National University of Computer and Emerging Sciences

0015

Lahore Campus

Simulation and Modeling (CS 4056)

Final

Name = Hasan Yahya

Date: December 20, 2025

Course Instructor(s)

Dr. Mirza Mubasher Baig

Total Time (Hrs):	3
Total Weight:	45 (65 Points)
Total Questions:	5

221-7471
Roll No

BSE-7A
Section

[illegible]
Student Signature

Instructions: Answer All Questions
In Case of any missing values state your assumptions clearly
Use of a two sided A4 size hand-written cheat-sheet is allowed

Question No 1: [Discrete Even System Simulation] [1 + 3 + 3 + 2 + 3 + 3 Marks]

i) Give an example of a system input that better be modeled as a random variable. Justify by giving a short reason.

ii) Prove that the maximum likelihood parameter estimate of an exponential random variable is

$$\boxed{\hat{\lambda} = \frac{1}{\bar{x}}}$$

where $\bar{x}$ is the sample mean.

iii) A seasoned expert decided to model a systems input (inter-arrival times) as an exponential random variable with parameter λ. Use the following sample to estimate the value of parameter using its ML estimate.

{ 0.8, 1.2, 0.5, 2.0, 1.5, 0.9, 1.1, 0.6, 1.1, 0.3}

{0.3, 0.5, 0.6, 0.8, 0.9, 1.1, 1.1, 1.2, 1.5, 2 }

iv) Name two methods, described in the book, for measuring goodness of distribution fit while modeling the inputs.

v) Use the input model of part iii to generate 4 samples from this distribution. If you need uniform random numbers then use the following starting from left to right and reusing if more numbers needed

0.18 0.42 0.82 0.52 0.73

vi) Use the method of linear congruence's to generate 4 uniform random numbers between 0 and 1 using $X_0 = 27$, $a = 17$, $c = 43$ and $m = 100$

Question No 2:[Dynamical Systems Mdeling] [3 + 2 + 4 + 4 + 2 Marks]

i) The Taylor Series of a function is a way to represent value of a function at an arbitrary point using its value and values of its derivative at one point. It is given as

$$f(x) = f(a) + f'(a)(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \frac{f'''(a)}{3!}(x-a)^3 + \cdots$$

Starting from the Taylor series expansion of ***y (x + h)***, derive Euler's method for solving the initial value problem of the form

$$\frac{dy}{dx} = f(x,y), \quad y(x_0) = y_0.$$

ii) Give example of an initial value problem that can be exactly solved using a single step Euler's method.

iii) Consider the following initial value problem

$$\frac{dy}{dx} = y, \quad y(0) = 1$$

Use two step Euler's method to find Y(1)

iv) Solve the following initial value problem using a single step of RK4 to find Y(1). Clearly computing each of the terms K_1, K_2, K_3, and K_4

$$\frac{dy}{dx} = y - x^2 + 1, \quad y(0) = 0.5, \quad 0 \le x \le 2.$$

v) Give an example to demonstrate how a second order system can be written as a system of two linear systems.

Question No 3: [Networks (HOPFIELD)] [2 + 3 + 5 Points]

i. What are the basic parts of a network model?

ii. Explain the difference between **directed vs. undirected** and **weighted vs. unweighted** networks.

iii. Consider a Hopfield network with **3 neurons** and two stored patterns:

Pattern 1:	[1 -1 1]
Pattern 2:	[-1 1 -1]

Using **Hebbian learning rule**, compute the **weight matrix**.

National University of Computer and Emerging Sciences

0015

Lahore Campus

Question No 4: [AGENTS and (BOIDS)] [3 + 2 + 3 + 2 Points]

i. Describe the core components of an agent based model

ii. Describe the three main characteristics of i) Separation, ii) Alignment and iii) Cohesion used in simulation of BOIDS during the course

iii. Following code has been taken from an implementation of Boids simulation but the function names have been lost while coping the code. Specify which function is used to compute i) Alignment, ii) Cohesion and iii) Separation respectively. Provide a short justification of your answer

```
def compute_A(B):
    F = Vector(0, 0)
    N = B.get_neighbors(radius=neighbor_radius)
    if N:
        for n in N:
            F += n.velocity          Alignmnt
        F /= len(N)
    return (F - B.velocity)
    return alignment_force

def compute_C(B):
    F = Vector(0, 0)
    for N in B.get_neighbors(radius=desired_distance):
        F += (B.position - N.position).normalize() / distance(B, N)
    return F                       cohesion

def compute_B(B):
    F = Vector(0, 0)
    N = B.get_neighbors(radius=neighbor_radius)
    if N:
        for n in N:
            F += n.position          Seperation
        F /= len(N)
    return (F - B.position)
```

iv. Consider the following modeling paradigms: Cellular Automata (CA), Network Models, and Agent-Based Models (ABM). Arrange these models in order of **increasing generality**, and justify your ordering.

National University of Computer and Emerging Sciences

Lahore Campus

0015

Question No 5: [Cellular Automata and Conway's Game of Life]

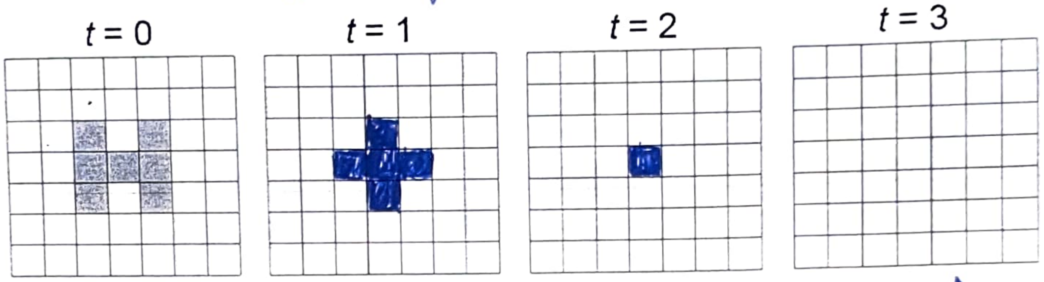
[2 + 1 + 3 + 2 + 2 + 1 + 1 + 1 + 2 Points]

i. What do we mean by the fact that a CA is Turing complete? Also describe rule 110 and its significance in the context of cellular automata.

ii. What is the difference between Moore and Neumann neighborhoods?

iii. Shown below is an example of a two-dimensional totalistic CA model with von Neumann neighborhoods and with periodic boundary. White means 0 (= quiescent state), while gray means 1. Each cell switches to round(S/5) in every time step, where S is the local sum of the states within its neighborhood. Complete the time evolution of this CA.

Assuming, we also count centre cell (0,0) as part of neighbourhood, & 0.5 & above are rounded to 1.0 vs 0.4 & below are rounded to 0.0.



All zeros

iv. State the rules of Conway's Game of Life

v. Considering the Game of Life rules, describe a pattern of 1's that
a. Do not change when these rules are applied on it.
b. That will keep on changing periodically

vi. Describe the state of cells in a 2D-grid at time t = 1 if we start with all cells set to 1 at time t = 0 and use Conway's Game of Life rules with infinite boundary while simulating the CA

vii. Calculate the number of possible state-transition functions for a two-dimensional CA model with two states and Moore neighborhoods (i.e., r = 1)

viii. Calculate the number of all possible configurations of a two dimensional, two-state CA model with L = 100.

ix. For a CA with 3 neighbors per cell and probability p=0.6 that a neighbor is 1, compute the expected fraction of cells becoming 1 in the next time step if a cell turns 1 if exactly 2 neighbors are 1.

0.24