north, east, south, and west. Then, within each stratum, we might randomly select survey respondents.

b. Cluster sampling

With cluster sampling, every member of the population is assigned to one, and only one, group. Each group is called a cluster. A sample of clusters is chosen, using a probability method (often simple random sampling). Only individuals within sampled clusters are surveyed.

Note the difference between cluster sampling and stratified sampling. With stratified sampling, the sample includes elements from each stratum. With cluster sampling, in contrast, the sample includes elements only from sampled clusters.

c. Multistage Sampling

With multistage sampling, we select a sample by using combinations of different sampling methods.

For example, in Stage 1, we might use cluster sampling to choose clusters from a population. Then, in Stage 2, we might use simple random sampling to select a subset of elements from each chosen cluster for the final sample.

d. Systematic Random Sampling

With systematic random sampling, we create a list of every member of the population. From the list, we randomly select the first sample element from the first *k* elements on the population list. Thereafter, we select every *kth* element on the list.

This method is different from simple random sampling since every possible sample of *n*

elements is not equally likely.

3.7 Experiments Method in Data Collection

In an experiment, a researcher manipulates one or more variables, while holding all other variables constant. By noting how the manipulated variables affect a response variable, the researcher can test whether a causal relationship exists between the manipulated variables and the response variable.

3.7.1 Parts of an Experiment

All experiments have independent variables, dependent variables, and experimental units.

a. Independent variable. An independent variable (also called a **factor**) is an explanatory variable manipulated by the experimenter.

Each factor has two or more **levels**, i.e., different values of the factor. Combinations of factor levels are called **treatments**. The table below shows independent variables, factors, levels, and treatments for a hypothetical experiment.

		Vitamin C		
		0 mg	250 mg	500 mg
Vitamin E	0 mg	Treatment 1	Treatment 2	Treatment 3
	400 mg	Treatment 4	Treatment 5	Treatment 6

In this hypothetical experiment, the researcher is studying the possible effects of Vitamin C and Vitamin E on health. There are two factors - dosage of Vitamin C and dosage of Vitamin E. The Vitamin C factor has 3 levels - 0 mg per day, 250 mg per day, and 500 mg per day. The Vitamin E factor has 2 levels - 0 mg per day and 400 mg per day. The experiment has six treatments. Treatment 1 is 0 mg of E and 0 mg of C, Treatment 2 is 0 mg of E and 250 mg of C, and so on.

b. Dependent variable. In the hypothetical experiment above, the researcher is looking at the effect of vitamins on health. The dependent variable in this experiment would be some measure of health (annual doctor bills, number of colds caught in a year, number of days hospitalized, etc.).

c. Experimental units. The recipients of experimental treatments are called experimental units. The experimental units in an experiment could be anything - people, plants, animals, or even inanimate objects.

In the hypothetical experiment above, the experimental units would probably be people (or lab animals). But in an experiment to measure the tensile strength of string, the experimental units might be pieces of string. When the experimental units are people, they are often called **participants**; when the experimental units are animals, they are often called **subjects**.

3.7.2 Characteristics of a Well-Designed Experiment

A well-designed experiment includes design features that allow researchers to eliminate extraneous variables as an explanation for the observed relationship between the independent variable(s) and the dependent variable. Some of these features are listed below.

1. **Control**. Control refers to steps taken to reduce the effects of extraneous variables (i.e., variables other than the independent variable and the dependent variable). These extraneous variables are called **lurking variables**.

Control involves making the experiment as similar as possible for experimental units in each treatment condition. Three control strategies are control groups, placebos, and blinding.

- Control group. A control group is a baseline group that receives no treatment or a neutral treatment. To assess treatment effects, the experimenter compares results in the treatment group to results in the control group.
- Placebo. Often, participants in an experiment respond differently after they receive a treatment, even if the treatment is neutral. A neutral treatment that has no "real" effect on the dependent variable is called a **placebo**, and a participant's positive response to a placebo is called the **placebo effect**.

To control for the placebo effect, researchers often administer a neutral treatment (i.e., a placebo) to the control group. The classic example is using a sugar pill in drug research. The drug is effective only if participants who receive the drug have better outcomes than participants who receive the sugar pill.

- Blinding. Of course, if participants in the control group know that they are receiving a placebo, the placebo effect will be reduced or eliminated; and the placebo will not serve its intended control purpose.

Blinding is the practice of not telling participants whether they are receiving a placebo. In this way, participants in the control and treatment groups experience the placebo effect equally. Often, knowledge of which groups receive placebos is also kept from people who administer or evaluate the experiment. This practice is called ***double blinding***. It prevents the experimenter from "spilling the beans" to participants through subtle cues; and it assures that the analyst's evaluation is not tainted by awareness of actual treatment conditions.

2. **Randomization**. Randomization refers to the practice of using chance methods (random number tables, flipping a coin, etc.) to assign experimental units to treatments. In this way, the potential effects of lurking variables are distributed at chance levels (hopefully roughly evenly) across treatment conditions.

3. **Replication**. Replication refers to the practice of assigning each treatment to many experimental units. In general, the more experimental units in each treatment condition, the lower the variability of the dependent measures.

3.8 Confounding

Confounding occurs when the experimental controls do not allow the experimenter to

reasonably eliminate plausible alternative explanations for an observed relationship between independent and dependent variables.

Consider this example. A drug manufacturer tests a new cold medicine with 200 participants - 100 men and 100 women. The men receive the drug, and the women do not. At the end of the test period, the men report fewer colds.

This experiment implements no controls at all! As a result, many variables are confounded, and it is impossible to say whether the drug was effective. For example, gender is confounded with drug use. Perhaps, men are less vulnerable to the particular cold virus circulating during the experiment, and the new medicine had no effect at all. Or perhaps the men experienced a placebo effect.

This experiment could be strengthened with a few controls. Women and men could be randomly assigned to treatments. One treatment could receive a placebo, with blinding. Then, if the treatment group (i.e., the group getting the medicine) had sufficiently fewer colds than the control group, it would be reasonable to conclude that the medicine was effective in preventing colds.

 **4.0 Self-Assessment Exercise(s)**

Answer the following questions:

1. Which of the following statements are true and why?

I. Blinding controls for the effects of confounding.

II. Randomization controls for effects of lurking variables.

III. Each factor has one treatment level.

(A) I only (B) II only (C) III only (D) All of the above. (E) None of the above.

2. An auto analyst is conducting a satisfaction survey, sampling from a list of 10,000 new car buyers. The list includes 2,500 Ford buyers, 2,500 GM buyers, 2,500 Honda buyers, and 2,500 Toyota buyers. The analyst selects a sample of 400 car buyers, by randomly sampling 100 buyers of each brand.

Is this an example of a simple random sample and state why?

3. Which of the following statements is/are true and why?

I. A sample survey is an example of an experimental study.

II. An observational study requires fewer resources than an experiment.

III. The best method for investigating causal relationships is an observational study.

(A) I only (B) II only (C) III only (D) all of the above (E) none of the above.

4. Distinguish between the following sampling methods: Stratified, Clusterand Multistage

sampling.

5.0 Conclusion

Model Simulations is simply playing with relevant data to learn and make decision. The result or accuracy of such decision which may be far reaching is subject to the quality of data. This high quality and thus good decision is achieved through conscious choice, planning and execution of data collection method.

6.0 Summary

In this unit, we discussed:

- The various methods of data collection including:
- **Census**- is a study that obtains data from every member of a population which in most case is not practical, because of the cost and/or time required.
- **Sample survey**- is a study that obtains data from a subset of a population, in order to estimate population attributes.
- **Experiment**- is a controlled study in which the researcher attempts to understand cause-and-effect relationships. Whereby the study is "controlled" in the sense that the researcher controls: how subjects are assigned to groups and which treatments each group receives.

In the analysis phase, the researcher compares group scores on some dependent variable. Based on the analysis, the researcher draws a conclusion about whether the treatment (independent variable) had a causal effect on the dependent variable.

- **Observational study –** which attempt to understand cause-and-effect relationships. But however, unlike experiments, the researcher is not able to control: how subjects are assigned to groups and/or and which treatments each group receives.
- Indepth the various methods by futher subdividing them as follows:
  - Sampling survey into: Simple, Stratified, Cluster, Systematic random and multistage
  - Experiments into: Independent, Dependent and Experimental variable types and the characteristics of EM: Controlled, Randomized and Replicated experiments and Confounding.

7.0 Further Readings

- Gamerman, D., & Lopes, H. F. (2006). *Markov chain Monte Carlo: Stochastic simulation*

for Bayesian inference. Boca Raton: Chapman & Hall/CRC.

- Graham, C., & Talay, D. (2015). *Stochastic Simulation and Monte Carlo Methods Mathematical Foundations of Stochastic Simulation*. Berlin: Springer Berlin Heidelberg.
- Mai, J., Scherer, M., & Czado, C. (2017). *Simulating copulas: Stochastic models, sampling algorithms, and applications*. New Jersey: World Scientific Publishing.
- Nelson, B. (2015). *Foundations and methods of stochastic simulation: A first course*. Place of publication not identified: Springer.
- Kozachenko, Y., Pogorilyak, O., Rozora, I., & Tegza, A. (2016). *Simulation of stochastic processes with given accuracy and reliability*. London: ISTE Press.
- Cochard, G. (2019). *Introduction to stochastic processes and simulation*. London, UK: ISTE.
- Rao, C. R., & Shanbhag, D. N. (2007). *Stochastic processes: Modelling and simulation*. Amsterdam: Elsevier.
- Gallager, R. G. (2017). *Stochastic Processes: Theory for Applications*. Cambridge, United Kingdom: Cambridge University Press.
- Jones, P. W., & Smith, P. (2018). *Stochastic processes: An introduction*. Boca Raton (Fla.): Chapman & Hall/CRC.

Unit 4: Data Coding and Screening

Contents

1.0 Introduction

We have earlier state that simulation is about play with data to study the behaviour of a system. To actually benefit from such exercise, true representative data must be collected and screen. Quality data comes through careful transformation from its raw stages into appropriate system parameters variables which can be quantitatively analyzed before it can yield true results.

2.0 Intended Learning Outcomes (ILOs)

By the end of this unit, the reader should be able to:

- Define and explain data coding
- Describe the steps involved in coding
- Explain code determination
- Describe the outlier and how to handle it

3.0 Main Content

3.1 Data Coding

Data Coding is a systematic way used to condense extensive data-sets into smaller analyzable units through the creation of categories and concepts derived from the data. It may also be

described as the process by which data are converted into variables and categories of variables using numbers, so that the data can be entered into computers for analysis.

3.2 Framework for data Coding

In order to begin coding your data for your plan and report, read your interview and field notes transcripts looking for relationships, themes, and concerns that are relevant to your professional context.

Use the following framework to guide the marking and coding of your data. The following general steps outline the process through which you will be working:

1. Read through data.
2. Reread data marking significant points.
3. Reread data sorting, categorizing, and coding significant points in terms of issues. (Note: Successful coding often requires multiple runs through your data. Thus, repeat this step if necessary.)
4. Insert coded data into the downloadable "Data Coding Grid" form.
5. Select a manageable set of issues for analysis.
6. Draft your plan based upon the results of your coding.

Step 2: marking your data

After carefully reading through your data at least once (Step 1), you need to mark your data. That is, underline, highlight, or circle points that seem significant or relevant to your professional context. In marking your data, consider:

- Points that are mentioned consistently across your data.
- Points that are significant anomalies--that are important because theyindicate special or unique circumstances, standards, or procedures.
- Points that contradict one another.
- Connections (among persons, sites, documents, procedures, objects, etc.) that are made explicitly.
- Connections or traces (among persons, sites, documents, procedures, objects, etc.) that are implied.

Step 3: coding your data

After marking your data for significant points and connections, you need to reread your data, sorting and categorizing the points for more specific issues for analysis. That is, you should develop a system of notations or symbols and literately code the marks in your notes and transcripts using this system. In identifying relationships, themes, and concerns that are relevant to your professional context, code for issues such as:

- power/authority,
- knowledge/expertise,
- status,

- worker-worker or student-student relationships,
- management-worker or teacher-student relationships,
- initiation of contact or discourse, and
- completion of contact or discourse.

Step 4: inserting coded data into "data coding grid"

After coding all of your data, download the "Data Coding Grid" form. Then, cut and paste coded examples from your data into the appropriate categories in the grid.

As noted above, some examples will fit into more than one column of your grid. Once you have plotted your coded data into these issue categories, you can begin considering which issues are most significant to your professional context. This grid also should allow you to develop better your "Contextual Analysis Plan."

Step 5: selecting issues for analysis in relation to your context

After completing your data coding grid, you need to select two or three issues that you will analyze in more detail. To determine which issues are most significant for your report, you should review the examples in your data grid noting which columns contain the details most relevant to your professional context. Use the following questions to guide your selection process.

- Can you locate details in your notes to illustrate your issues?
- Can you support their relevance to your professional context?
- Can you make connections among your issues?

3.3 Why do data coding?

- It lets you make sense of and analyze your data.
- For qualitative studies, it can help you generate a general theory.
- The type of statistical analysis you can use depends on the type of data you collect, how you collect it, *and* how it‘s coded.
- Coding facilitates the organization, retrieval, and interpretation of data and leads to conclusions on the basis of that interpretation.

3.4 When to code

- When testing a hypothesis (deductive), categories and codes can be developed b efore data is collected.
- When generating a theory (inductive), categories and codes are generated after examining the collected data.

3.5 Steps of coding (for qualitative data)

- Open - Break down, compare, and categorize data

- Axial - Make connections (relationships) between categories after open coding
- Selective - Select the core category, relate it to other categories and confirm and explain those relationships

3.6 Uses of Data Screening

- It is used to identify miscoded, missing, or messy data
- May be used find possible outliers, non-normal distributions and other anomalies in the data
- It can improve performance of statistical methods
- To make data conform to particular analysis methods.

3.7 When To Determine Codes

- For surveys or questionnaires, codes are finalized as the questionnaire is comp- leted
- For interviews, focus groups, observations, etc., codes are developed inductively after data collection and during data analysis

3.7.1 Why Code Determination

- Exhaustive – this ensures that a unique code number has been created for each category for example, if religions are the category, also include agnostic and atheist.
- Mutually Exclusive – ensures that information being coded can only be assigned to one category.
- Residual other – allows for the participant to provide information that was not anticipated, i.e. ―Other‖ _____
- Missing Data – ensures that conditions such as ―refused,‖ ―not applicable,‖ ―missing,‖ ―don‘t know‖ are accommodated.
- Heaping – ensure that the condition where too much data falls into same category, example, college undergraduates in 18-21 range (variable becomes useless because it has no variance) is avoided.

3.7.2 Creating Code Book

This allows study to be repeated and validated. It:

- Makes methods transparent by recording analytical thinking used to devise codes.
- Allows comparison with other studies.

Creating Code Frame Prior to data Collection

Code frame is used when you know the number of variables and range of probable data in advance of data collection, e.g. when using a survey or questionnaire.

- Use more variables rather than fewer
- Do a pre-test of questions to help limit ―other‖ responses

Transcription features

- Appropriate for open-ended answers as in focus groups, observation, individual interviews, etc.
- Strengthens ―audit trail‖ since reviewers can see actual data
- Uses identifiers that make participant anonymous but still reveal information to researcher; example Staff/Employee/Student number

Three Parts To Transcript

- Background information, e.g. time, date, organizations involved and participants.
- Verbatim (i.e. word for word) transcription
- Observations made by researcher, e.g. diagram showing seating, intonation of speakers, description of system, etc.

3.8 Coding Mixed Methods:

We use code mixed due to limitations of only using one method, especially:

- Quantitative data lack of thick description, and
- Qualitative data lacks visual presentation of numbers

3.8.1 Advantages of Mixed Methods:

- Improves validity of findings
- More in-depth data
- Increases the capacity to cross-check one data set against another
- Provides detail of individual experiences behind the statistics
- Provides more focused questionnaire
- Additional in-depth interviews can be used to tease out problems and seek solutions

3.8.2 Disadvantages of Mixed Methods

- Inequality in **data** sets
- **Data** sets must be properly designed, collected, and analyzed
- Numerical **data** set treated less theoretically, mere proving of hypothesis‖
- Presenting both **data** sets can overwhelm the reader
- Synthesized findings might be ―numbed-down‖ to make results more readable

Key Point in **Coding** Mixed Methods **Data**

The issue to be most concerned about in mixed methods is ensuring that your qualitative **data** have not been poorly designed, badly collected, and shallowly analyzed.‖

3.9 Outliers in Data Analysis

What Is an Outlier?

- According to Miller (1981): '... An outlier is a single observation or single mean which does not conform with the rest of the **data** ‘
- Barnett & Lewis (1984): '**.....**An outlier in a set of data is an observation which

appears to be inconsistent with the remainder of that set of data '

How to Handle outliers

Outliers can influence the analysis of a set of **data** and should be properly handled. There are four basic ways in which outliers can be handled:

- The outlier can be *accommodated* into the **data** set through sophisticatedstatistical refinements
- An outlier can be *incorporated* by replacing it with another model
- The outlier can be used to *identify* another important feature of the population being analyzed, which can lead to new experimentation
- If other options are of no alternative, the outlier will be *rejected* and regarded as a ―contaminant‖ of the **data** set

 **4.0 Self-Assessment Exercise(s)**

Answer the following questions:

1. What is the purpose of adopting a mixed data coding methods?
2. What is an outlier and in what ways can it be accommodated?
3. Define data-coding and explain its uses
4. Why is data screening necessary?
5. Explain when code determination is done and the purpose

 **5.0 Conclusion**

We cannot but transcribe data into appropriate format for system analysis. The choice of transcription method adopted is mostly dependent on the system being investigated. Care therefore must be exercised in selecting coding level especially for qualitative data.

 **6.0 Summary**

In this unit data coding and Screening, we:

- defined data coding as a systematic way used to condense extensive data-sets into smaller analyzable units through the creation of categories and concepts derived from the data
- explained why we have to code data,
- listed the uses of data screening
- explained Code determination of codes, why and the need for creating code book
- discussed mixed coding method and listed its advantages and disadvantages
- discussed outliers and how it can be handle in data analysis

7.0 Further Readings

- Gamerman, D., & Lopes, H. F. (2006). *Markov chain Monte Carlo: Stochastic simulation for Bayesian inference*. Boca Raton: Chapman & Hall/CRC.
- Graham, C., & Talay, D. (2015). *Stochastic Simulation and Monte Carlo Methods Mathematical Foundations of Stochastic Simulation*. Berlin: Springer Berlin Heidelberg.
- Mai, J., Scherer, M., & Czado, C. (2017). *Simulating copulas: Stochastic models, sampling algorithms, and applications*. New Jersey: World Scientific Publishing.
- Nelson, B. (2015). *Foundations and methods of stochastic simulation: A first course*. Place of publication not identified: Springer.
- Kozachenko, Y., Pogorilyak, O., Rozora, I., & Tegza, A. (2016). *Simulation of stochastic processes with given accuracy and reliability*. London: ISTE Press.
- Cochard, G. (2019). *Introduction to stochastic processes and simulation*. London, UK: ISTE.
- Rao, C. R., & Shanbhag, D. N. (2007). *Stochastic processes: Modelling and simulation*. Amsterdam: Elsevier.
- Gallager, R. G. (2017). *Stochastic Processes: Theory for Applications*. Cambridge, United Kingdom: Cambridge University Press.
- Jones, P. W., & Smith, P. (2018). *Stochastic processes: An introduction*. Boca Raton (Fla.): Chapman & Hall/CRC.