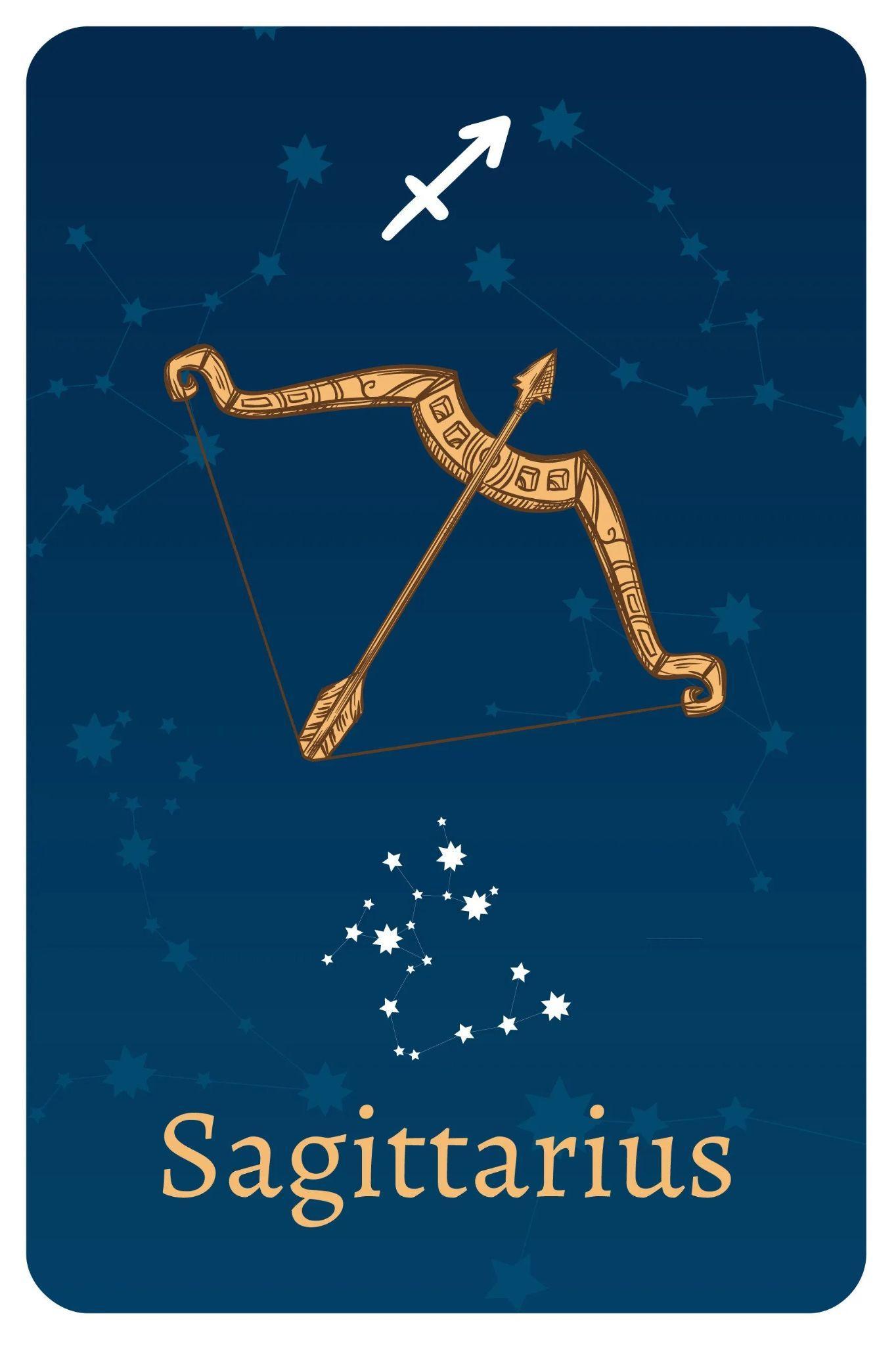
****

TEAM MEMBERS

Shivansh Shukla

Nathan Lee

Edwin Kulakkattolikel

Contents

[PURPOSES AND GOALS 3](#_heading=h.gjdgxs)

[HOW TO INSTALL THE PROGRAMS 4](#_heading=h.30j0zll)

[DEPENDENCIES 4](#_heading=h.1fob9te)

[HOW TO RUN THE PROGRAMS 4](#_heading=h.3znysh7)

[DESIGN ARCHITECTURE 5](#_heading=h.2et92p0)

[PROCESS & WORKFLOW 5](#_heading=h.tyjcwt)

[TEST DATA](#_heading=h.3dy6vkm) 7

[VIDEO RECORDINGS](#_heading=h.1t3h5sf) 10

[CONCLUSIONS](#_heading=h.4d34og8) 11

[REFERENCES](#_heading=h.2s8eyo1) 11

# PURPOSES AND GOALS

The goal of this laboratory assignment is to give students hands-on experience in developing mathematical models for real-world scenarios in the automotive industry, electrical engineering industry, and process capacity modeling. Furthermore, the lab intends to teach students how to develop and implement a simple game matching numbers in numerical order.

Goals:

1. Head Injury Criterion (HIC) Numerical Integration:
   1. Learn about the Head Injury Criterion (HIC) in the context of automotive safety.
   2. Create an airbag system that meets specified HIC specifications, such as matching Mercedes-Benz or Audi's HIC.
   3. Calculate HIC with and without the airbag using the provided equations.
   4. Prove your ability to use numerical integration techniques.
2. ER Modeling Using Differential Equations:
   1. Model a hospital's emergency room (ER) and intensive care unit (ICU).
   2. Identify and describe variables associated with emergency room and intensive care unit operations, such as medical staff, equipment, patient inflow, patient outcomes, and death rates.
   3. Create mathematical models for the ER and ICU using the ODEINT technique, taking into account at least three essential variables for each unit.
   4. Investigate various scenarios and generate graphs to visualize system behavior.
   5. Showcase scenarios in which both the ER and the ICU achieve saturation, as well as the variable values that cause these conditions.
   6. Apply knowledge gained from the lecture to create custom ER and ICU models.
3. Game - Follow the Numbers:
   1. Build a simple game that challenges players' hand-eye coordination.
   2. Implement multiple levels of increasing difficulty to keep players engaged.
   3. Track and display game statistics, such as time taken to complete each level.
   4. Provide an enjoyable and entertaining gaming experience for users.
   5. Learn from the development process and improve coding skills.

# HOW TO INSTALL THE PROGRAMS

## DEPENDENCIES

**HIC:** pip install numpy, scipy, and matplotlib

**ERU/ICU:** pip install numpy, scipy, and matplotlib

**Follow the Numbers:** pip install pygame, pgzrun, time, and random

# HOW TO RUN THE PROGRAMS

**HIC:** Run Code and see plots. Chosen code models the equations from the website provided that closely match the HIC score of the Mercedes Benz HIC = 310.

**ERU/ICU:** Alter Code to closely match current or possible inflow, outflow, and differential equation coefficient of current population for either departments, then run code in preferred IDE.

**Follow the Numbers:** To engage with the developed game, it is necessary to have Python and Pygame Zero installed on your system. After copying the provided code into a Python file, execute it using the 'pgzrun' command through a terminal or command prompt. The gameplay involves connecting dots by clicking on them in the prescribed order. The objective is to successfully complete all levels while adhering to the maximum allowed number of misses. Winning the game entails achieving this goal within the established parameters.

# DESIGN ARCHITECTURE

**HIC:**

Hardware includes the computer system, display, and input devices.

Software utilizes Python libraries (math, matplotlib, numpy, scipy.integrate) for calculations and visualization.

**ERU/ICU:**

Hardware components consist of the computer system, display, and input devices.

Software relies on Python libraries ( math, matplotlib, numpy, and scipy.integrate) for mathematical calculations and data visualization.

**Follow the Numbers:**

The code's architecture centers around a software-driven game developed using the Pygame Zero framework, which interacts with underlying hardware components like the display and input devices. The game logic manages various aspects, including level progression and user interactions, while the user interface renders the game screen and communicates with the player through messages. Dots and lines are represented as software components, and time measurement is facilitated through the software-based time module. Additionally, the code relies on essential software dependencies like Python and Pygame Zero. While the code executes locally on the user's computer, potential integration with cloud-based components for features such as online scoreboards or multiplayer functionality could further enhance the game's capabilities.

# PROCESS & WORKFLOW

**HIC:**

The Python code defines two functions, H1(t) (with airbag) and H2(t) (without airbag), to calculate HIC values based on time t.

Numerical integration from scipy.integrate is used to perform calculations.

Data points for plotting are generated using numpy.

Two plots, one for HIC with an airbag and another without, are created using matplotlib.

Plots are displayed on the computer screen, allowing users to visualize the HIC values.

**ERU/ICU:**Constants are defined for ERU and ICU variables, such as the number of doctors, nurses, beds, and respirators.

An inflow rate function qin(t) is established, which remains constant at 1.

The hospital function models the entire hospital, calculating outflow from ERU and ICU based on available resources and staff.

Differential equations dhdt1 and dhdt2 represent the patient population changes in ERU and ICU over time.

The code handles unit saturation, setting the rate of change to zero if capacity is exceeded.

Initial states for both units are set to zero patients.

A time vector t is defined for the simulation.

The odeint function is used to solve the differential equations and simulate patient flow.

Patient data for ERU and ICU are extracted.

Matplotlib is used to create a plot showing patient numbers in ERU and ICU over time, with appropriate labels and a legend for clarity.

**Follow the Numbers:**

The process of playing the game involves initiating the game through the Python script, which initializes the Pygame Zero game engine. Players start at Level 1, where random dots are displayed, and they interact by clicking on the dots in the specified sequence, connecting them with lines. Successful completion of a level leads to progression to the next level, each with its unique dot configurations. However, exceeding the maximum allowed misses (4) results in a game over. Conversely, successfully connecting all dots in all levels leads to a win, accompanied by a message displaying completion times for each level. The workflow entails starting the game, progressing through levels by successfully connecting dots, and either concluding the game with a win or encountering a game over scenario, after which players can exit the game.

# TEST DATA

| **Test Name** | **Test Output** |
| --- | --- |
| ERU/ICU Plots |  |
|  |  |
|  |  |
|  |  |
| HIC with Airbag |  |
| HIC no Airbag |  |
| Follow the Numbers: Level 1 |  |
| Follow the Numbers: Level 2 + 2 Misses |  |
| Follow the Numbers: Level 3 with 2 different sets of dots |  |
| Follow the Numbers: You Lost screen and Final screen after finishing the levels |  |

# VIDEO RECORDINGS

| **Recording Title** | **URL** | **Notes** |
| --- | --- | --- |
| EE104 Lab 4 ERU ICU | <https://youtu.be/TJykqn_SRVE> | Hospital Occupancy Modeling |
| EE104 Lab 4 HIC | <https://youtu.be/hLuESWNJaXI> | HIC Modeling |
| EE 104 - Lab 4: Follow the Numbers | <https://youtu.be/KTXn3qIgsG8> | Game Demo for Follow the Numbers |

# 

# CONCLUSIONS

In this lab, our team had three primary objectives spanning different tasks. First, in the context of automotive safety, we aimed to understand the Head Injury Criterion (HIC) and design an airbag system that met specific HIC specifications, akin to those used by Mercedes-Benz or Audi. We successfully achieved this goal by creating the airbag system and calculating HIC using numerical integration techniques. This task taught us the importance of precision in numerical integration and the practical application of HIC in automotive safety. To improve in the future, we can focus on refining our numerical integration methods and acquiring better resources for more accurate simulations.

Secondly, our task involved modeling a hospital's emergency room (ER) and intensive care unit (ICU) operations. This included identifying variables related to these units, such as medical staff, equipment, patient inflow, outcomes, and death rates, and creating mathematical models using the ODEINT technique. We successfully developed these models, explored various scenarios, and visualized system behavior. We learned about the complexity of healthcare system modeling and the significance of variables in emergency care. To enhance our models in future projects, we can incorporate additional variables and real-world data to improve accuracy, provided we have access to better resources and data.

Lastly, our team designed a hand-eye coordination game called "Follow the Numbers," with multiple difficulty levels and statistical tracking features. We successfully developed the game and achieved the objective of providing an enjoyable gaming experience while tracking player statistics. This project helped us gain insights into game development and coding skills and understand user engagement principles. To further enhance our future game projects, we can focus on improving the user experience, adding more features, and refining game mechanics, possibly with better development resources and user feedback to guide our improvements. In summary, we successfully met our lab objectives, and to do even better next time, we aim to refine our techniques and leverage improved resources for superior results in each area.

# REFERENCES

* What Is Agile Reporting? (Definition and How To Complete) <https://www.indeed.com/career-advice/career-development/agile-reporting>
* Test Report <https://strongqa.com/qa-portal/testing-docs-templates/test-report>