Project Report

Polytech Nice

Simple Road Traffic Modeling

14th December 2023

Gerbaud Florent
Fatima Rharrour

9-th october to 20-th December september 2023

Academic tutor : Didier Auroux

Table of Contents

List of Figures

List of Algorithms

1 Presentation of the Subject

1.1 Useful Definition

1. **Ordinary Differential Equation (ODE):**

   An ODE is a mathematical equation that relates a function to its derivatives with respect to one or more independent variables. ODEs are commonly represented given a function F of x, y, and derivatives of y. Then, an equation of the form

   $$F\left(x, y, y', \dots, y^{(n-1)}\right) = y^{(n)}$$

2. **SMRT**

   SMRT means Simple road traffic simulation.

3. **Microscopic simulation**

   Microscopic simulation is a computer-based modeling technique that simulates the behavior of individual entities, such as vehicles or pedestrians, within a system. It focuses on detailed modeling of each entity's movements, interactions, and behaviors to understand and predict the dynamics of a larger system, like traffic flow or crowd movement.

4. **Macroscopic simulation**

   Macroscopic simulation models systems at a higher, aggregated level, considering overall behaviors like traffic flow without detailing individual movements. It helps analyze system-level trends and capacities, useful for broader planning.

   The relation between microscopic and macroscopic simulation lies in their approach to studying systems. Microscopic simulation focuses on individual entities' detailed behaviors, while macroscopic simulation analyzes aggregated behaviors of the entire system without considering individual entities. They complement each other: microscopic for detailed insights into individual behaviors, and macroscopic for understanding overall system trends and capacities.

5. If you are interested in these types of modelizations, I suggest you go and see this article [**dugois:tel-01505473**].

1.2 Simple Road Traffic Modeling

Road traffic modeling involves studying how vehicles behave on road networks, with the aim of simulating and analyzing aspects such as traffic flow, congestion, and driver behavior. This field employs mathematical and computer models to understand and predict traffic patterns, playing a crucial role in urban planning, traffic management, and the development of intelligent transportation systems (as shown in Figure 1).

Explore the integration of a driving simulator with a traffic simulator in Jeihani et al.'s study [**JEIHANI2017164**], revealing enhanced traffic density and Variable Message Sign (VMS) reliability. The findings suggest that integration positively influences compliance behavior and factors affecting route diversion.

[**SMRT**]

Figure 1: **Road Traffic:** An illustrative example of a road traffic phenomenon, which is the focus of our study.

1.2.1 The Objective of SMRT

The primary goal of simple road simulation is to analyze and understand vehicle behavior in a controlled and reproducible environment. Researchers and engineers use these simulations to study the impact of various factors on traffic flow, safety, and efficiency. This contributes to the development and testing of traffic management strategies and vehicle control systems, providing insights into the dynamic interactions between vehicles and the road environment.

1.2.2 Use Case to Explain the Interest of These Simulations

Simple road simulations find applications in various fields, including urban planning, transportation engineering, and autonomous vehicle development. For instance, they can assess the effectiveness of new traffic signal timings, study the implications of road design changes, or test the performance of autonomous vehicles in different traffic scenarios. These simulations offer a cost-effective and risk-free way to evaluate real-world interventions and innovations, enhancing decision-making processes and supporting the development of sustainable and efficient transportation systems.

1.2.3 Field of Application

Simple road simulation is applied across a wide range of fields, including transportation research, traffic engineering, and urban planning. It plays a crucial role in the development and validation of autonomous vehicle technologies. The ability to model and simulate various road conditions helps researchers and practitioners make informed decisions, improving the design and management of transportation systems. Simple road simulation serves as a powerful tool for enhancing our understanding of traffic dynamics and contributes to the ongoing evolution of smart and adaptive transportation systems.

1.2.4 Using GitHub for Project Management

In this project, we selected GitHub as the primary management tool to facilitate our collaboration, track changes, and manage the source code. The following representation provides an overview of our collaborative use of GitHub as a pair throughout the project.

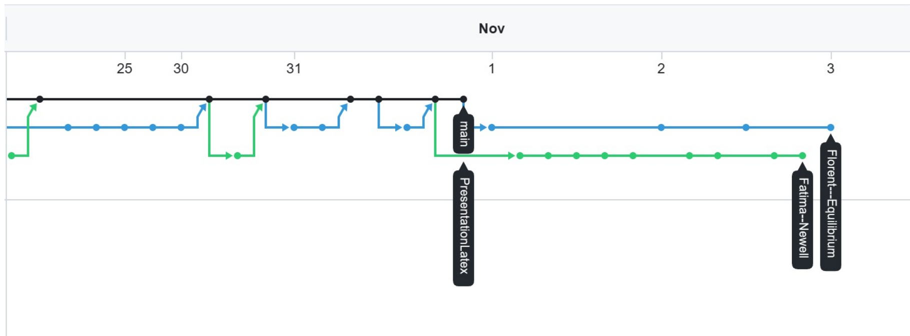


Figure 2: **Road traffic :** In this picture, you can see an example of a road traffic phenomenon that we could study

GitHub's branch feature enabled the two of us to work on different aspects of the project simultaneously without impacting the main codebase. It facilitated the creation of separate environments tailored to specific tasks, allowing for parallel development while maintaining a distinct separation between various project components. This approach significantly improved our collaborative workflow, reducing conflicts and boosting overall productivity.

Furthermore, the issues feature on GitHub played a pivotal role in task management and issue tracking. We utilized this functionality to efficiently create, assign, and monitor tasks. Each issue provided a dedicated space for discussing specific problems, enhancements, or new features, fostering a structured approach to problem-solving and enabling seamless communication between the two of us. This not only helped us maintain a clear overview of pending tasks but also cultivated a collaborative environment where discussions and resolutions were centralized.

2 Project Objectives

This project could be decomposed into different parts. The primary objectives of the project were to conduct microscopic and macroscopic simulations.

2.1 Microscopic Simulation

In the case of microscopic simulation, the objective was to simulate traffic flow phenomena using two types of models. The first model was a linear approach that considered only the position of the car in front of it. This model, known as the 'Follow-the-Leader' model, simplistically replicates vehicle behavior by focusing solely on its proximity to the preceding car. It assumes that each vehicle adjusts its speed to maintain a safe following distance.

The second model employed was Newell's method, which is more intricate and offers greater realism. Newell's model considers not only the preceding vehicle's position but also incorporates the speed and acceleration of adjacent vehicles. This method captures the interactions between neighboring cars, resulting in a more nuanced representation of traffic dynamics.

In this modeling, we will focus on the position and velocity of multiple vehicles following each other in a single lane. We will assume that vehicles accelerate when they have open space, maintain their speed when they are close to the vehicle in front, and brake more strongly as they get very close. We will arrange two and then three vehicles, one behind the other, with the first vehicle maintaining a constant velocity, and observe the behavior of the following vehicles. Once we have completed these simulations, we plan to introduce more complex and realistic conditions and behaviors, such as varying acceleration and encountering obstacles on the road.

2.2 Macroscopic Simuation

In the second part of the project, we encountered the limitations of the Microscopic Model. We will delve deeper into these limitations shortly. In essence, it lacks realism and provides fewer details compared to the macroscopic simulation.

To address this issue, we chose to employ Partial Differential Equations (PDEs). In this type of simulation, we do not solely focus on individual car behaviors but rather treat traffic as a fluid-like entity. This approach allows us to gain a better understanding of traffic flow simulation. Considering traffic as a fluid is justified by its collective behavior resembling fluid dynamics—vehicles move in a continuous flow, similar to how fluids move in pipes or channels. Thus, employing PDEs assists in capturing the overall flow dynamics.

Consequently, we utilize a simplified form of the Navier-Stokes equation to tackle the problem.

This enables us to refine our description of the previous simulations by providing a more comprehensive understanding of traffic dynamics within a fluid-like framework.

In clear, the main objectives of the project are:

- Modelling the behaviour using coupled ordinary differential equations.
- Implementing this simplistic modelling.
- Modelling the behaviour of traffic flow with PDE
- Studying the outcomes based on different parameters.

2.3 Problems Encountered

2.3.1 Implementation of the Macroscopic model

During the phase of developing the Macroscopic models, the difficulties lie in the fact that it was not easy to determine if the graphs we obtained were as expected. To verify their accuracy, we sought feedback from Mr. Auroux and also consulted various websites to compare our results with those of others. Initially, we believed the graphs were correct. However, we later discovered errors in our implementation, which we subsequently rectified.

One of the main problems we encountered was related to the boundary conditions. Initially, for each time step, we used constant boundary conditions. Consequently, the resulting graphs exhibited anomalies at the edges. To address this issue, we conducted research and sought advice from teachers to refine our approach.

Additionally, understanding the models posed a challenge. While the model worked for some cases, it failed for others. This was attributed to the violation of the CFL condition. When the density was too high, the model became unstable. Through an empirical approach, we determined the threshold value and conducted appropriate tests to implement the second model correctly.

2.3.2 The time

One of the significant challenges we faced revolved around managing our time effectively. Balancing the demands of our project alongside preparing for exams posed a considerable obstacle. Juggling these concurrent commitments was undeniably demanding. Nonetheless, it presented an invaluable challenge that encouraged us to refine our time management skills and prioritize tasks efficiently. The primary difficulty lay in our inability to delve deeply enough into the subject matter. Due to the time constraints imposed by various commitments, we found it challenging to explore the topic to its fullest extent. Despite this limitation, navigating these time-related obstacles taught us valuable lessons about effective multitasking and the importance of allocating time judiciously to various responsibilities.

2.3.3 The perfoming of Equlibrium

Achieving equilibrium was relatively straightforward for the Linear Model. However, it proved to be somewhat arduous for Newell's method. Initially, our approach involved attempting to find Analytical Solutions. However, after spending several hours on this calculus and conducting research online, we came to realize that an analytical solution may not exist, and if it does, it could be exceedingly challenging to obtain. This led us to recall the ODE course from previous years, specifically the Lyapunov indirect theorem, which provides criteria to ascertain whether solutions converge or diverge.

3 The equation for SMRT

3.1 Differential Ordinary Equation

In this part, the idea is to resolve two types of systems of Ordinary Differential Equations (ODEs) that allow us to simulate traffic flow. To achieve this, we will use the Euler Explicit method to numerically solve the solutions. The Euler Explicit method is given by the folowing eqution :

- EDO to solve: $\boxed{y'(t) = f\big(t, y(t)\big)}$.
- First step of the resolution: $\boxed{y_0 = y(t_0)}$.
- Recursive process to find the n-th solution of the EDO: $\boxed{y_{n+1} = y_n + hf(t_n, y_n)}$

3.1.1 Linear Model

Mathematical Theory:

In this section, we're going to explain the math behind our model for understanding how cars behave in traffic. To make things simple, we use a discrete model, which means we look at cars one at a time and how they interact on the road.

Each car's movement is governed by a basic equation:

$$\dot{x_i}(t) = V_i = \alpha_i(x_{i-1} - x_i)$$

In this equation, $x_i(t)$ represents where the i-th car is at a given time, V_i is how fast the i-th car is going, and α_i is a number that describes how that car behaves. The right side of the equation, $\alpha_i(x_{i-1} - x_i)$, tells us how the car's speed changes based on how close it is to the car in front.

When we put this equation to work for all the cars, we end up with a bunch of equations (one for each car), which helps us understand how they all move together in traffic. These equations give us a dynamic view of how cars influence each other as they drive.

The system of equations is written like this for each car, where i can be 1, 2, and so on, up to the number of cars:

$$\begin{cases} \dot{x}_1 & = V_1 \\ \dot{x}_2(t) & = \alpha_2(x_1 - x_2) \\ & \vdots \\ \dot{x}_n(t) & = \alpha_n(x_{n-1} - x_n) \end{cases}$$

Finally the Sytem associated to the position of each car is :

$$\begin{cases} x_1(t + \Delta t) & = x_1(t) + \Delta t \cdot V_1 \\ x_2(t + \Delta t) & = x_2(t) + \Delta t \cdot \alpha_2(x_1 - x_2) \\ & \vdots \\ x_n(t + \Delta t) & = x_n(t) + \Delta t \cdot \alpha_n(x_{n-1} - x_n) \end{cases}$$

These equations help us understand how traffic flows and how individual cars influence one another on the road.

In the first part of the study, we consider α_i as a constant. However, in the next part of the simulation, we define some functions $\alpha_i(t)$.

In fact, we have three types of simulations:

1. **Constant value:**

$$\alpha_i(t) = C_i$$

2. **Sinusoidal with noise:**

$$\alpha_i(t) = |W \cdot \sin(\omega t + \phi) + \mathcal{N}(0, 0.1)|$$

3. **Stochastic Driver Model:**

Consider a random driver model with an output labeled as $\alpha_i(t)$, which relies on various factors like the average, spread, and time t. This model produces a random noise part from a usual distribution with an average and spread of $\sqrt{\text{spread}}$. If a limit is given, the generated noise is confined within the interval $[-\text{limit}, \text{limit}]$.

Mathematically, we can express this as:

$$\alpha_i(t) = \begin{cases} \left|\mathcal{N}(\text{average}, \sqrt{\text{spread}})\right|, & \text{if } -\text{limit} \leq \alpha_i(t) \leq \text{limit}, \\ -\text{limit}, & \text{if } \alpha_i(t) < -\text{limit}, \\ \text{limit}, & \text{if } \alpha_i(t) > \text{limit}. \end{cases}$$

Here, average signifies the mean of the acceleration, and spread stands for the acceleration's spread. The limit is defined to prevent or create specific situations, like accidents, depending on the desired result.

Implementation :

Algorithm 1 FirstCarSpeed

Output:
- $\dot{x_1} = V_1$

Algorithm:
- $\dot{x_1} = 130 \times \frac{1000}{3600}$
- **return** $\dot{x_1}$

This algorithm allow to return the speed for the first car

Algorithm 2 SpeedFOrCarNLinear

Input:
- t := "time step"
- x_i := Table of position for the i-th Car
- x_{i-1} := Table of position for the (i-1)-th Car

Output:
- $\dot{x_i} = V_i$

Algorithm:
- $\dot{x_i}(t) = V_i = \alpha_i(x_{i-1}[t] - x_i[t])$
- **return** $\dot{x_i}$

This algorithm allows to return the speed given by the Linear Model for each car of the system

Algorithm 3 sinusoidal_model

Input:

- W := "Amplitude of the perturbation"
- ω := "Angular frequency"
- t := "time"
- ϕ := "Phase (in radians)"

Output:

- This function returns a value of acceleration ($\alpha_i(t)$) following a sinusoidal model with random noise.

Algorithm:

- $\alpha_i(t) = |W \cdot \sin(\omega \cdot t + \phi) + N|$
- N:="random noise such as N $\sim \mathcal{N}(0, 0.01)$"
- **return** $\alpha_i(t)$

This algorithm allows for variations in acceleration that follow a sinusoidal model to simulate drunk drivers.

Algorithm 4 stochastic_driver_model

Input:

- mean := Mean value of the normal distribution
- variance := Variance of the normal distribution
- t := time
- threshold := Threshold value for clipping noise (optional)

Output:

- noise: The generated random noise value.

Algorithm:

- Generate a random number x using a normal distribution with μ and $\sqrt{\sigma^2}$ as parameters.
- noise $\leftarrow |x|$
- **if** threshold $\neq$ None **then**
-     **if** noise $< -$threshold **then**
-         noise $\leftarrow -$threshold
-     **end if**
-     **if** noise $>$ threshold **then**
-         noise $\leftarrow$ threshold
-     **end if**
- **end if**
- **return** noise

This algorithm allows for variations in acceleration that follow a Stochastic model to simulate unpredictable drivers.

Algorithm 5 UpdatePositionsAndVelocities

Input:
- t := time step
- x_1Pos := Table of position for the 1st Car
- x_2Pos := Table of position for the 2nd Car
- v_1 := Table of velocity for the 1st Car
- v_2 := Table of velocity for the 2nd Car
- $time$:= Table of time values
- h := time step size

Output:
- x_1Pos
- x_2Pos
- $accident$:= Flag indicating whether an accident occurred

Algorithm:
- **for** t in range(1, n_steps) **do**
-     $x_1Pos[t] = x_1Pos[t-1] + \dot{x_1}() \times h$
-     $x_2Pos[t] = x_2Pos[t-1] + \dot{x_2}(t-1) \times h$
-     $v_1[t] = \dot{x_1}()$
-     $v_2[t] = \dot{x_2}(t)$
-     $time[t] = time[t-1] + h$
-     **if** isAccident(t) **then**
-         $accident$ = True
-         **break**
-     **end if**
- **end for**

Return x_1Pos, x_2Pos

This algorithm enables the implementation of the Euler Explicit Method on the Linear Model.

3.1.2 Newell's Model

Mathematical Theory: In this section, we model the velocity using an exponential model, so the movement of each car is given by:

$$\dot{x_i}(t) = V_i(1 - e^{-\frac{\lambda_i}{V_i}(x_{i-1}(t) - x_i(t) - d_i})$$

V_i, λ_i, d_i parameters associated with the ith car:

- V_i : the maximum velocity of the ith car.
- λ_i : the capacity of acceleration/decelaration.
- d_i : the minimum headway(safe following distance).

By using this equation for each car, we get a bunch of equations—one for each car. This helps us grasp how they all move together in traffic. These equations give us a dynamic picture of how cars affect each other as they drive.

The set of equations is written like this for each car, where i can be 1, 2, and so on, up to the total number of cars:

$$\begin{cases} \dot{x}_1 & = V_1 \\ \dot{x}_2(t) & = V_2(1 - e^{-\frac{\lambda_2}{V_2}(x_1(t) - x_2(t) - d_2}) \\ & \vdots \\ \dot{x}_n(t) & = V_n(1 - e^{-\frac{\lambda_n}{V_n}(x_{n-1}(t) - x_n(t) - d_n}) \end{cases}$$

And so, the position of each vehicle is:

$$\begin{cases} x_1(t + \Delta t) & = x_1(t) + \Delta t \cdot V_1 \\ x_2(t + \Delta t) & = x_2(t) + \Delta t \cdot V_2(1 - e^{-\frac{\lambda_2}{V_2}(x_1(t) - x_2(t) - d_2}) \\ & \vdots \\ x_n(t + \Delta t) & = x_n(t) + \Delta t \cdot V_n(1 - e^{-\frac{\lambda_n}{V_n}(x_{n-1}(t) - x_n(t) - d_n}) \end{cases}$$

Implementation:

for $\dot{x_1}$, we use the same algorithm than for the Linear model., However, to calculate the $\dot{x_i}$ values we use the following algorithm.

Algorithm 6 SpeedFOrCarNNewell

Input:

- t := "time step"
- x_i := Table of position for the (i)th Car
- x_{i-1} := Table of position for the (i-1)th Car
- d_i := Safe following distance the (i)th Car

Output:

- $\dot{x_i} := V_i$

Algorithm:

- $\dot{x_i}(t) := V_i(1 - e^{-\frac{\lambda_i}{V_i}(x_{i-1}(t) - x_i(t) - d_i})$
- **return** $\dot{x_i}$

This algorithm determines the derivative of x_i with respect to time, $\dot{x_i}$, based on the positions of the ith and $(i-1)$th cars, the safe following distance d_i, and the time step t. The result represents the velocity V_i of the ith car.

For other implementations such as the Euler Explicit method or variations of acceleration, the algorithm remains exactly the same as used in the previous section.

3.2 Partial Differential Equations

3.2.1 Euler Explicit

Mathematical Theory:

In the second part, the objective was to simulate the phenomenon of car traffic as a fluid. It means that instead of studying each car independently of the others, we focus on analyzing the density of the cars. To conduct this study, we need to employ a Partial Differential Equation (PDE) model.

The PDE model used is the LWR (Lighthill-Whitham-Richards) equation system. This system allows us to simulate car traffic using more or less complex equations For further insights on this topic, explore how uncertainty in the LWR traffic model's fundamental diagram affects prediction reliability, showcasing increased location uncertainty over time while accurately predicting disturbance magnitude. [**article**]. The equation is defined exactly as follows:

$$\boxed{\begin{array}{ll} \partial_t \rho + \partial_x F(\rho) = 0, & x \in \Omega, t \geq 0, \\ \rho(x,0) = \rho_0(x), & x \in \Omega, \\ \rho(0,t) = \rho(L,t), & t \geq 0 \end{array}}$$

Where:

$\Omega :=]0, L[$,
$\rho(x,t)$ represents the traffic density at position x and time t,
$F(\rho)$ denotes the traffic flux as a function of density,
$F(\rho)$ is often represented by a function modeling the relationship between traffic density and traffic velocity,
$F(\rho) = V(\rho) \cdot \rho,$ where $V(\rho)$ is the traffic velocity as a function of density.

For the study, we choose two types of equations for LWR:

$$\partial_t \rho + \partial_x \left[\rho \left(1 - \frac{\rho}{\rho_{\max}} \right) \cdot V_{\max} \right] = 0 \tag{1}$$

Also known as the Free Flow model, it's a model that allows the simulation of systems with lower density.

The first step to doing some simulation was to define a numerical model to calculate the appoximation of the solution. We choose the mzthod the most usual. Euler excplit method. So for the Free Flow model, the shem is given by :

$$\boxed{\rho_i^{n+1} = \rho_i^n - \frac{\Delta t}{\Delta x} \cdot \left(\rho_i^n \cdot v_i^n - \rho_{i-1}^n \cdot v_{i-1}^n \right) = 0}$$

<u>**Where :**</u>

$$\boxed{v_i^n = \left(1 - \frac{\rho_i^n}{\rho_{max}} \right) \times V_{max}}$$

The approach for calculating density involves creating a matrix where each row represents a time step and each column represents a spatial point. Initially, this matrix is populated with zeros, except for the first row, which holds the initial conditions.

This matrix structure appears as follows:

$$
\begin{array}{c c}
 & \begin{array}{cccc} \Delta_x & 2\cdot\Delta_x & \dots & L \end{array} \\
\begin{array}{c} \Delta_t \\ \\ 2\cdot\Delta_t \\ \vdots \\ T \end{array} &
\begin{pmatrix}
\rho_1^1 & \rho_2^1 & \dots & \rho_L^1 \\
 & & & \\
\rho_1^2 & \rho_2^2 & \dots & \rho_L^2 \\
\vdots & \vdots & \dots & \vdots \\
\rho_1^T & \rho_2^T & \dots & \rho_L^T
\end{pmatrix}
\end{array}
$$

Initially, the matrix is filled with zeros except for the first row, which contains the initial density conditions:

$$
\begin{array}{c c}
 & \begin{array}{cccc} \Delta_x & 2\cdot\Delta_x & \dots & L \end{array} \\
\begin{array}{c} \Delta_t \\ 2\cdot\Delta_t \\ 0 \\ T \end{array} &
\begin{pmatrix}
u_{0_{(\Delta_x)}} & u_{0_{(2\cdot\Delta_x)}} & \dots & u_{0_{(L)}} \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0
\end{pmatrix}
\end{array}
$$

Subsequent values in the matrix are calculated based on the previous values using the formula:

$$\rho_i^{n+1} = \rho_i^n - \frac{\Delta t}{\Delta x} \cdot \left(\rho_i^n \cdot v_i^n - \rho_{i-1}^n \cdot v_{i-1}^n\right) = 0$$

This calculation employs all values from the spatial step at the preceding time step. The boundary conditions for time steps are specified only for time step 0. The boundary condition at the last time step is determined using the Euler Explicit method.

The boundary condition for $t = 0$ is redefined iteratively during the process as it considers the density performing by Euler Explicit from the previous time step for the first spatial time.

3.3 Lax-Friedrich Model

The problem with the previous scheme is that for a higher density, the solution becomes unstable and diverges towards infinity. To solve this problem, we decided to implement the Lax-Friedrich model. The theoretical model for the Lax-Friedrich method in the non-linear system is as follows:

$$\boxed{u_i^{n+1} = \frac{1}{2}(u_{i+1}^n + u_{i-1}^n) - \frac{\Delta t}{2\Delta x}(f(u_{i+1}^n) - f(u_{i-1}^n))} \tag{2}$$

In the case of our equation the sheme to solve is the follwoing one :

$$\boxed{\rho_j^t = \frac{1}{2}\left(\rho_{j+1}^{t-1} + \rho_{j-1}^{t-1}\right) - \frac{\Delta t}{2\cdot\Delta x}\left(\rho_{j+1}^{t-1}\left(1 - \frac{\rho_{j+1}^{t-1}}{R}\right)\cdot V_{\max} - \rho_{j-1}^{t-1}\left(1 - \frac{\rho_{j-1}^{t-1}}{R}\right)\cdot V_{\max}\right)} \tag{3}$$

Implementation:

Algorithm 7 EulerExplicitTrafficFlow

Input:
u_{0_x}: Initial condition function for x
Δ_x: Spatial step size
Δ_t: Time step size
T: Total time of Simulation
L: TOtal space for the Simulation
$V_{\max}$: Maximum velocity
R: Maximal density of the simulation
Output:
ρ: Matrix storing the density for each time step and time space
Algorithm:
maxT $\leftarrow \lfloor \frac{T}{\Delta_t} \rfloor + 1$

maxL $\leftarrow \lfloor \frac{T}{\Delta_x} \rfloor + 1$

$U \leftarrow$ Matrix of size(maxT, maxL)
$U[0,:] \leftarrow$ [compute $u_{0_x}(x)$ for x in range$(0, L)$]
for t **in** range(1, maxT) **do**
  for j **in** range(0, maxL) **do**
    if typeSimu == "EE" **then**
      $\rho_i^n \leftarrow U[t-1, j]$
      $\rho_{i-1}^n \leftarrow U[t-1, j-1]$
      $v_i^n \leftarrow \left(1 - \frac{\rho_i^n}{R}\right) \times V_{max}$

      $v_{i-1}^n \leftarrow \left(1 - \frac{\rho_{i-1}^n}{R}\right) \times V_{max}$
      $U[t, j] \leftarrow \rho_i^n - \frac{\Delta t}{\Delta x} \cdot \left(\rho_i^n \cdot v_i^n - \rho_{i-1}^n \cdot v_{i-1}^n\right)$
    else if typeSimu == "LF" **then**
      if $j + 1 <$ maxL **then**
$$\boxed{\rho_j^t = \frac{1}{2}\left(\rho_{j+1}^{t-1} + \rho_{j-1}^{t-1}\right) - \frac{\Delta t}{2 \cdot \Delta x}\left(\rho_{j+1}^{t-1}\left(1 - \frac{\rho_{j+1}^{t-1}}{R}\right) \cdot V_{\max} - \rho_{j-1}^{t-1}\left(1 - \frac{\rho_{j-1}^{t-1}}{R}\right) \cdot V_{\max}\right)}$$
      else
$$\boxed{\rho_j^t = \frac{1}{2}\left(\rho_0^{t-1} + \rho_{j-1}^{t-1}\right) - \frac{\Delta t}{2 \cdot \Delta x}\left(\rho_0^{t-1}\left(1 - \frac{\rho_0^{t-1}}{R}\right) \cdot V_{\max} - \rho_{j-1}^{t-1}\left(1 - \frac{\rho_{j-1}^{t-1}}{R}\right) \cdot V_{\max}\right)}$$
      end if
    end if
  end for
end for
return U

---

4 Types of Simulations Performed with ODE models

For the simulation, we conducted tests on both two cars and three cars to demonstrate various outcomes. In this section, we will initially explore the simulation involving two cars to illustrate the 'accordion phenomenon,' where the acceleration remains constant throughout the simulations. Subsequently, we will analyze the positional variations of three cars, with a specific focus on the changing acceleration over time. Interestingly, the observed phenomenon remains consistent across both types of simulations. However, the simulations involving three cars are more intriguing. For simulations involving two cars, please refer to the 'Study of Stability' section or the Annex.

4.1 Simulation with Drivers Reacting Similarly

4.1.1 Accordion Phenomenon

For this parts, we make the following assumptions:

- The first car maintains a constant speed over time.
- The second car reaction is based on the behavior of the first one.
- The acceleration of the second car is governed by a constant function, denoted as alpha, which remains fixed. In the algorithm 6 we took $\alpha_2 = 2.0$

The initial simulation we aimed to conduct was intended to demonstrate and illustrate the behavior of drivers in real-life scenarios. Figure 3 accurately depicts this phenomenon. Specifically, we Similar to real-life situations, we observe that if the second car is too far behind the first one, it decelerates. Conversely, when it is adequately distant from the first car, it increases its speed.

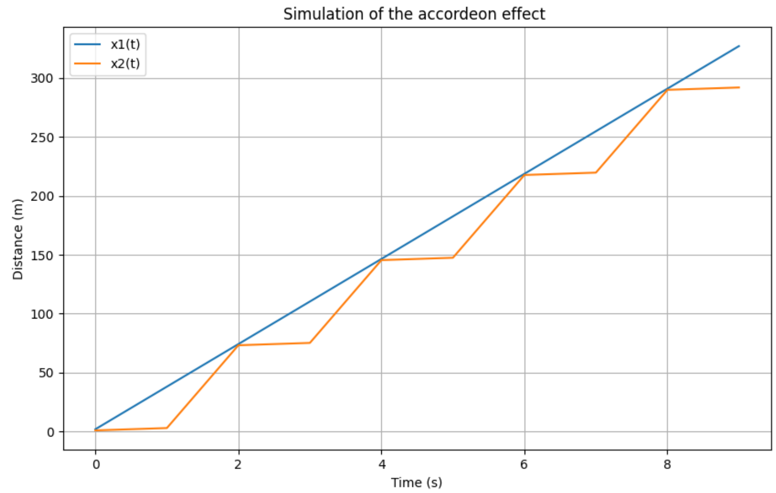


Figure 3: **Simulation of accordion phenomenon with Linear Model :** In this picture, you can observe the position evolutions of two cars, the first one (blue) maintaining a constant speed, and the second one (orange) following behind the first one. We could see that orange graph behaves like a periodic accordion with respect to time.

Similar to the previous case, applying Newell's method yields identical results. As depicted in Figure 4, the graph illustrates the accordion phenomenon, featuring distinct periods of acceleration followed by deceleration.

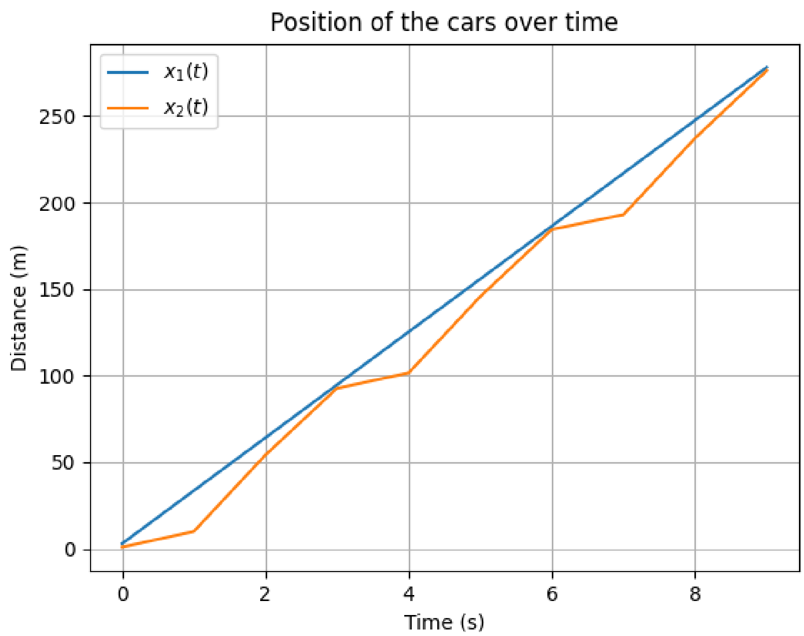


Figure 4: **Simulation of accordion phenomenon with Newell's Model :** In this picture, you can observe the positional evolutions of two cars. The first one (blue) maintains a constant speed, while the second one (orange) follows behind the first. It is evident that the orange graph behaves like a periodic accordion with respect to time.

What we can conclude from these two graphs is that, in the linear model, the acceleration and deceleration are slightly less realistic compared to Newell's method. Specifically, in the linear model, we observe that when the car is too close to another, it almost doesn't move and remains near 0 km/h, then it accelerates very rapidly. Conversely, with Newell's model, we notice that the acceleration and deceleration are more realistic as they follow a more progressive pattern.

4.2 Simulation with Drunk Drivers

In this type of simulation, we aimed to make the model more realistic by incorporating variations in the acceleration of the car.

Specifically, in this section, we consider a scenario where one of the drivers is intoxicated. Consequently, the driver's acceleration fluctuates over time. The Hypotesis are the following one

- The first car is traveling at a constant speed but is subject to some obstacles on the road at times.
- The second car (orange) is erratic as it is driven by a drunk driver, causing completely random variations in its position.
- The last one (blue) accelerates when it has distance and decelerates when it is too close.

The graph illustrates a noticeable departure from the periodic accordion phenomenon observed earlier. Instead, it becomes non-periodic and entirely unpredictable.

In Figure 5, we observe that compared to the standard model (without acceleration variations), the behavior of the car becomes entirely unpredictable and may lead to accidents. For instance, around time 20, the orange car nearly collided with the blue car. Since this model employs sinusoidal functions to simulate the erratic behavior of an intoxicated driver, instances of accidents can occasionally appear on the graph. Thus, this simulation becomes more interesting as it represents a real-life scenario.

However, it is interesting to remark that compared to Newell's method (Figure 6), the variations are very significant, potentially resulting in a lack of precision. In fact, in the Newell's model, the simulation is more realistic due to the smoothing of variations and much less abrupt changes in positions.

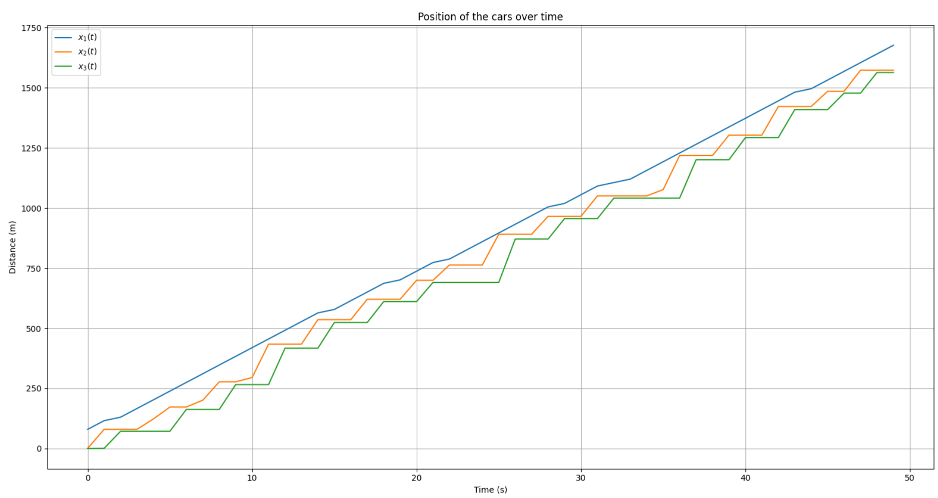


Figure 5: **Simulation of Traffic Flow with one drunk driver (Linear Model) :** In this picture, you can observe the positional changes of three cars. The first one (blue) maintains a constant speed but occasionally faces obstacles on the road. The second one (orange) follows behind the first car with a driver who is intoxicated. The last car slows down when it gets too close to the orange car and accelerates if it has enough space ahead.

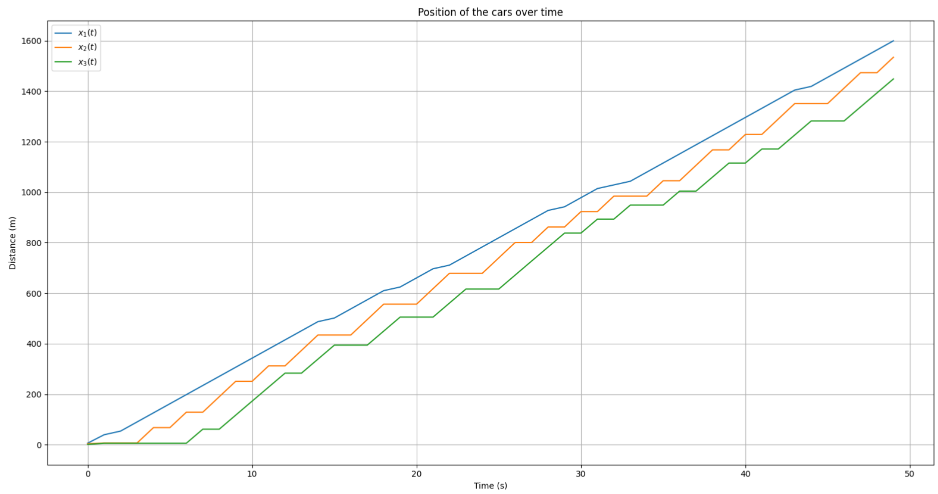


Figure 6: **Simulation of Traffic Flow with one drunk driver (Newell's Model) :** In this picture, you can observe the positional changes of three cars. The first one (blue) maintains a constant speed but occasionally faces obstacles on the road. The second one (orange) follows behind the first car with a driver who is intoxicated. The last car slows down when it gets too close to the orange car and accelerates if it has enough space ahead. You could observe thath the variations of position is more constant than in the linear model.

4.3 Simulation with Unpredictable Drivers

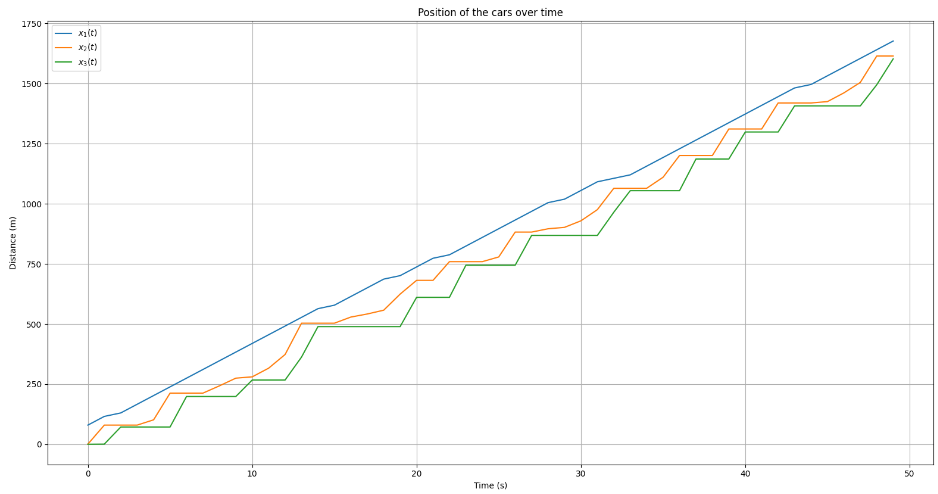


Figure 7: **Simulation of Traffic Flow with one drunk driver (Newell's Model) :** In this picture, you can observe the positional changes of three cars. The first one (blue) maintains a constant speed but occasionally faces obstacles on the road. The second one (orange) follows behind the first car with a driver who is intoxicated. The last car slows down when it gets too close to the orange car and accelerates if it has enough space ahead. You could observe thath the variations of position is more constant than in the linear model.

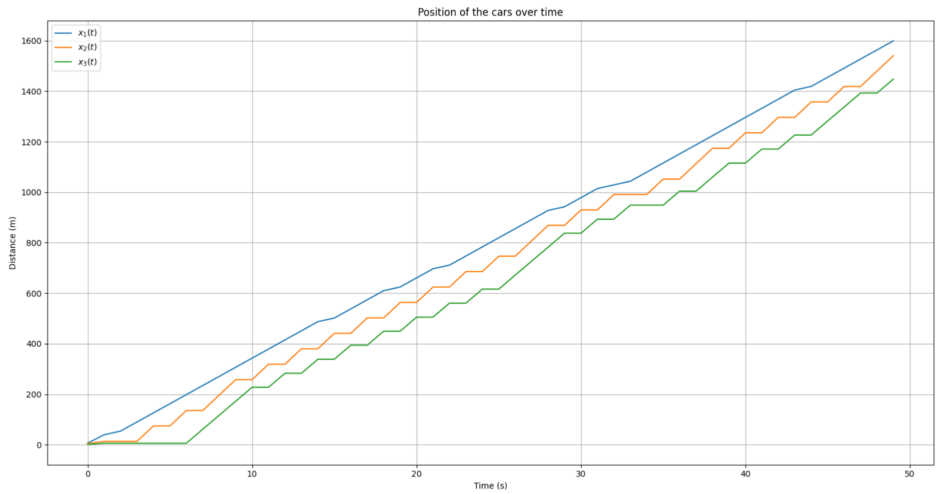


Figure 8: **Simulation of Traffic Flow with one drunk driver (Newell's Model) :** In this picture, you can observe the positional changes of three cars. The first one (blue) maintains a constant speed but occasionally faces obstacles on the road. The second one (orange) follows behind the first car with a driver who is intoxicated. The last car slows down when it gets too close to the orange car and accelerates if it has enough space ahead. You could observe thath the variations of position is more constant than in the linear model.

4.3.1 Accident Phenomenon For The Linear Model

In this scenario, the objective is to demonstrate that if a driver has a longer reaction time, an accident may occur.

In this particular case, we retained the same assumptions as before, except with a slightly lower acceleration capacity and a longer reaction time. In algorithm 6, we used $\alpha_2 = 1.75$, and in algorithm 5, we used $h = 1.5$.

As depicted in figure 9, the accident is confirmed due to a curve intersection (the position of the second car (orange) surpasses that of the first car (blue)). This indicates that the second car collided with the first car.

In this scenario, the aim is only to illustrate that an accident only becomes apparent when the distance to the preceding car is greater than that to the first car. However, in any simulation, the accident can be observed in the same way, so we are no longer particularly interested in this type of simulation For further graphical studies.

If you wish to see other graph i invite you to see the figure in annexe or to go on the GitHub project.

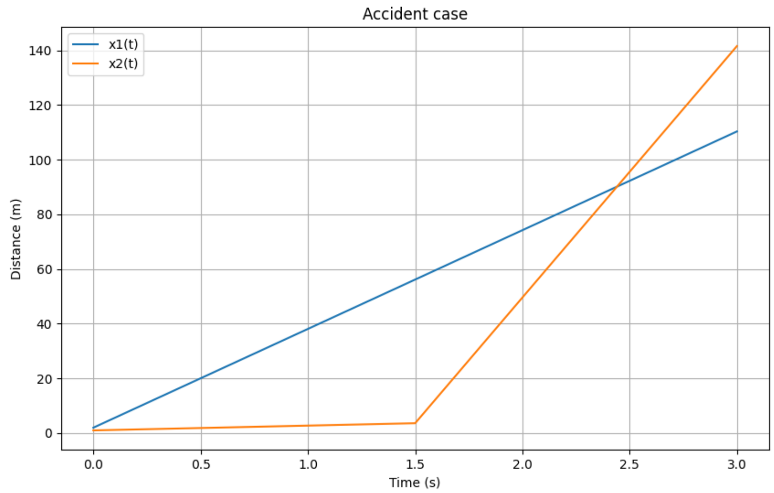


Figure 9: **Simulation of Accident case between two cars :** In this picture, you can observe the position evolutions of two cars, the first one (blue) maintaining a constant speed, and the second one (orange) following behind the first one. We could see on The graph shows that the acceleration of the orange car is too large. This implies an accident between two cars.

4.4 Study of Equilibrium, Stability, and Instability of the Solution

4.4.1 System Stability and Equilibrium for the Linear Model

In this section, the goal is to investigate the difference in distance between two cars to determine the stability of solutions and ascertain whether an equilibrium exists. To achieve this, we analyze the solutions of the following system of equations:

$$\begin{cases} \dot{d_1} & = V_1 - \alpha_2 \cdot d_1 \\ & \vdots \\ \dot{d_n} & = \alpha_n \cdot d_{n-1} - \alpha_{n+1} \cdot d_n \end{cases}$$

In our context, two cars constitute a 1D system, whereas with 3 cars, we encounter a 2D system of equations. Now, let's delve into the system's stability and equilibrium:

With 2 Cars:
The equation of interest for analysis is:

$$\dot{d_1} = V_1 - \alpha_2 \cdot d_1$$

Upon solving (7.1.1), we obtain the following equation for $d_1(t)$:

$$\boxed{d_1(t) = \frac{V_1 - V_1 \cdot e^{-\alpha_2(t-t_0)} + \alpha_2 \cdot d_1(t_0) \cdot e^{-\alpha_2(t-t_0)}}{\alpha_2}}$$

Initially, the intuition was that if $\alpha_2 < 0$, the solution would diverge as one car moves backward and the other forward. However, upon studying the equation, we observe that the terms V_1 and $\alpha_2 \cdot d_1(t_0)$ are constant and less significant.

Nevertheless, both terms, $e^{-\alpha_2(t-t_0)}$, represent exponential functions of time. For negative α_2, these exponentials grow exponentially over time.

Thus, the equation is stable and converges to the equilibrium $\boxed{d_1(t) = \frac{V_1}{\alpha_2}}$ if $\alpha_2 > 0$.

Graphically, these observations are depicted in figures 10 and 11. Notably, across all figures, there's no occurrence of solution explosion. Specifically, in figure 10, the car's position exhibits periodic behavior concerning time. Meanwhile, in figure 11, the depiction illustrates the convergence of the distance between two cars towards an equilibrium state.

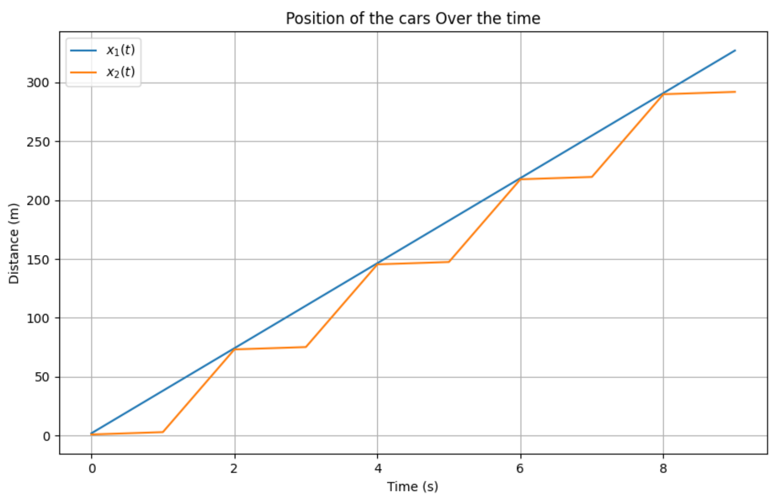


Figure 10: **<u>Realistic Case for the Linear Model with Two Cars:</u>** This simulation illustrates real-life behavior, characterized by a reaction time of approximately 1 second, achieved using a time step of 1 second instead of smaller increments in the algorithm.

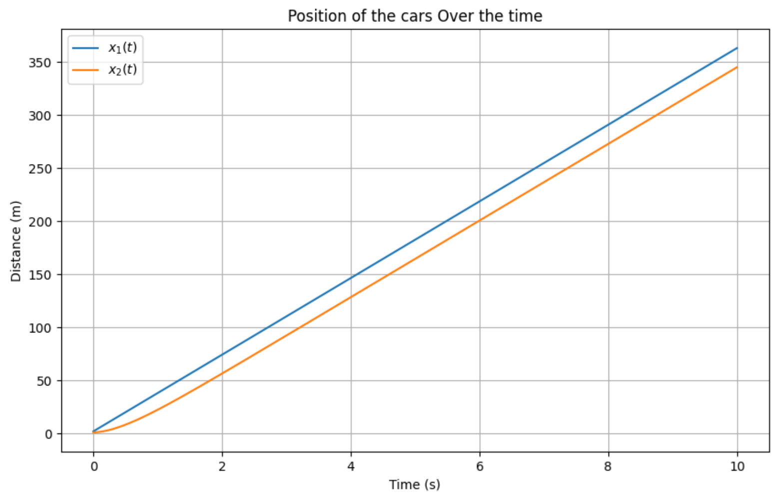


Figure 11: **Real Solution for the Linear Model with Two Cars:** This simulation depicts the accurate solution of the ODE (Ordinary Differential Equation), portraying an infinitesimally small reaction time through the use of smaller time increments in the algorithm, rather than representing real-life scenarios.

The subsequent part of our study aims to emphasize the impact of α_2 values on the stability of the system. For $\alpha_2 > 0$, the distance between two cars converges, while for $\alpha_2 < 0$, the ODE becomes unstable.

To visualize this, Figure 12 showcases different analytical solutions of the ODE with α_2 values greater than 0. Conversely, Figure 13 displays analytical solutions for α_2 values lower than 0. These plots align with theoretical predictions, demonstrating stable solutions in Figure 12, which converge to equilibrium, while unstable solutions are evident in Figure 13.

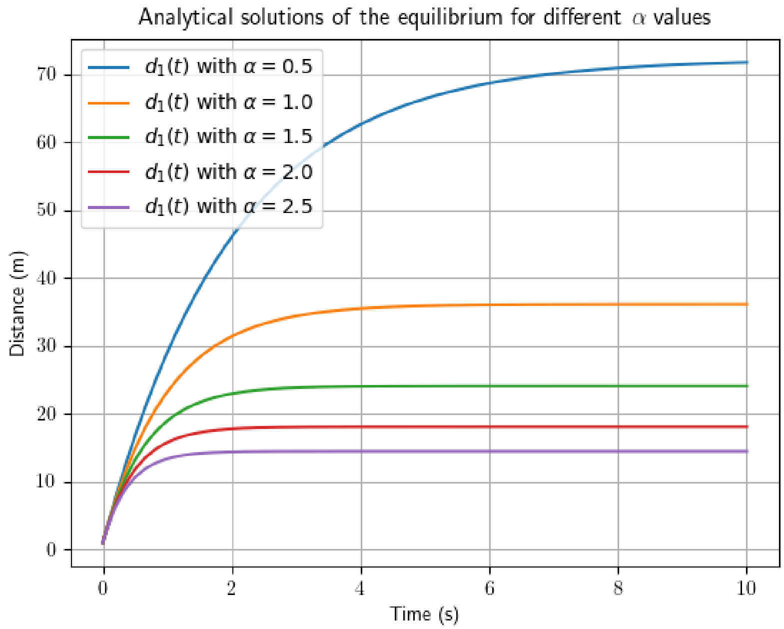


Figure 12: **Analytical Solutions for the Linear Model with Two Cars (Stability):** Different plots displaying the Analytical Solutions for various acceleration values.

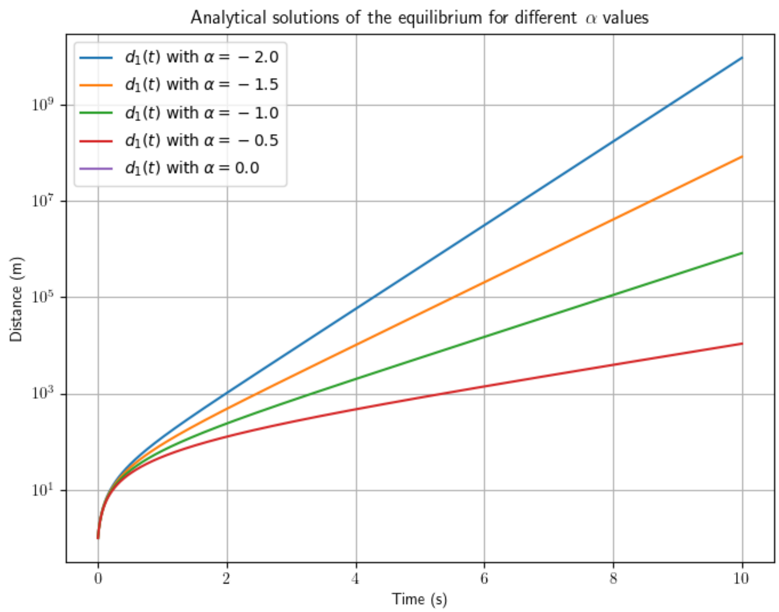


Figure 13: **Analytical Solutions for the Linear Model with Two Cars (Instability):** Different plots displaying the Analytical Solutions for various acceleration values.

Finally, the last step was to show how the solution behaves. On Figure 14, for a given acceleration greater than 0, the field of vectors converges onto the equilibrium. However, for negative acceleration, the field of vectors on Figure 15 diverges from the equilibrium.

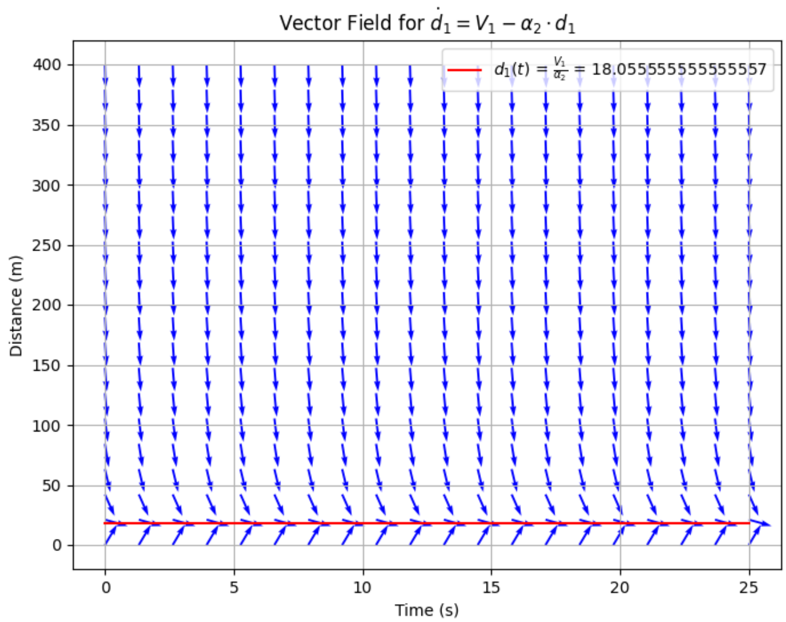


Figure 14: **Field Of Vector for the Linear Model (Stability):** On this figure, you could see the field of vectors that converges to the equilibrium.

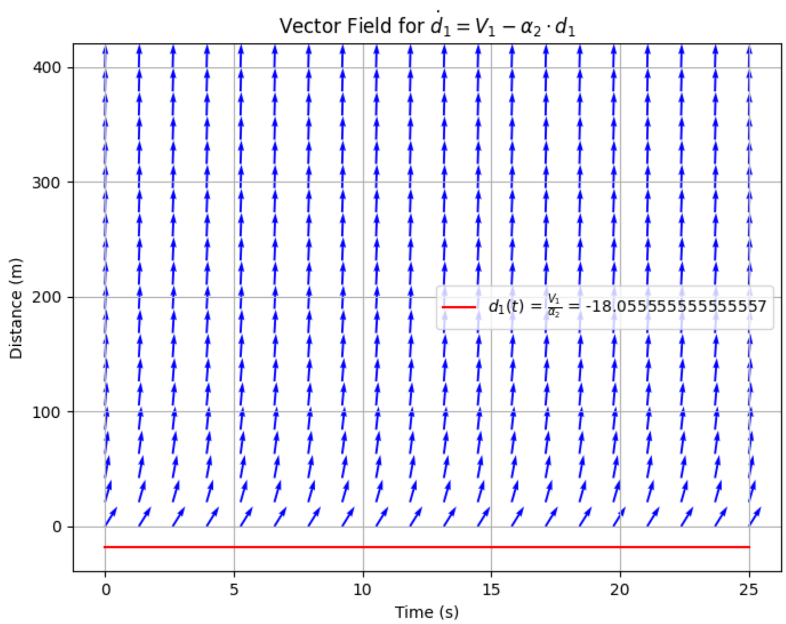


Figure 15: **Field Of Vector for the Linear Model (Instability):** On this figure, you could see the field of vectors that diverges from the equilibrium.

With 3 cars :

After the resolution (7.1.2) we obtain the next equation for $d_1(t)$ and $d_2(t)$:

$$\begin{cases} d_1(t) &= \frac{C_1(\alpha_3-\alpha_2)}{\alpha_2 e^{\alpha_2 t}} + \frac{V_1}{\alpha_2} \\ d_2(t) &= \frac{C_1}{e^{\alpha_2 t}} + \frac{C_2}{e^{\alpha_3 \cdot t}} + \frac{V_1}{\alpha_3} \end{cases}$$

we set the initial condition to $d_1(0) = m, d_2(0) = n$ So we find easily C_1 and C_2 :

$$C_1 = \frac{m\alpha_2 - V_1}{\alpha_3 - \alpha_2}$$

$$C_2 = n - \frac{V_1}{\alpha_3} - C_1$$

In the scenario involving just three cars, our analysis of the formula's terms leads us to a critical observation: divergence occurs if the value of α_2 or α_3 is less than zero. This finding confirms our initial intuition.

Given that the system of equation remains the same for 'n' cars, except for a constant, this analysis of equilibrium can extend to 'n' cars.

The visual representation in Figure 16 elucidates the equilibrium dynamics. In the left subfigure, the vector field demonstrates convergence towards the equilibrium point. Simultaneously, the right subfigure presents the Analytical Solutions derived from the system of Ordinary Differential Equations (ODEs).

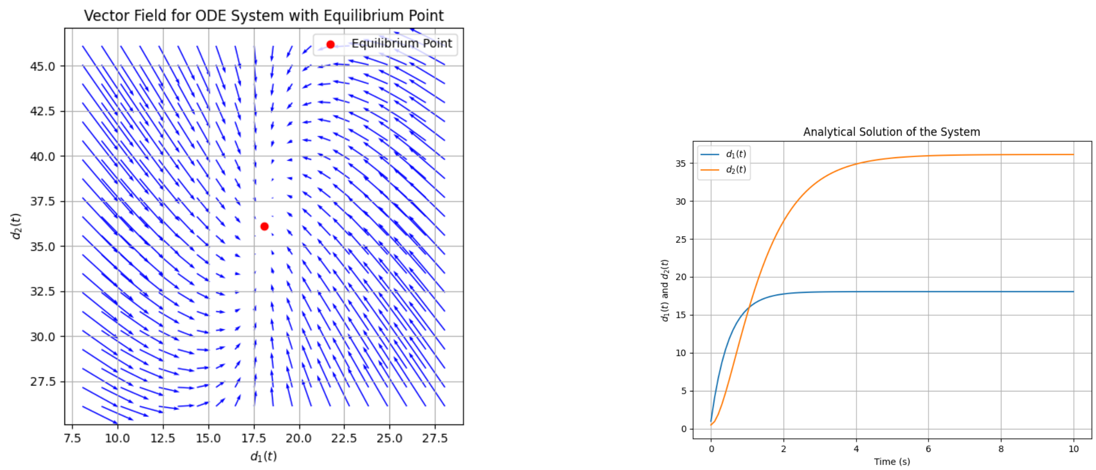
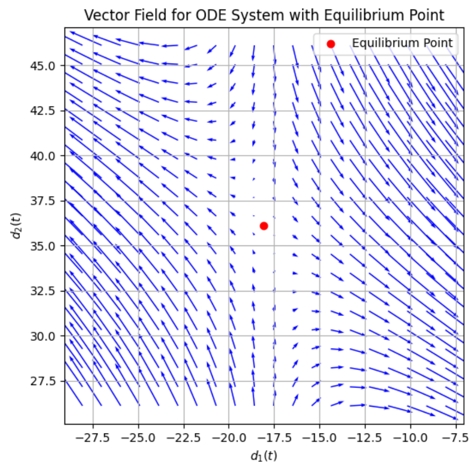


Figure 16: **Visualisation of the equilibrium:** On the left figure, you could see the vector field that converges onto the equilibrium point. On the right, the Analytical Solutions of the system of ODEs.

Examining Figure 16, it's evident that for positive acceleration, the vector field directs towards equilibrium. The distances between car 1 and 2, as well as between car 2 and 3, converge toward the equilibrium as indicated by the vector field.

Contrarily, in Figure 17, representing a scenario with negative acceleration, similar to the case of two cars, divergence is observed. Consequently, this divergence implies the absence of a feasible solution.



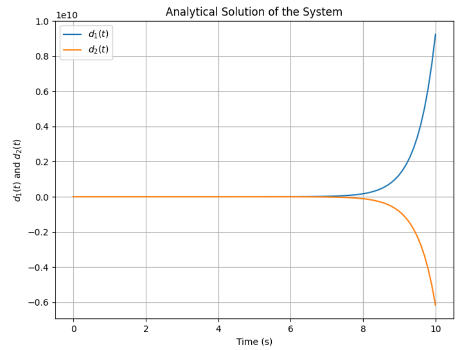


Figure 17: **Visualisation of the Divergence:** On the left figure, you could see the vector field that diverges from the theoretical equilibrium. On the right, the Analytical Solutions of the system of ODEs that completely diverge to infinity.

Figure 17 visually portrays this divergence phenomenon. The left subfigure illustrates the vector field diverging from the theoretical equilibrium point. Meanwhile, the right subfigure showcases the Analytical Solutions derived from the set of ODEs, demonstrating a complete divergence leading to infinity. This divergence pattern is consistent with the negative acceleration scenario, echoing the lack of attainable solutions, akin to the three-car case.

4.4.2 System Stability and Equilibrium for the Newell's Model

The Study of the equilibrium for two cars :

The objective here, as before, is to study the stability of the model. The 1D equation is as follows:

$$\dot{d_1}(t) = \dot{x_1}(t) - \dot{x_2}(t) = V_1 - V_2 + V_2 \cdot e^{-\frac{\alpha_2}{V_2}(d_1 - d_2^{sec})} = f(d_1, t),$$

$$\begin{cases} V_1 := \text{Maximum speed for car number 1,} \\ V_2 := \text{Maximum speed for car number 2,} \\ \alpha_2 := \text{capacity of acceleration for car number 2,} \\ d_2^{sec} := \text{security distance that car number 2 maintains.} \end{cases}$$

Firstly, let's determine the equilibrium:

$$
\begin{aligned}
V_1 - V_2 + V_2 \cdot e^{-\frac{\alpha_2}{V_2}(d_1 - d_2^{sec})} &= 0 \\
e^{-\frac{\alpha_2}{V_2}(d_1 - d_2^{sec})} &= \frac{V_2 - V_1}{V_2} \\
e^{-\frac{\alpha_2}{V_2} d_1} &= \frac{V_2 - V_1}{V_2} e^{-\frac{\alpha_2}{V_2} d_2} \\
-\frac{\lambda_2}{V_2} d_1 &= \ln\left(\frac{V_2 - V_1}{V_2} e^{-\frac{\alpha_2}{V_2} d_2}\right) \\
d_1^* &= -\frac{V_2}{\lambda_2} \ln\left(\frac{V_2 - V_1}{V_2} e^{-\frac{\alpha_2}{V_2} d_2}\right)
\end{aligned}
$$

The challenge with this equation is the inability to find an analytical solution. Consequently, we apply Lyapunov's Indirect Theorem to investigate the equilibrium.

In the context of 1D analysis, we begin by calculating $f'(d_1, t)$, which results in:

$$
f'(d_1, t) = -\alpha_2 e^{-\frac{\alpha_2}{V_2}(d_1 - d_2^{sec})}
$$

It is evident that $f'(d_1, t)$ is negative for all values of α_2 greater than zero. Therefore, in our specific scenario, the equilibrium remains stable at all times, as negative acceleration is not considered.

From the equilibrium, we also observe that the equilibrium exists if and only if $V_2 > V_1$.

Clearly, the condition for a stable equilibrium is $\alpha_2 > 0$ and $V_2 > V_1$.

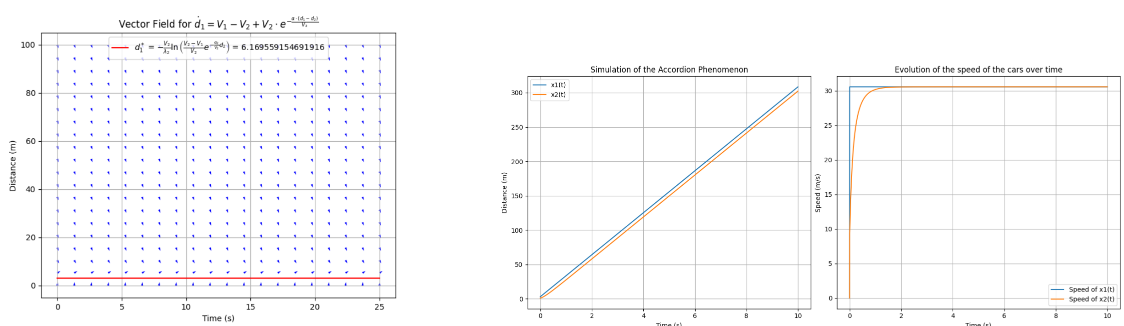


Figure 18: **Visualisation of the equilibrium:** On the left figure, you could see the vector field that converges onto the equilibrium point. On the right, the Approximation of the real Solutions of the system of ODEs.

Graphically, this can be observed in Figure 18. On the left side of the figure, the phenomenon of convergence towards equilibrium can be demonstrated when the previously performed conditions are adhered to.

On the right side, we depict the solutions of the system using Newell's equation. It is noticeable that over time, the distance between the two cars gradually increase at the beginning of the simulation, eventually approaching a constant value, which aligns with the equilibrium observed in the left figure.

The Study of the equilibrium for three cars :

$$\begin{cases} \dot{d_1}(t) = \dot{x_1}(t) - \dot{x_2}(t) = V_1 - V_2 + V_2 \cdot e^{-\frac{\alpha_2}{V_2}(d_1 - d_2^{sec})} = f(d_1, t), \\ \dot{d_2}(t) = \dot{x_2}(t) - \dot{x_3}(t) = V_2 - V_2 \cdot e^{-\frac{\alpha_2}{V_2}(d_1 - d_2^{sec})} - V_3 + V_3 \cdot e^{-\frac{\alpha_3}{V_3}(d_2 - d_3^{sec})} \end{cases}$$

$$\begin{cases} V_1 := \text{Maximum speed for car number 1,} \\ V_2 := \text{Maximum speed for car number 2,} \\ \alpha_2 := \text{capacity of acceleration for car number 2,} \\ d_2^{sec} := \text{security distance that car number 2 maintains.} \end{cases}$$

again, we apply Lyapunov's Indirect Theorem to investigate the equilibrium.
We are going to calculkate the Jacobienne matrix :

$$J_{\bar{x}} = \begin{pmatrix} -\alpha_2 e^{-\frac{\alpha_2}{V_2}(d_1 - d_2^{sec})} & 0 \\ \alpha_2 e^{-\frac{\alpha_2}{V_2}(d_1 - d_2^{sec})} & -\alpha_3 e^{-\frac{\alpha_3}{V_3}(d_2 - d_3^{sec})} \end{pmatrix}$$

However, in the 2D case, Lyapunov tells us that the equilibrium is stable if:

$$\boxed{\begin{matrix} Tr(J_{\bar{x}}) < 0 \\ \det(J_{\bar{x}}) > 0 \end{matrix}}$$

The both condition are respected iff $\lambda_2 > 0, \lambda_3 > 0$

In the case of two cars, if the conditions are respected, we can observe graphically the convergence towards the equilibrium point. As the equations apply uniformly to all cars, the equilibrium exists under the same conditions, even for the n-th car.

On the Figure 19, the equilibrium point is represented by the coordinates corresponding to the equilibrium distance on the y-axis between car 2 and 3, and on the x-axis between car 1 and 2 on the figure 20. This distance, similar to other cases, can be found on the figure 20 by calculating the difference between the positions of the cars.

Similar to the previous scenario, the plotted solution is not the actual solution but an approximation using the Euler Explicit method because obtaining an analytical solution is impossible.

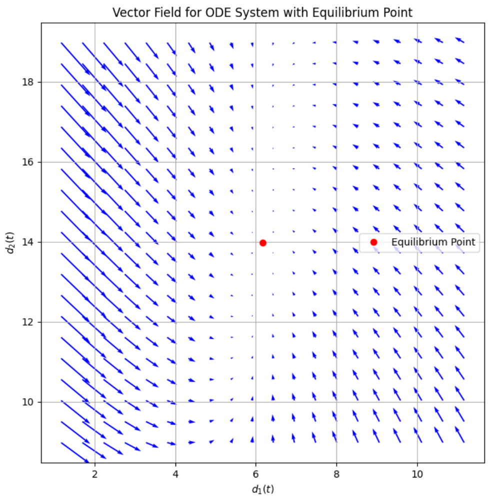


Figure 19: **Field Of Vector for the Newell's Model (Stability):** On this figure, you could see the field of vectors that converges from the equilibrium.

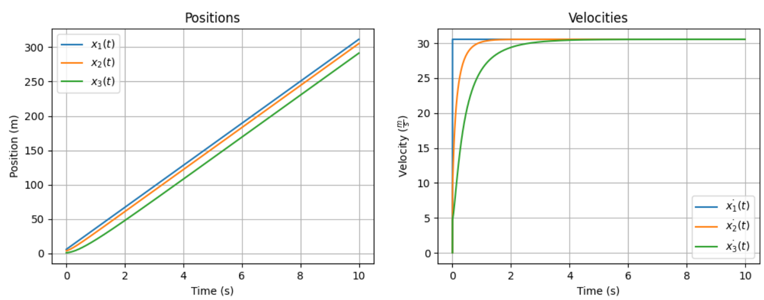


Figure 20: **Approximation of the real solution for the Newell's Model (Stability):** On this figure, you could see the distance between the curves that converges.

5 Types of Simulations Performed with PDE models

In this section, the objective is to demonstrate an alternative method for simulating traffic flow on roads. To achieve this, we will employ PDEs, considering traffic flow as analogous to fluid dynamics.

We conduct these simulations using the FDM with two types of models: the Euler Explicit and Lax-Friedrichs models. The Euler Explicit model demonstrates the simulation with free-flowing traffic, while the Lax-Friedrichs model depicts congested traffic scenarios.

We are going to see why we don't do the same simulation with the two models in following part.

5.1 Simulation Performed with Euler Explicit Model

For the simulation that we are going to do in this part, in the 1 we decide to take

$$\boxed{\begin{array}{l}\rho_0(x) = 0.2 \cdot \sin\left(2 \cdot \pi \cdot \dfrac{x}{L}\right) + 0.3 \\ \Delta t = 0.01 \\ \Delta x = 1\end{array}}$$

We decided to choose these parameters for some reasons. If interested in the function $\rho_0(x)$, we chose this function because, upon completing the simulation, we observed that for any time step, even an 'infinitely small time step,' the solution became unstable. Therefore, the function we selected allowed us to observe a phenomenon where we have peaks of density, thus enabling us to observe the creation of a 'traffic flow,' as well as its movement. More particularly, we empirically observed that the values of density at which the solutions become unstable occur when the density is greater than 0.5, indicating that the traffic flow reaches half of its capacity. Within the function, the coefficient 0.2 and the addition of 0.3 ensure that the function stays in the interval $[0, 0.5]$. For the time step, we chose 0.01 because the simulation is sufficiently precise and does not require excessive resources. As for the time interval, we selected 1 because we aim to observe the evolution at each meter.

Finally we decide to plot the solutions under two forms. The first is a Density Map, and the second is an animations that you could saw on the GitHub Link.

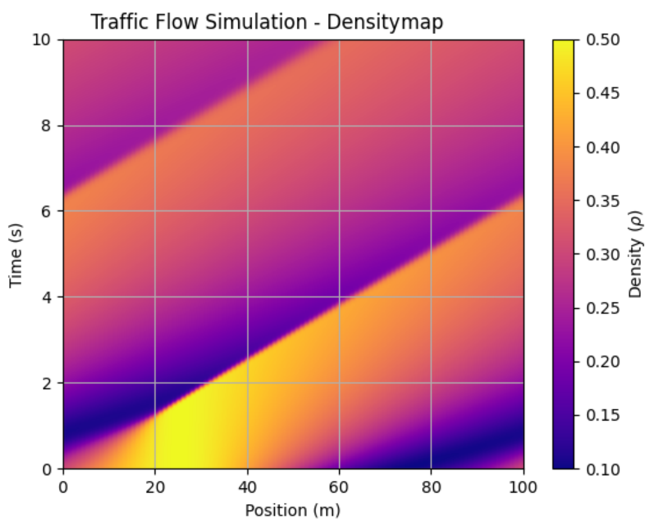
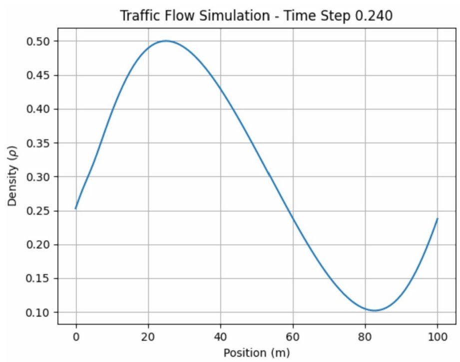
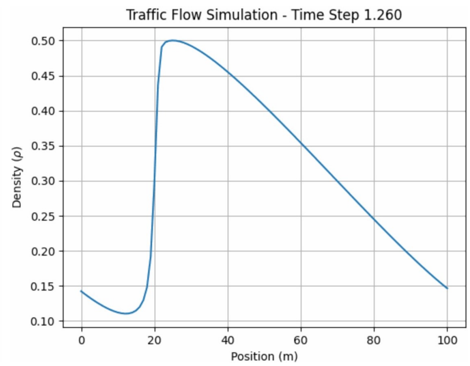


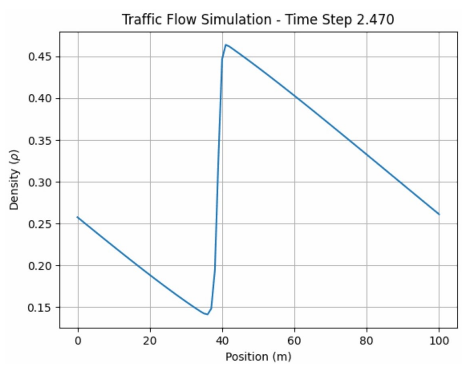
Figure 21: **Traffic Flow Simulation With Euler Explicit:** This figure illustrates the solution of the PDE at any time and position.



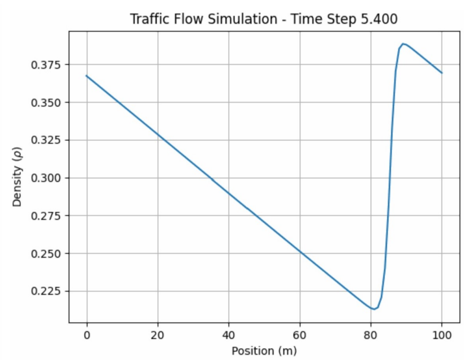
(a) T=0.240



(b) T=1.260



(c) T=2.470



(d) T=5.400

Figure 22: **Graph of the Movement of the Traffic Flow:** On this graph, you can see that the Traffic Flow is moving forward and that the information is propagating over time. There is the link to see the animation : **Traffic Flow Simulation**

These two graphs are very interesting because we can observe some phenomena. The first phenomenon is the diffusive nature concerning the model. In fact, on Figure 21, you can see that each time the peak of density appears, the density of drivers decreases. On Figure 22, this becomes evident as the solution to the equation diminishes progressively. If we continue the simulation for a longer period, we will observe the curve collapsing.

The second phenomenon is the propagation of density over time. In fact, we can observe in Figure 21 that density propagates both temporally and spatially. For instance, the initial peak of density begins at 0 meters and time 0, extending to 100 meters by 6 seconds. In the animation, you can also observe the movement of the peak. In Figure b of the animation, we witness the peak moving from 0 to 100 meters in approximately 6 seconds, as indicated by the Density map.

Finally, the last phenomenon that we can observe is the creation of peaks in density. In fact, in the animation, you can see that when the density is relatively low, a peak is formed. This phenomenon occurs because when the density is lower, cars tend to travel faster, catching up with the cars already in a peak of density. So they need to slow down and it is for this reason that we could observe this peak.

However, as discussed previously, we encounter a problem with this model. The limitation of this model is that for a density greater than 0.5, the model becomes unstable. Therefore, we need to utilize another model, the Lax-Friedrichs Model. This model is slightly more complex but resolves the issue of instability. With Jim's assistance, we will explore the behavior of high density scenarios.

5.2 Simulation Performed with Lax-Friedrichs Model

For the simulation that we are going to do in this part, in the 1 we decide to take

$$\boxed{\begin{array}{l} \rho_0(x) = 0.2 \cdot \sin\left(2 \cdot \pi \cdot \dfrac{x}{L}\right) + 0.8 \\ \Delta t = 0.01 \\ \Delta x = 1 \end{array}}$$

As we observed earlier, the model was unable to accurately simulate high density. Consequently, the aim now is to accurately simulate this particular scenario. To do this, in the function $\rho_0(x)$, the inclusion of a coefficient of 0.8 ensures that the function, represents a high density for the initial conditions, allowing us to simulate this new phenomenon.

For the time step and space step, we have opted for values identical to those used in the previous Euler Explicite method, for the same underlying reasons.

Finally, we have chosen to represent the solutions in two different forms. The first form is a density map, while the second is an animation that can be viewed via the GitHub link provided.

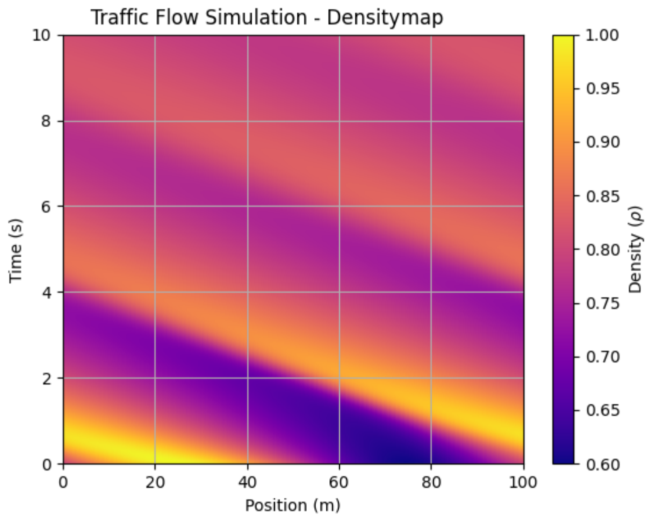
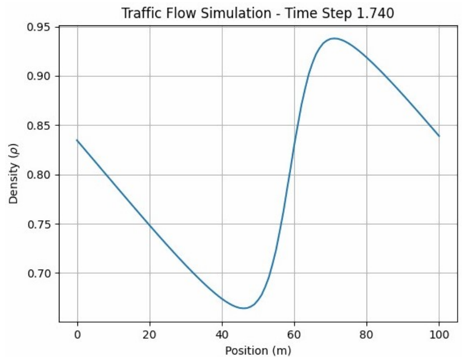
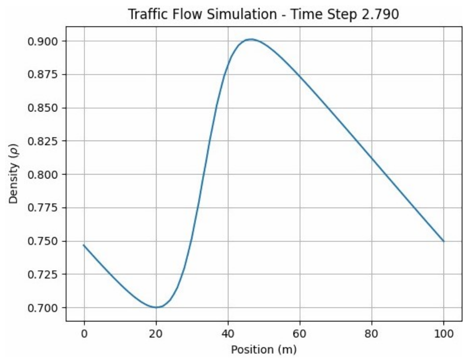


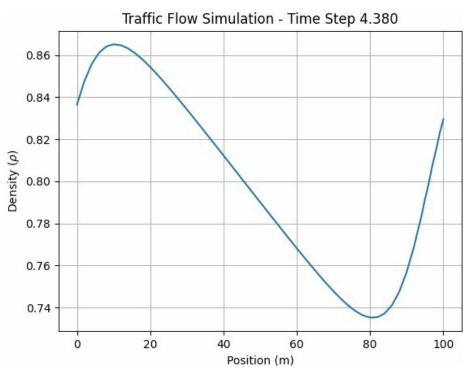
Figure 23: **Traffic Flow Simulation With Euler Explicit:** This figure illustrates the solution of the PDE at any time and position.



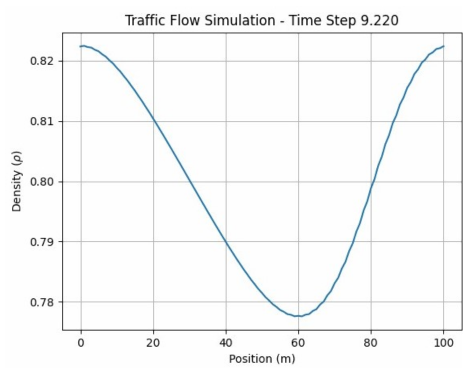
(a) T=1.740



(b) T=2.790



(c) T=4.380



(d) T=9.220

Figure 24: **Graph of the Movement of the Traffic Flow:** On this graph, you can see that the Traffic Flow is moving backwards and that the information is propagating over time. There is the link to see the animation : **Traffic Flow Simulation**

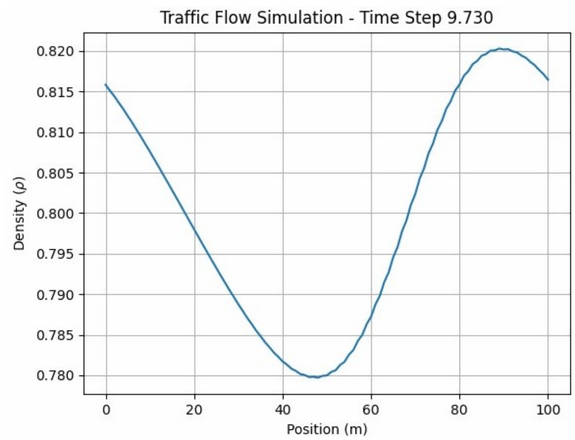


Figure 25: This graph allows to highlight the oscillation of the graph that show the dispersivity of the scheme

This simulation is quite intriguing because despite some similarities with the previous one, we observed several interesting phenomena.

The first two phenomena pertain to the scheme itself. As depicted in Figure 23 and in Simulation 24, the density decreases over time and space. This illustrates that the scheme is a diffusive model. Furthermore, the animation 25 also demonstrates oscillations in the curve towards the end of the simulation. These oscillations may be explained by the model also being a dispersive one. Hence, we can conclude that this scheme might not be well-suited for this type of equation. Nevertheless, for short simulations like the ones we conducted, this is not problematic.

The second notable phenomenon is similar to the Euler-Explicit method: the formation of traffic jams. Observing the animation, each time the density is lower than the maximum density, a traffic jam reappears. This occurrence could be explained by cars accelerating when the density decreases. However, they catch up with the car in front of them, resulting in a traffic jam.

Finally, the most intriguing phenomenon is the appearance of upside-down traffic jams. In the animation, we can see the curves moving from right to left. For example, at time t=0, the maximum density is between 15m and 35m. However, at t=1, the maximum density moves towards 0m. This backward movement can be explained by the fact that people at the head of the traffic jam can accelerate, as it's the end of the jam. Conversely, those at the rear of the traffic jam become the tail end. After a while, they become the head of the traffic jam and can accelerate out of it.

6 Summary

7 Annexe

7.1 Calculation of the Analytical solutions

7.1.1 Linear Model for Two Cars

$$\dot{d}_1(t) = V_1 - \alpha_2 \cdot d_1$$

We use the variable separation method to Solve this EDO, so we obtain :

$$\begin{aligned}
&\int_{x(t_0)}^{x(t_f)} \frac{1}{V_1 - \alpha_2 \cdot d_1(t)} d\ d_1(t) = \int_{t_0}^{t_f} dt \\
&-\frac{1}{\lambda_2}\left[\ln |V_1 - \alpha_2 \cdot d_1(t)|\right]_{x(t_0)}^{x(t_f)} = t_f - t_0 \\
&-\frac{1}{\lambda_2}\ln\left|\frac{V_1 - \alpha_2 \cdot d_1(t_f)}{V_1 - \alpha_2 \cdot d_1(t_0)}\right| = t_f - t_0
\end{aligned}$$

We could remove the $|.|$ because the sign is always the same. And we get :

$$\begin{aligned}
&\ln\left(\frac{V_1 - \alpha_2 \cdot d_1(t_f)}{V_1 - \alpha_2 \cdot d_1(t_0)}\right) = -\lambda_2(t_f - t_0) \\
&\frac{V_1 - \alpha_2 \cdot d_1(t_f)}{V_1 - \alpha_2 \cdot d_1(t_0)} = e^{-\lambda_2(t_f - t_0)} \\
&V_1 - \alpha_2 \cdot d_1(t_f) = (V_1 - \alpha_2 \cdot d_1(t_0))e^{-\lambda_2(t_f - t_0)} \\
&\boxed{d_1(t) = \frac{V_1 - [V_1 - \alpha_2 \cdot d_1(t_0)]e^{-\lambda_2(t_f - t_0)}}{\alpha_2}}
\end{aligned}$$

7.1.2 Linear Model for Three Cars

The system under matricial form coul be write like that :

$$\dot{D} = (t)\begin{pmatrix} -\alpha_2 & 0 \\ \alpha_2 & -\alpha_3 \end{pmatrix}\begin{pmatrix} d_1 \\ d_2 \end{pmatrix} + \begin{pmatrix} V_1 \\ 0 \end{pmatrix}$$

The first step is to perfom the eigenvectors of the matrix:

$$\begin{vmatrix} X + \alpha_2 & 0 \\ \alpha_2 & X + \alpha_3 \end{vmatrix} = (X + \alpha_2)(X + \alpha_3 = X^2 + X(\alpha_3 + \alpha_2) + \alpha_2\alpha_3$$

by simple resolution of the second order polynomial, we could fin two eigenvalues: $\lambda_1 = -\alpha_2, \lambda_2 = -\alpha_3$ we looking for the two eigen vectors :

$$\begin{cases} \left[\begin{array}{cc|c} 0 & 0 & 0 \\ \alpha_2 & -\alpha_3+\alpha_2 & 0 \end{array}\right] \\ \\ \left[\begin{array}{cc|c} -\alpha_2+\alpha_3 & 0 & 0 \\ \alpha_2 & 0 & 0 \end{array}\right] \end{cases}$$

It is easy to see that the two following vector are vector that verify the condition :

$$\begin{cases} u_1 = \begin{pmatrix} 1 \\ \frac{\alpha_3-\alpha_2}{\alpha_2} \end{pmatrix} \\ \\ u_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \end{cases}$$

Then, we know that a general solution of the equation is :

$\bar{D}(t) = \sum_{i=1}^{2} X_i$

where $X_i = C_i e^{\lambda_i t} u_i$,

with C_i being constants to be determined, and u_i as the eigenvectors, **and**λ_i**the eigenvalues**.

So :

$$X_1 = \begin{pmatrix} \frac{C_1(\alpha_3-\alpha_2)}{\alpha_2 e^{\alpha_2 t}} \\ \\ \frac{C_1}{e^{\alpha_2 t}} \end{pmatrix}$$

$$X_2 = \begin{pmatrix} 0 \\ \frac{C_2}{e^{\alpha_3 \cdot t}} \end{pmatrix}$$

Finally the general solution is given by :

$$\bar{X} = \begin{pmatrix} \frac{C_1(\alpha_3-\alpha_2)}{\alpha_2 e^{\alpha_2 t}} \\ \\ \frac{C_1}{e^{\alpha_2 t}} + \frac{C_2}{e^{\alpha_3 \cdot t}} \end{pmatrix}$$

Then we are going to find for particular solution :

$$\begin{cases} \dot{d}_1 & = 0 \\ \dot{d}_2 & = 0 \end{cases} \Longleftrightarrow \begin{cases} d_1^* & = \frac{V_1}{\alpha_2} \\ d_2^* & = \frac{V_1}{\alpha_3} \end{cases}$$

In clear, we have :

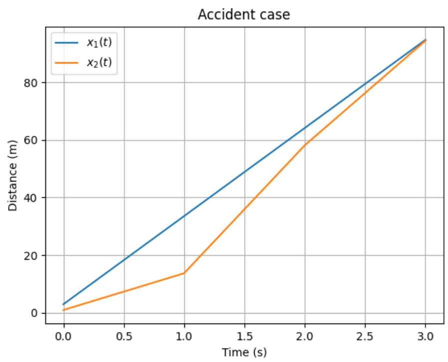
$$\begin{cases} d_1(t) & = \frac{C_1(\alpha_3-\alpha_2)}{\alpha_2 e^{\alpha_2 t}} + \frac{V_1}{\alpha_2} \\ d_2(t) & = \frac{C_1}{e^{\alpha_2 t}} + \frac{C_2}{e^{\alpha_3 \cdot t}} + \frac{V_1}{\alpha_3} \end{cases}$$

we set the initial condition to
$d_1(0) = m, d_2(0) = n$ So we find easily $C_1 and C_2$:

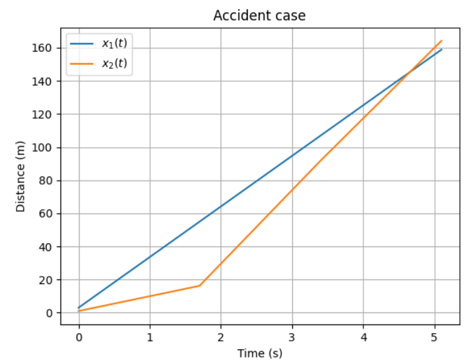
$$C_1 = \frac{m\alpha_2 - V_1}{\alpha_3 - \alpha_2}$$
$$C_2 = n - \frac{V_1}{\alpha_3} - C_1$$

7.2 Accident Simulation with the Newell's Model For Two cars

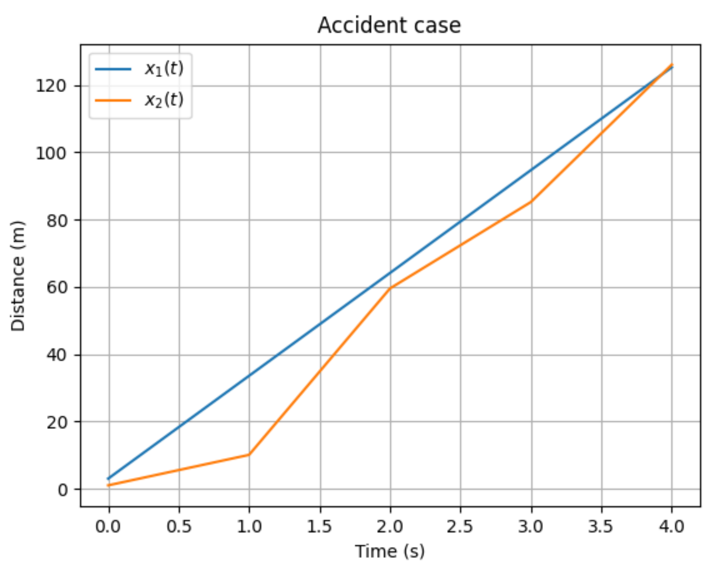
In the three subsequent figures 26a, 26b, and 26c, we can observe a similar graph pattern to that of the linear model. However, in this scenario, we have the flexibility to vary additional parameters, such as the maximum speed of the cars (for instance, introducing a new car with a more powerful motor) and the safety distance, which is an inherent aspect of the Newell model. This variation allows us to witness accidents occurring at each curve intersection on the graph.



(a) In this picture, you can observe the position evolutions of two cars, the first one (blue) maintaining a constant speed, and the second one (orange) following behind the first one. The graph shows that the acceleration of the orange car is too large. This implies an accident between two cars.



(b) In this picture, you can observe the position evolutions of two cars, the first one (blue) maintaining a constant speed, and the second one (orange) following behind the first one. However, the low reaction time of driver 2 implies an accident between two cars.



(c) In this picture, you can observe the position evolutions of two cars, the first one (blue) maintaining a constant speed, and the second one (orange) following behind the first one. However, the increasing of the maximum speed of car 2 implies an accident between two cars.

Figure 26: Simulation of Accident case between two cars