001

DISCRETE EVENT SIMULATION

VINNA RAHMAYANTI SETYANING NASTITI, S.SI., M.SI

SOCIAL SCIENCE CLASS | LAMFORD SCHOOL

WHAT IS A DISCRETE EVENT SIMULATION?

- **Discrete Event Simulation (DES)** is a method for modeling the operation of a system **as a sequence of events** occurring at **discrete points in time**.
- Unlike **continuous simulation** (which models changes over time), DES **jumps from one event to the next**, updating the system state only when an event occurs.

Characteristics of DES:

✅ **Discrete events:** The system changes state only at **specific event times**.
✅ **Queue-based:** Many DES models involve queues (e.g., customers waiting for service).
✅ **Time-driven:** The simulation clock advances **event by event**, not continuously.

EXAMPLES OF DES APPLICATION

- **Queueing systems** (e.g., bank tellers, customer service centers)
- **Manufacturing processes** (e.g., assembly lines)
- **Computer networks** (e.g., data packet transmission)
- **Healthcare systems** (e.g., patient flow in hospitals)
- **Supply chain logistics** (e.g., warehouse management)

KEY COMPONENTS OF A DES MODEL

COMPONENT	DESCRIPTION
Entities	The objects being modeled (e.g., customers, machines, vehicles).
Events	Actions that change the system state (e.g., customer arrival, service completion).
Queue	A waiting line for resources (e.g., people in a bank queue, packets in a network).
Resources	Limited facilities (e.g., servers, machines, agents) that entities interact with.
Simulation Clock	Keeps track of the current simulation time.
Scheduler	Manages event execution in chronological order.

HOW DOES DISCRETE EVENT SIMULATION WORK?

Step-by-Step Process:

1. **Initialize the system state** (e.g., empty queue, available servers).
2. **Schedule the first event** (e.g., first customer arrival).
3. **Process the next event** (update system state, generate new events).
4. **Advance the simulation clock** to the next event time.
5. **Repeat until termination condition is met** (e.g., after 100 customers).

DISCRETE vs CONTINUOUS SIMULATION

Feature	Discrete Event Simulation (DES)	Continuous Simulation
Time Progression	Moves from one event to the next	Evolves continuously
State Changes	Happens only at discrete event times	Happens at every time step
Example	Customer queue at a bank	Temperature change in a reactor

- **DES IS PREFERRED** when changes happen at distinct times (e.g., queues, transaction)
- **Continuous simulation** is used for **physical systems** (e.g., fluid flow, weather)

EXAMPLE : Bank Queue Simulation Using SimPy (Python)

Scenario:

- Customers arrive at a bank at **random intervals**.
- Each customer waits in a queue until a teller is available.
- Service time varies **randomly** based on a probability distribution.

PYTHON CODE : BANK QUEUE SIMULATION

https://colab.research.google.com/drive/1zzEX8jbNPXmiwL6pZgx2o5DlNjstMau7

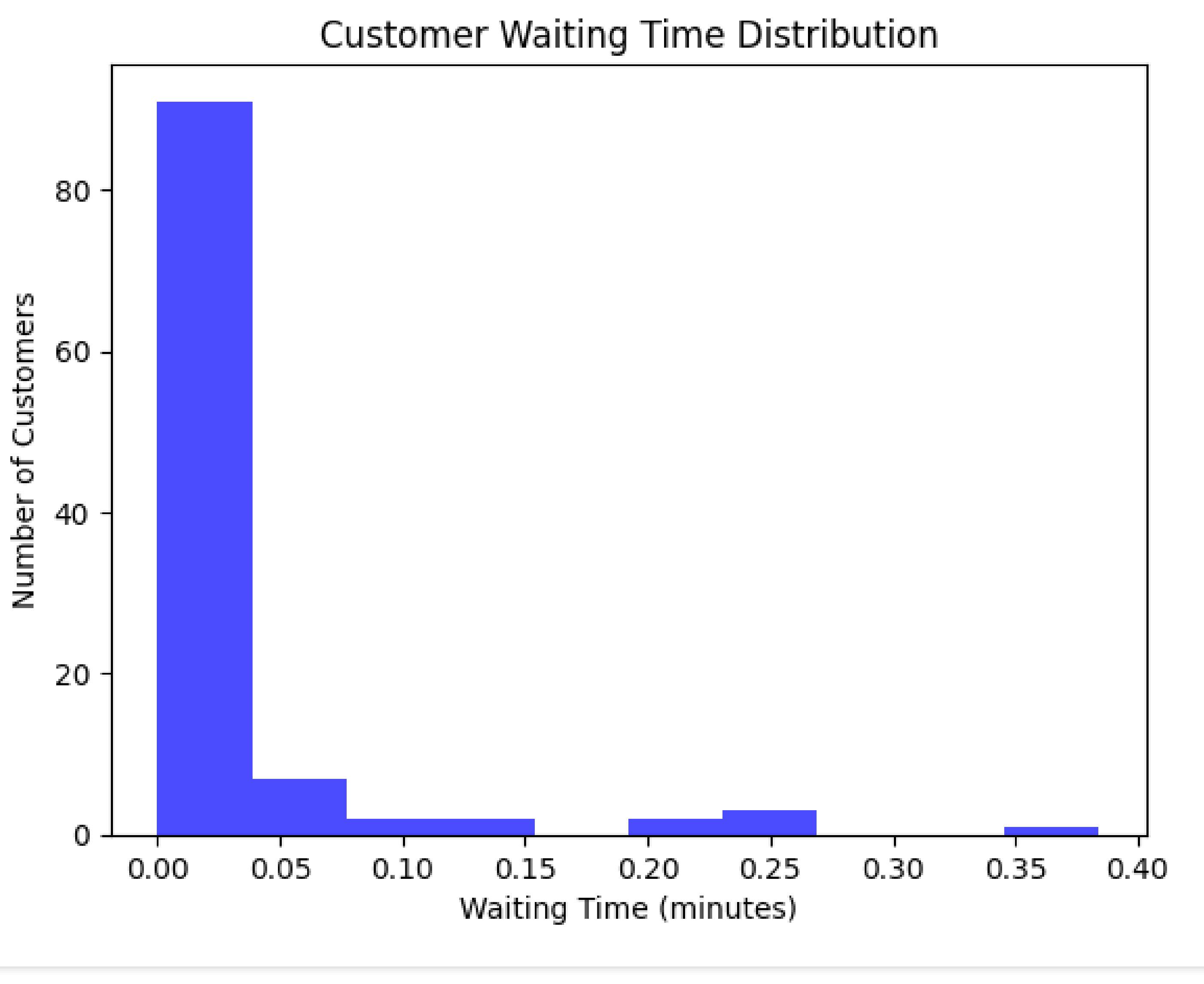
EXPLANATION OF SIMULATION OUTPUT

1. **Customers arrive randomly** at different times.
2. **If tellers are free, customers are served immediately**.
3. **If tellers are busy, customers wait in the queue**.
4. **Waiting times are recorded** for analysis.
5. **Histogram visualizes waiting times** across customers.

SAMPLE CONSOLE OUTPUT

```
Requirement already satisfied: simpy in /usr/local/lib/p
Customer 1 arrives at 0.20 min.
Customer 1 starts service after waiting 0.00 min.
Customer 2 arrives at 0.21 min.
Customer 2 starts service after waiting 0.00 min.
Customer 1 finished service at 0.25 min.
Customer 3 arrives at 0.26 min.
Customer 3 starts service after waiting 0.00 min.
Customer 2 finished service at 0.40 min.
Customer 4 arrives at 0.49 min.
Customer 4 starts service after waiting 0.00 min.
Customer 5 arrives at 0.50 min.
Customer 6 arrives at 0.51 min.
Customer 7 arrives at 0.56 min.
Customer 4 finished service at 0.56 min.
Customer 5 starts service after waiting 0.06 min.
Customer 5 finished service at 0.57 min.
Customer 6 starts service after waiting 0.06 min.
Customer 3 finished service at 0.58 min.
Customer 7 starts service after waiting 0.02 min.
Customer 6 finished service at 0.60 min.
Customer 8 arrives at 0.70 min.
Customer 8 starts service after waiting 0.00 min.
Customer 7 finished service at 0.73 min.
Customer 8 finished service at 0.74 min.
Customer 9 arrives at 0.86 min.
Customer 9 starts service after waiting 0.00 min.
Customer 10 arrives at 1.04 min.
Customer 10 starts service after waiting 0.00 min.
Customer 11 arrives at 1.04 min
```

EXPLANATION OF SIMULATION OUTPUT



🔷 **Expected Graph Output (Histogram of Waiting Times)**

- **X-axis:** Waiting times (in minutes).
- **Y-axis:** Number of customers.
- **Peak around 0-2 min:** Most customers experience **low wait times.**

✅ **The simulation provides insight into queue performance and optimization!**

ASSIGNMENT FOR STUDENTS

Objective:

- Modify the bank queue simulation and analyze system performance.

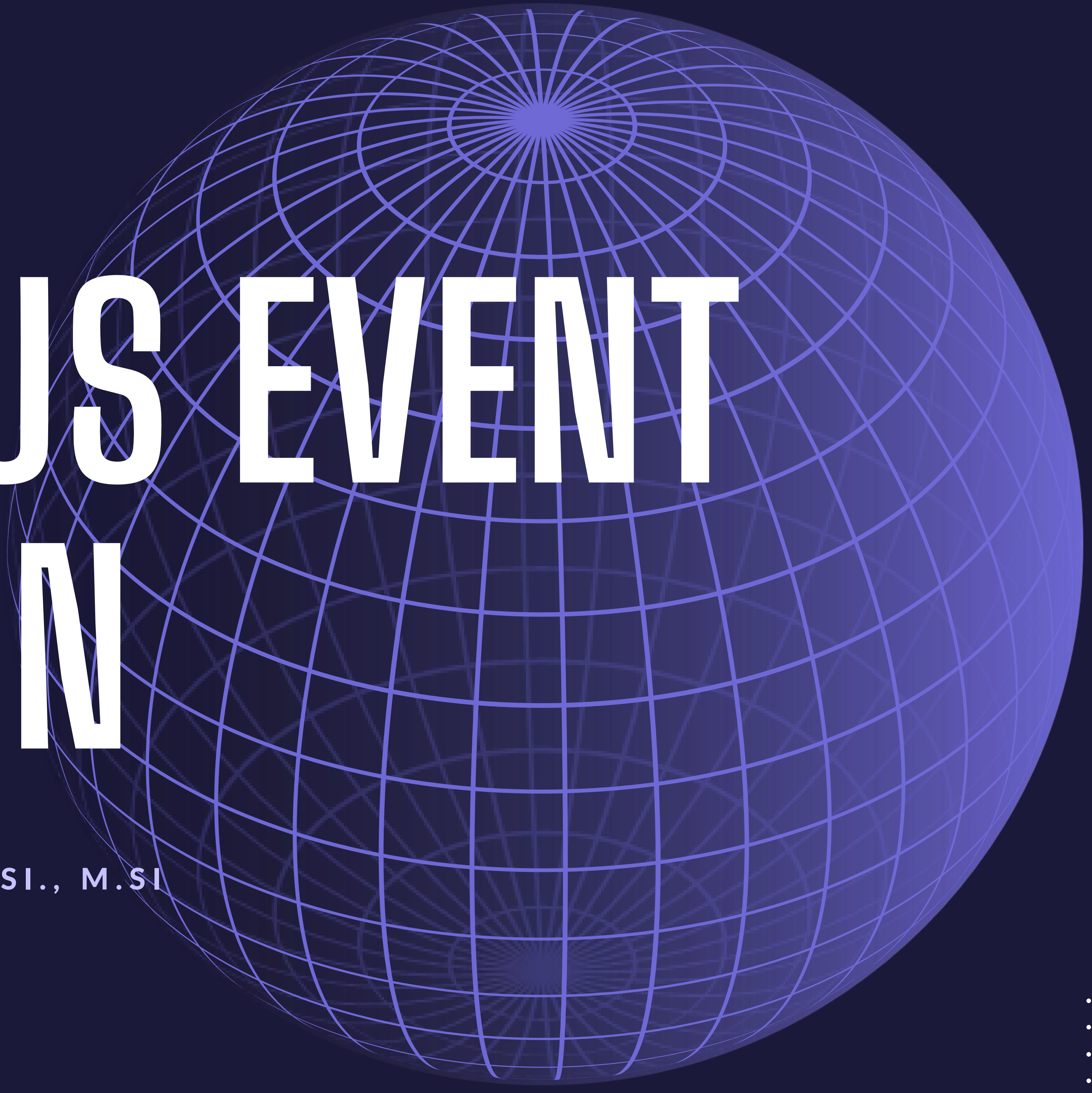
Tasks:

1. **Change the number of tellers** and analyze the effect on waiting times.
2. **Modify the arrival rate** to simulate peak hours.
3. **Visualize agent utilization** over time.
4. **Implement a priority queue** (VIP customers get served first).

SUMMARY

✅ DES models discrete events happening over time.
✅ It is used in real-world applications like banking, logistics, and healthcare.
✅ Python's SimPy makes it easy to implement DES simulations.

CONTINUOUS EVENT SIMULATION



VINNA RAHMAYANTI SETYANING NASTITI, S.SI., M.SI

SOCIAL SCIENCE CLASS | LAMFORD SCHOOL

WHAT IS A CONTINUOUS EVENT SIMULATION?

- **Continuous Event Simulation (CES)** models systems where changes occur **continuously over time**, rather than at **discrete points** like in **Discrete Event Simulation (DES)**.
- Used for **physical processes**, **biological systems**, and **engineering simulations** where state changes evolve smoothly over time.

Characteristics of CES:

✅ **Time is continuous**, rather than jumping from one event to another.
✅ **Changes happen at every time step**, instead of only at specific events.
✅ **Mathematical equations (differential equations)** often define system behavior.

EXAMPLES OF CES APPLICATION

📌 **Physics simulations** (e.g., fluid dynamics, heat transfer)
📌 **Epidemiology models** (e.g., COVID-19 spread)
📌 **Biological systems** (e.g., population growth, drug metabolism)
📌 **Control systems** (e.g., PID controllers in robotics)
📌 **Financial modeling** (e.g., stock market trends)

KEY COMPONENTS OF A CES MODEL

COMPONENT	DESCRIPTION
State Variables	Values that change continuously over time (e.g., temperature, velocity).
Time Step (Δt)	The small interval at which changes are calculated.
Differential Equations	Mathematical functions that describe system evolution.
Initial Conditions	The starting values of state variables.
Solver (Numerical Methods)	Algorithms used to compute changes over time (e.g., Euler's method, Runge-Kutta).

HOW DOES CONTINUOUS EVENT SIMULATION WORK?

Step-by-Step Process:

1. **Define the system state variables** (e.g., temperature, population size).
2. **Set up differential equations** to describe system behavior.
3. **Choose an appropriate solver** (e.g., Euler's method).
4. **Set an initial state** and specify simulation time.
5. **Iterate through small time steps**, updating system states continuously.
6. **Visualize results over time**.

EXAMPLE : Comparing DES vc CES

SIMULATION TYPE	EXAMPLE SCENARIO
Discrete Event Simulation (DES)	A **bank queue**, where customers arrive at random times.
Continuous Event Simulation (CES)	The **temperature change** of a metal plate over time.

◈ **CES models systems where time evolution is smooth and continuous.**

DIFFERENTIAL EQUATIONS IN CONTINUOUS SIMULATION

Example: Population Growth Model (Exponential Growth)

The rate of population change is proportional to the current population:

$$\frac{dP}{dt} = rP$$

Where:

- P = Population size
- r = Growth rate
- t = Time

🔷 **To simulate this, we use numerical methods like Euler's method.**

EXAMPLE : Simulation Population Growth Using Python

We simulate a **growing population** where the birth rate is proportional to the population size

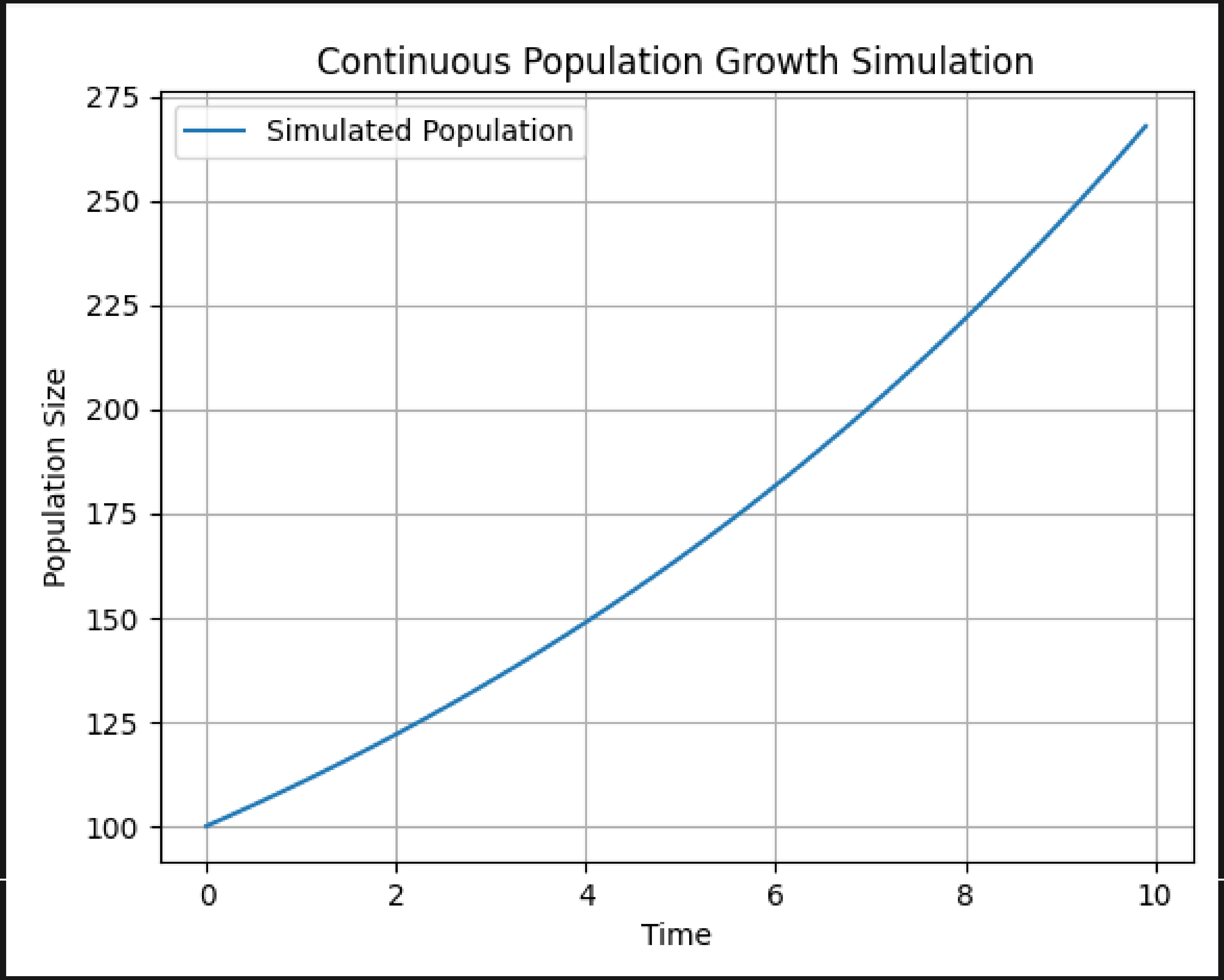
◈ **Explanation of Output**

1. **Initial Population** = 100.
2. **Each time step (dt = 0.1), population grows continuously.**
3. **Using Euler's method**, the system updates population size.
4. **Graph shows exponential growth**, proving that CES correctly models population dynamics.

◈ Sample Graph Output

✅ A smooth exponential curve showing continuous population growth over time.

✅ Confirms that population increases smoothly without discrete jumps.



INFORMATICS, UNIVERSITY MUHAMMADIYAH OF MALANG

EXAMPLE : Heat Transfer Simulation

We simulate **temperature change over time** as an object **cools down** according to **Newton's Law of Cooling**:

$$\frac{dT}{dt} = -k(T - T_{ambient})$$

Where:

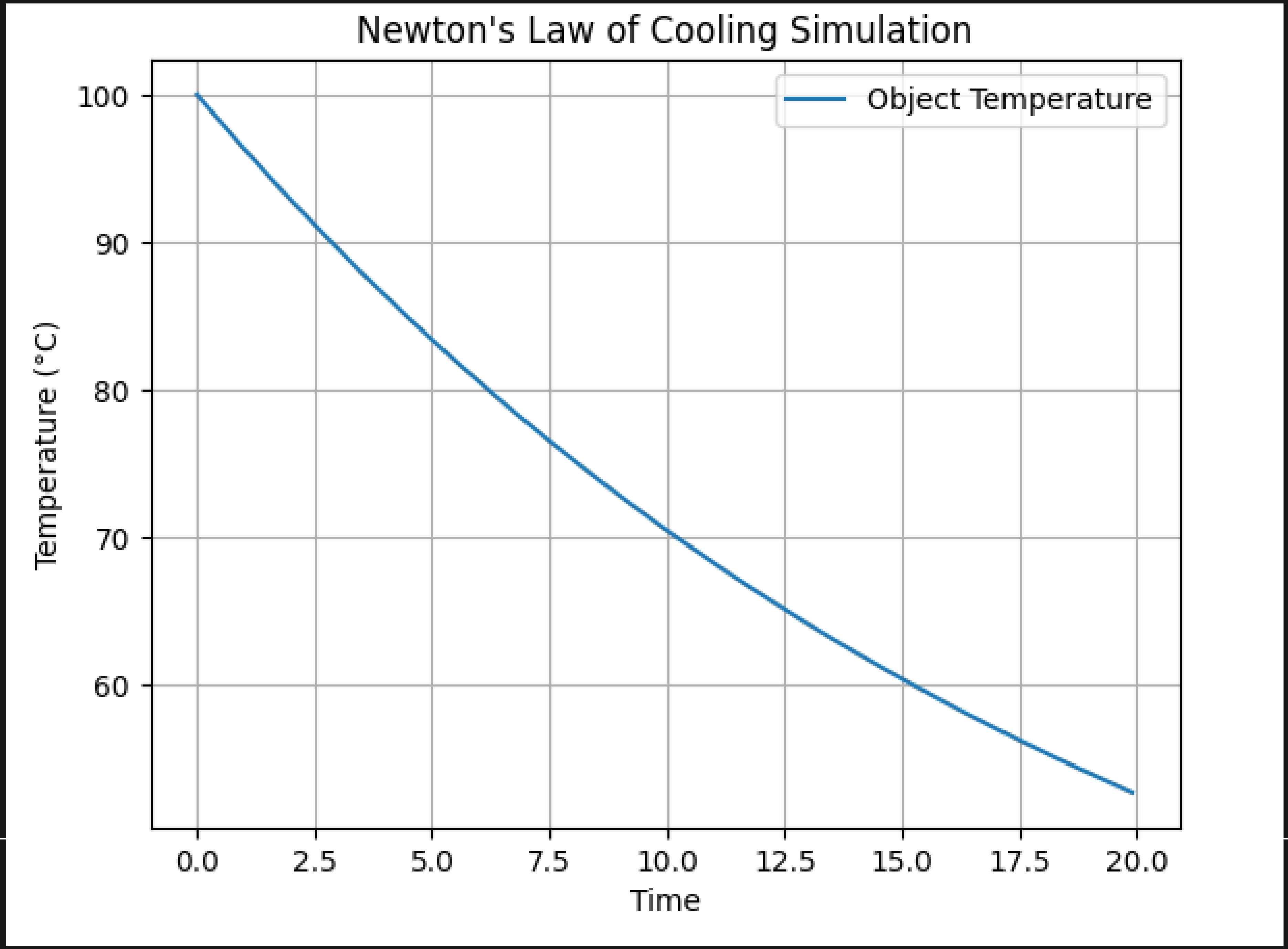
- T = Object temperature
- Tambient = Surrounding temperature
- k = Cooling rate

📝 Explanation of Output

1. **Initial object temperature = 100°C**, cooling towards **25°C** (ambient).
2. **Over time, temperature gradually decreases** based on cooling rate k.
3. **Graph shows an exponential decay curve**, consistent with Newton’s Law of Cooling.

Sample Graph Output

- Temperature gradually decreases towards room temperature.
- Confirms that CES correctly models heat transfer.



INFORMATICS, UNIVERSITY MUHAMMADIYAH OF MALANG

LINK PYTHON CODE

https://colab.research.google.com/drive/1zzEX8jbNPXmiwL6pZgx2o5DlNjstMau7

ASSIGNMENT FOR STUDENTS

Objective:

- Modify the **heat transfer simulation** and analyze different scenarios.

Tasks:

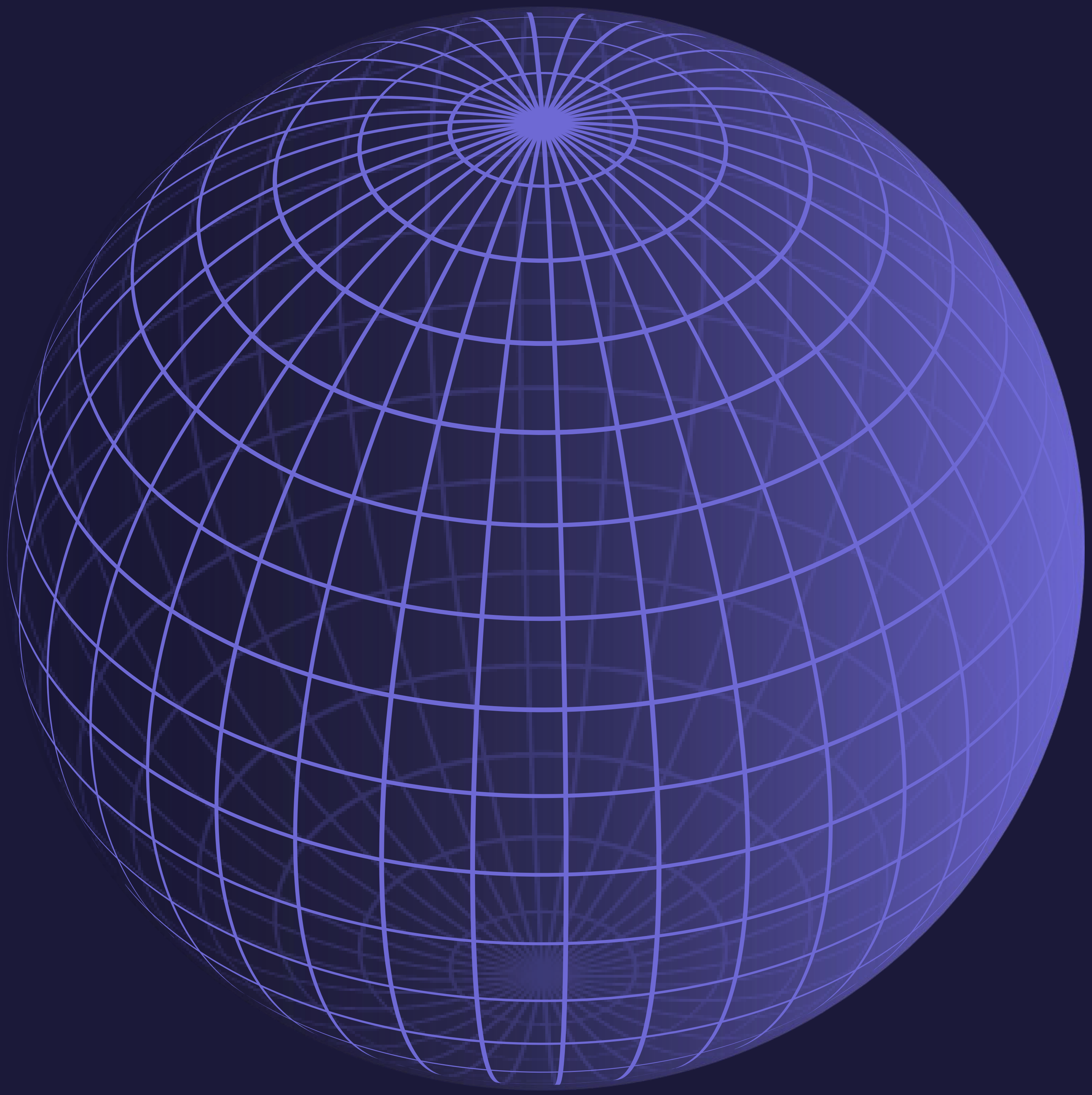
1. **Change the cooling rate** and observe the effect on temperature decay.
2. **Simulate heating** instead of cooling (modify differential equation).
3. **Use a real-world dataset** from **Kaggle (e.g., weather data)** to simulate **continuous temperature changes**.
4. **Compare CES vs. DES for similar systems.**

SUMMARY

Feature	Continuous Event Simulation (CES)
Time Handling	Continuous (every time step)
Changes	Smooth, gradual updates
Typical Applications	Physics, biology, finance
Mathematical Representation	Differential equations

✅ **CES is ideal for modeling smooth, real-world processes!**

✅ **Python's NumPy and Matplotlib make CES implementation easy!**



THANK YOU

Vinna Rahmayanti Setyaning Nastiti., S.Si., M.Si