Estimating congestion and traffic patterns when planning road work

Literature Review

by

Vosse Meijssen

to obtain the degree of Master of Science
at the Delft University of Technology,

Student number:	4593901	
Project duration:	April 3, 2023 – September 21, 2023	
Thesis committee:	Prof. dr. ir. C. Vuik,	TU Delft
	Dr. A. Heinlein,	TU Delft, supervisor
	Ir. L. Portengen	CGI, supervisor

This thesis is confidential and cannot be made public until December 31, 2024.

TU Delft

Contents

Nomenclature

Abbreviations

Abbreviation	Definition	Page
DTA	Dynamic Traffic Assignment	2
DUE	Dynamic User Equilibrium	3
FR	Fundamental relation	3
LWR model	The Lighthill-Whitham-Richards traffic model	1, 3
ML	Machine Learning	1, 11
NDW	Nationaal Dataportaal Wegverkeer, the Dutch databank that handles traffic data	1
OD matrix	Origin Destination matrix	2
TA	Traffic Assignment	2
UE	User Equilibrium	3

Symbols

Symbol	Definition	Unit
q	The density of vehicles on a (piece of) road	[veh.$/m$]
q_c	A threshold density from which all vehicles drive at the same speed, and from which traffic flow will behave differently	[veh.$/m$]
q_j	The density at which traffic comes to a full stop	[veh.$/m$]
t	Time	$[s]$
u	The speed of vehicles on a (piece of) road	$[m/s]$
u_0	The speed of vehicles on an empty (piece of) road	$[m/s]$
x	The distance along a one-dimensional road	$[m]$
ϕ	The flux of vehicles through a point	[veh.$/s$]

1 Introduction

Traffic congestion is a big problem for transportation networks. It is one of the main incentives for modelling traffic, because good traffic models can help with pinpointing problems in the network. Over the past 70 years the understanding of traffic has steadily increased and traffic models have become better in modelling traffic in a realistic way, even though traffic will always be unpredictable to a certain level. Modelling traffic requires a deep understanding of the behaviour of traffic flow in certain circumstances.

The fundamental relation has been the main focus of traffic flow analysis since the 1955 paper of Lighthill and Whitham [13]. Their traffic model approached traffic as a continuous flow through a network, with a positive real-valued density and speed on every point. Such a model is called a macroscopic traffic model, and the model developed in [13] is called the Lighthill-Whitham-Richards (LWR) model. The LWR model consists of a few equations, where the *fundamental relation* has been a research topic for numerous decades [20].

Estimating the fundamental relation poses significant challenges due to the complex and random nature of traffic. Over the past few years, the Netherlands has greatly improved the amount and accessibility of their traffic data, which contributes to researching these fundamental relations. Furthermore, the in various fields like computer vision, natural language processing and different optimization problems. This achievement is also noticable in traffic model research [20].

In this paper, we propose a new approaccombination of historical data and different machine learning (ML) algorithms have shown great results in predictive traffic models [10, 1]. Using ML algorithms to learn complex patterns from data has shown success h that combines the LWR traffic flow model with ML algorithms to estimate the fundamental relation. We will build this macroscopic traffic model by using a part of an existing implementation of the LWR model. The fundamental relations are estimated using open historical traffic data, available through the web portal of Nationaal Dataportaal Wegverkeer (NDW). The end goal of this project is predicting the traffic congestion for certain road work projects. The traffic flow model should be able to model traffic in realistic work road scenarios, e.g. where one lane on a highway is closed down.

This literature study is part of the master thesis project for the Master Applied Mathematics on the Delft University of Technology, in collaboration with the Infrastructure and Asset Management Lab of CGI Netherlands.

2

Theoretical background

2.1. Introduction

Everyone is used to a weather forecast on the evening news, where high-end weather models utilize measurements to predict the weather up to a week in the future. Why is there not such a forecast on traffic congestion? Traffic is difficult to model for a few reasons. Small perturbations in traffic speeds can have big effects, like the butterfly effect. Furthermore, even the best models can't model the stochastic nature of traffic itself, which is a result of random human influence in routing, lane-switching and accidents [3, 12, 14]. Still, traffic is definitely not fully random and traffic models have improved a lot through research over the past decades.

In 1955, M. J. Lighthill and G. W. Whitham wrote the first groundbreaking paper about kinematic waves and its application in traffic models [13]. Around the same time, P. I. Richards investigated this mathematical traffic model as well [17], and the set of equations was named the LWR model (after Lighthill, Whitham and Richards). The LWR model approaches traffic as a continuous flow using a few differential equations. This approach is also known as macroscopic traffic model. The knowledge about traffic models has steadily increased until we can now distinguish three different types of traffic models; macroscopic models, microscopic models, where each vehicle is modelled independantly, and mesoscopic models, which uses ideas from the first two types [3]. In this master thesis, we will only look at macroscopic traffic models, specifically at variations of the LWR model.

All traffic models rely on some basic concepts, which will be discussed in the following section. Afterwards, a more in-depth description of the LWR model and the Gudonov scheme will be given, which is used for the numerical approximation. This chapter is concluded with the appropriate pseudocode and some benchmark problems. In the next chapter, the results of our implementation of the Godunov scheme will be discussed and we will explain how machine learning will be applied in the prediction of fundamental relations.

2.2. Basic concepts in route choice algorithms

Traffic can be modelled in different ways, depending on the situation that should be modelled. Usually, traffic models consist of a route choice algorithm and a simulation of traffic flow [3]. The route choice algorithm aims to solve some traffic assignment problem, defined as follows:

Definition 1 (Origin-Destination matrix) *An Origin-Destination (OD) matrix shows the number of vehicles that want to travel from one destination (represented by the row) to some other destination (represented by the column).*

Definition 2 (Traffic Assignment) *A Traffic Assignment (TA) problem is determining how demand traffic, usually in the form of an OD matrix, is loaded onto the network. It provides a means for computing traffic flows on the network links.*

Definition 3 (Dynamic Traffic Assignment) *A Dynamic Traffic Assignment (DTA) problem is the time-dependent extension of the traffic assignment problem, able to determine the time variations in link or* *path flows, and capable of describing how traffic flow patterns evolve in time and space in the network [3].*

Related to the Nash equilibruim in game theory, John Wardrop defined a state of equilibrium for traffic models. This is known as Wardrop's principle, Wardrop's equilibrium or user equilibrium. Furthermore, this equilibrium is a solution of the TA problem and also has a time-dependent version called the dynamic user equilibrium. Other solutions to TA and DTA problems are other ways that the vehicles complete their journey, but which are not in a user equilibrium.

Definition 4 (User Equilibrium) *A User Equilibrium (UE) means that the journey times on all the used routes are equal, and less than those which would be experienced by a single vehicle on an unused route. It is a solution to the TA problem.*

Definition 5 (Dynamic User Equilibrium) *A DTA problem has a Dynamic User Equilibrium (DUE) when the network has a UE at every moment. It is a solution to the DTA problem.*

Traffic models usually consist of two components: a DTA problem and a simulation of the routes. Finding a DUE in a traffic model is a difficult task and is sometimes impossible, but with macroscopic traffic models it is often possible to find this DUE as it approaches flow continuously instead of discreetly [5]. We will discuss this macroscopic traffic model further in the next section.

2.3. Lighthill-Whitham-Richards model

An example of a macroscopic traffic model is the Lighthill-Whitham-Richards model, or LWR model. This LWR model assumes a positive real-valued density of vehicles q and vehicle velocity u at every point in some network of roads. We can calculate the flux ϕ using the definition:

$$\phi = qu. \tag{2.1}$$

The second equation of the LWR model is the one-dimensional continuity equation, which describes the transport of some conserved quantity. More precisely, it says that the change in vehicle density on some part of a road only depends on the in- and outflux of vehicles:

$$\frac{\partial q}{\partial t} + \frac{\partial \phi}{\partial x} = 0, \tag{2.2}$$

also often notated as $q_t + \phi_x = 0$. This equation defines the behaviour of a conserved quantity q, where in general $q : \mathbb{R} \times [0, \infty) \to \mathbb{R}^n$ is the conserved quantity and $\phi : \mathbb{R}^n \to \mathbb{R}^n$ is the flux of this quantity [4]. In the case of a 1D traffic flow problem, $n = 1$. Integrating this equation on an interval $[x_1, x_2]$ gives

$$\frac{d}{dt}\int_{x_1}^{x_2} q(x,t)dx = -\int_{x_1}^{x_2} \frac{d\phi(q(x,t))}{dx}dx = \phi(q(x_1,t)) - \phi(q(x_2,t)). \tag{2.3}$$

Equation 2.3 shows that the temporal change in amount of q inside the interval $[x_1, x_2]$ is equal to the flow entering or exiting the interval at x_1 and x_2. Because this holds for any x_1 and x_2, equation 2.2 means that the amount of q (representing the amount of traffic) can only be created at the edges of the network, and cannot be created or destroyed inside the network.

The third equation of the LWR model describes the velocity u as some function of the density q. This allows for a substitution into 2.2 which leads to a differential equation solely depending on the density q [3, 16]. The relation between u and q is called the fundamental relation (FR).

The idea of a fundamental relation is an important one. It relies on the assumption that the speed of traffic on some location is only dependent on the density of traffic on that location. In the real world this is of course not true. Differences in driving style and human errors can create different traffic situations, even when the conditions are identical. In traffic models, this uncertainty will always be present [12, 14]. The LWR model looks at traffic in an aggregated form and approaches it as a uniform flow, averaging over these differences.

2.4. Different fundamental relations

The simplest example of a FR is the linear FR, where the speed of traffic scales down linearly with the density. When the density is 0, the vehicles will drive at the maximum speed u_0. When the density reaches q_j, the velocity will be 0 and traffic will come to a stop. In an equation, this reads:

$$u_{\text{linear}}(q) = u_0(1 - \frac{q}{q_j}). \tag{2.4}$$

Substituting this into 2.2 will lead to the differential equation

$$\frac{\partial q}{\partial t} + u_0 \frac{\partial(q(1-\frac{q}{q_j}))}{\partial x} = 0$$

which, given initial conditions and boundary conditions, can be solved numerically. Of course, this linear FR is not the best representation of reality. At small densities, an increase in density will not decrease the speed that much. And at high densities, the traffic speed will not immediately drop to zero, but it will just stay very low. This has lead to a few different different FRs [3], for instance, Smulders' and De Romphs' FRs, given by:

$$u_{\text{Sm}}(q) = \begin{cases} u_0(1-\frac{q}{q_j}), & \text{for } q < q_c \\ \gamma(\frac{1}{q} - \frac{1}{q_j}), & \text{for } q > q_c \end{cases} \tag{2.5}$$

$$u_{\text{DR}}(q) = \begin{cases} u_0(1-\alpha q), & \text{for } q < q_c \\ \gamma(\frac{1}{q} - \frac{1}{q_j})^{\beta}, & \text{for } q > q_c \end{cases} \tag{2.6}$$

In both of these cases, the γ is chosen such that $u(q)$ is continuous. It can be seen that both of these models use the constant q_c, which is the congestion density. Densities lower than this threshold leave enough space for cars to move around eachother, which means the average speed will stay relatively constant in that density region. When the threshold has been reached, then the vehicles are stuck in a traffic jam and the average speed will drop drastically [16].

These FRs can be shown in fundamental diagrams, which are usually plots showing the relation between q and u or ϕ.. Figure 2.1 shows these three FRs in three plots, with the relations between density ρ, velocity u and flux ϕ. There is not one FR applicable on every single piece of road, as the relation between density and velocity is not globally the same. For example, this relation depends on the surroundings, the behaviour of the average road user but also on the season and time of day. In this master thesis, we will use machine learning algorithms and historical traffic data to find the best FR for pieces of road. More about this can be found in chapter 3.

2.5. Numerical solutions of traffic flow

As we have seen in equation (2.2), the 1D traffic flow problem is a conservation problem. Conservation problems are a type of Cauchy problems where ϕ represents the flux of q.

Definition 6 (Cauchy problem) *The problem*

$$\begin{aligned} q_t + \phi(q)_x = 0, &\quad x \in \mathbb{R}, \quad t > 0, \\ q(x,0) = q_0(x), &\quad x \in \mathbb{R}, \end{aligned}$$

for some function $\phi : \mathbb{R} \to \mathbb{R}$ is called a Cauchy problem [4]. In this context, ***Cauchy data*** *represents the initial conditions $q_0(x)$ from which a unique solution can be found.*

According to [4, 18], it is possible to approximate the solution to this problem using a finite-difference method. For example, set $\phi(q)$ as the linear fundamental relation (2.4): $\phi(q) = q(1-q)$. This gives the following problem:

$$q_t + (q - q^2)_x = 0, \quad x \in \mathbb{R}, \quad t > 0, \tag{2.7}$$

$$q(x,0) = q_0(x), \quad x \in \mathbb{R}, \tag{2.8}$$

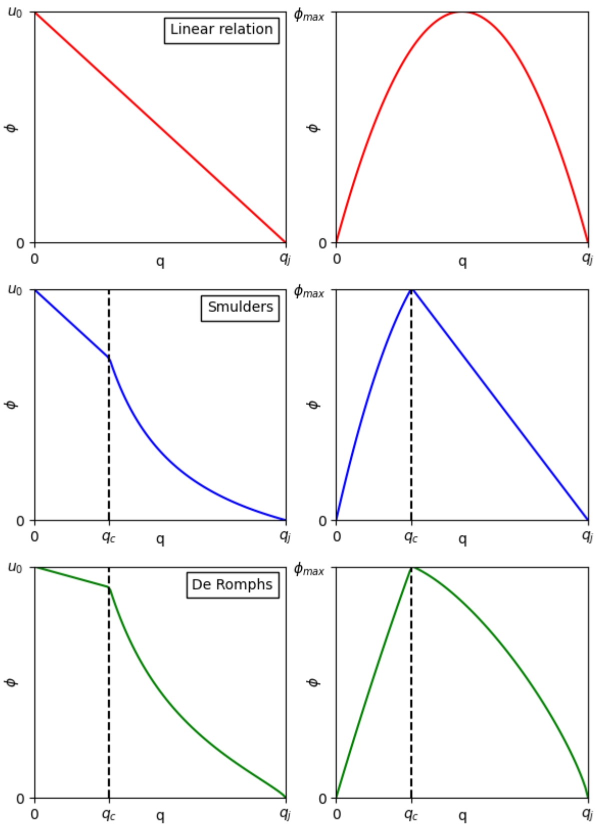

Figure 2.1: Different fundamental diagrams for three relations: the linear relation, Smulders' relation and De Romphs' relation. For simplicity in this example, $u_0 = q_j = 1$, and following [16] the other parameters have been set at a realistic value: $\alpha = 0.3$, $\beta = 0.8$, $q_c = 0.3$.

We will use the following discretization:

$$\begin{aligned} x_i &= ih & i &\in \mathbb{Z}, h > 0 \\ t_n &= nk & n &\in \mathbb{N}_0, k > 0 \\ q(x_i, t_n) &= q_i^n \end{aligned}$$

Using this discretization, [4] uses a few finite difference schemes to approximate the solutions of a few benchmark problems. A central, upwind, and Lax-Friedrichs scheme are applied, and the results are compared against the exact solution of these benchmark problems. The central scheme and upwind scheme both deviate from the exact solution around the shock. The Lax-Friedrichs scheme stays closest to the exact solution, using the numerical scheme:

$$q_i^{n+1} = \frac{1}{2}(q_{i+1}^n + q_{i-1}^n) - \frac{k}{2h}(\phi(q_{i+1}^n) - \phi(q_{i-1}^n)) \tag{2.9}$$

$$= \frac{1}{2}(q_{i+1}^n + q_{i-1}^n) - \frac{k}{2h}((q_{i+1}^n - (q_{i+1}^n)^2) - (q_{i-1}^n - (q_{i-1}^n)^2)) \tag{2.10}$$

There are some problems with a finite difference approximation of conservation problems [18]. This discretization method has difficulties around shockwaves, where the shock will diffuse over time or where the shock speed is calculated wrong [4]. Furthermore, some finite difference schemes like the upwind and central scheme don't implicitly conserve q. A better approach would be to use a finite volume approach like the Godunov method, introduced by Sergei Godunov in 1959 [6].

In finite volume, the area is devided into "volumes" with interfaces between them [15]. For 1-dimensional problems this means that the domain is devided into segments x_i. The Godunov method means keeping track of the amount of "conserved quantity" in each segment, and finding the flux at the segment boundaries every time step. The difficulty lies in finding this flux at every cell boundary. Setting the conserved value q_i constant in each cell gives a shock at every cell boundary. Each shock can be seen as a Riemann problem:

Definition 7 (Riemann problem) *A Cauchy problem with initial values*

$$q_0(x) = \begin{cases} q_l & \text{for } x < 0 \\ q_r & \text{for } x \geq 0 \end{cases} \tag{2.11}$$

where $q_l, q_r \in \mathbb{R}$ is called a Riemann problem. [8]

The discontinuity at $x = 0$ is the main focus of research about Riemann problems. In [4] this problem is solved by using the method of characteristics. They come to the conclusion that, if $q(x, t)$ is a solution, then $q(\alpha x, \alpha t)$ for some $\alpha > 0$ is a solution as well. By expressing $q(x, t) = q(\xi)$, where $\xi = \frac{x}{t}$, a few cases can be distinguished. In the light of the Godunov method, only $\phi(q(\xi = 0))$ is needed.

1. $q_l = q_r$ gives the constant solution $q(x, t) = q_0(x)$ and $\phi(q(0)) = 0$.

2. $q_l < q_r$ means that there is a higher density of traffic on the right than on the left. This higher density on the right leads to lower speeds. As traffic moves from left to right, it follows that the shockwave will stay a discontinuity. It is concluded that the solution has the form:

$$q(x, t) = \begin{cases} q_l & \text{for } x < st \\ q_r & \text{for } x \geq st \end{cases} \tag{2.12}$$

where the shock speed s is found to be

$$s = \frac{\phi(q_l) - \phi(q_r)}{q_l - q_r}. \tag{2.13}$$

This choice for s is called the Rankine-Hugoniot condition [4]. This means the value of $\phi(q(0))$ depends on the value of s; if $s > 0$, then the shock moves to the right and $q(0) = q_l$, while $s < 0$ yields $q(0) = q_r$. $s = 0$ is impossible in this situation, as that implies $\phi(q_l) = \phi(q_r)$ which is only possible if $q_l = q_r$.

3. $q_l > q_r$ has multiple weak solutions, but only one physically meaningful solution; the rarefraction wave. This means the shock will not stay a discontinuity, but it will spread out. This type of rarefraction wave is the correct solution in this situation as it satisfies the entropy condition as defined in [4] and [8]. Mathematically, this looks like this:

$$q(x, t) = \begin{cases} q_l & \text{for } x < \phi'(q_l)t \\ (\phi')^{-1}(\frac{x}{t}) & \text{for } \phi'(q_l)t \leq x \leq \phi'(q_r)t \\ q_r & \text{for } x > \phi'(q_r)t \end{cases} \tag{2.14}$$

For the Godunov method, we will need $\phi(q(0))$. This can be found from this equation:

$$q(0) = \begin{cases} q_l & \text{if } \phi'(q_l) > 0 \\ (\phi')^{-1}(0) & \text{if } \phi'(q_l) \leq 0 \leq \phi'(q_r) < 0 \\ q_r & \text{if } \phi'(q_r) < 0 \end{cases} \tag{2.15}$$

In the case of traffic models, $\phi(q)$ is a concave function and $(\phi')^{-1}(0)$ is the unique solution to $\phi'(q) = 0$ which represents the point of maximum flux. [4, 8]

As we told before, in finite volume the domain is devided into sections x_i with constant density q_i. On the interfaces between these sections is a Riemann problem. The Godunov scheme provides a method to calculate the behaviour of these shockwaves. The size of the time-step k should be chosen small enough such that the shock waves don't interact with eachother within the time interval. After the time-step, the flow is calculated between the sections and the density is again set to be constant in each section. This method has proven to stay closer to the exact solution in benchmark problems than finite-difference methods [6, 4].

2.5.1. Pseudocode

This pseudocode is the Godunov scheme for scalar conservation law problems [8].

Data: Some initial $q_0(x)$, a fundamental relation $\phi(q)$ with maximum flux $\phi(q_{max})$, a domain x and boundary values $Q_{\text{boundary point}}(t)$.
Result: An approximation of the traffic flow over time.
begin
  Discretize t as t_n size k, and x as x_i size h
  Discretize $q_0(x)$ as $q_i^0 = \frac{1}{h}\int_{x-\frac{h}{2}}^{x+\frac{h}{2}} q_0(x)dx$
  for n *in timerange* **do**
    for *all* i **do**
      We will find q_i^* at the interface between x_i and x_{i+1}
      if $\phi'(q_i^n) \geq 0$ *and* $\phi'(q_{i+1}^n) \geq 0$ **then** $q_i^* \longleftarrow q_i^n$;
      if $\phi'(q_i^n) < 0$ *and* $\phi'(q_{i+1}^n) < 0$ **then** $q_i^* \longleftarrow q_{i+1}^n$;
      if $\phi'(q_i^n) \geq 0$ *and* $\phi'(q_{i+1}^n) < 0$ **then**
        $s \longleftarrow \frac{\phi(q_i^n)-\phi(q_{i+1}^n)}{q_i^n - q_{i+1}^n}$
        if $s \geq 0$ **then** $q_i^* \longleftarrow q_i^n$;
        if $s < 0$ **then** $q_i^* \longleftarrow q_{i+1}^n$;
      if $\phi'(q_i^n) < 0$ *and* $\phi'(q_{i+1}^n) \geq 0$ **then** $q_i^* \longleftarrow q_{max}$;
    for *all interior* i **do**
      $q_i^{n+1} = q_i^n - \frac{k}{h}(\phi(q_i^*) - \phi(q_{i-1}^*))$
    for *all boundary points* i **do**
      $q_i^{n+1} = Q_i(t_{n+1})$

2.5.2. Benchmark problems

To test our own implementation of the Godunov scheme, we need some way to measure the performance against other implementation and against the exact solution. In the literature there are multiple benchmark problems that are often used to test the numerical scheme. Often they are Riemann problems with a piecewise constant initial density distribution and one initial discontinuity at $x = 0$. For simple conservation law models, the exact numerical solution is known. This makes it easy to compare different numerical schemes against eachother and against the exact solution. In [4], the first benchmark problem is

$$q_0(x) = \begin{cases} \frac{1}{4} & \text{for } x < 0 \\ \frac{1}{2} & \text{for } x \geq 0 \end{cases} \tag{2.16}$$

where, using the linear FR (2.4), the shock will move to the right with shock speed $\frac{1}{4}$:

$$q(x,t) = \begin{cases} \frac{1}{4} & \text{for } x < \frac{1}{4}t \\ \frac{1}{2} & \text{for } x \geq \frac{1}{4}t \end{cases} \tag{2.17}$$

The other benchmark problem is one with a rarefaction wave, with initial condition

$$q_0(x) = \begin{cases} \frac{3}{4} & \text{for } x < 0 \\ \frac{1}{4} & \text{for } x \geq 0 \end{cases} \tag{2.18}$$

and for $t > 0$ and the linear FR (2.4), this gives:

$$q(x,t) = \begin{cases} \frac{1}{4} & \text{for } x < -\frac{1}{2}t \\ \frac{1}{2}(1 - \frac{x}{t}) & \text{for } -\frac{1}{2}t \leq x \leq \frac{1}{2}t \\ \frac{1}{2} & \text{for } x > \frac{1}{2}t \end{cases} \tag{2.19}$$

In [4], the Godunov scheme is compared against a few finite difference methods (central scheme, Lax-Friedrichs scheme and upwind scheme). It concludes that the Godunov stays closer to the exact solution than the finite difference methods, as it can be seen in figure 2.2.

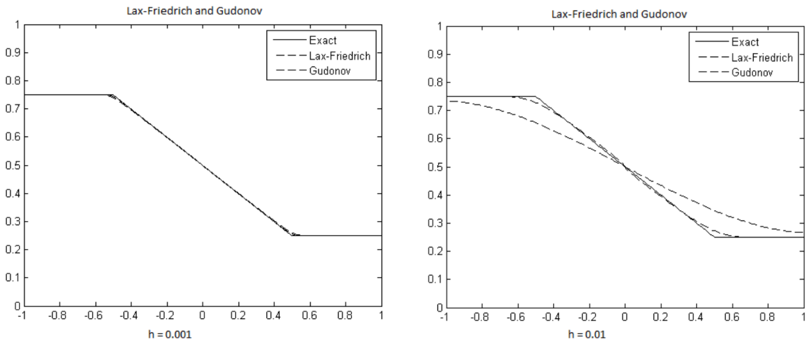

Figure 2.2: Godunov and Lax-Friedrichs scheme for a rarefaction solution with $h = 0.001$ (left) and $h = 0.01$ (right). $k = 0.001$ and $T_{end} = 1$. It can be seen that the Godunov scheme stays closer to the exact solution than the Lax-Friedrichs scheme. This plot is taken from [4], page 37.

3

Combining machine learning and fundamental relations

3.1. Implementation of the Godunov scheme

The Godunov scheme was implemented in Python following the pseudocode given in subsection 2.5.1. The full code can be found in appendix A.1. Trying the given rarefraction benchmark problem, we get the results from figure 3.1. This shows identical results to [4], which means that the implementation of the algorithm was succesful and we are able to further use the written python code.

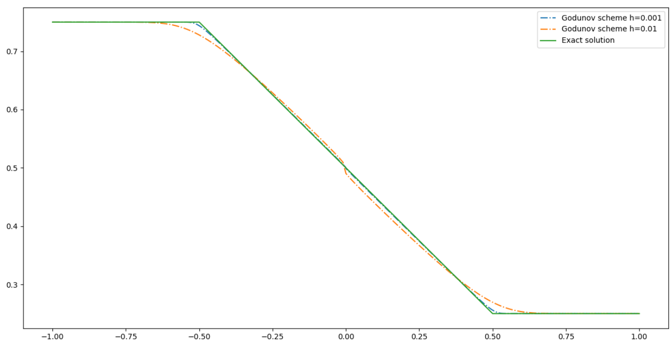

Figure 3.1: A rarefraction solution using a custom implementation of the Godunov scheme. It can be seen that the results are the same as [4].

3.2. Finding the fundamental relation

Macroscopic traffic models heavily rely on the idea of a fundamental relation, or a relation $\phi(q)$ between the traffic density q and the flux ϕ. It is also possible to find this relation by expressing the vehicle speed u in terms of q.

Numerous FRs have been introduced in this literature study, which all rely on some set of variables. For example, the Smulder's FR (given in 2.1) relies on u_0, q_j and q_c [3]. Finding the values of these variables for a certain piece of road is nothing more than fitting the function on historical road data. This type of historical road data is openly accessible online via the "Nationaal Dataportaal Wegverkeer" (NDW), which is the Dutch databank for traffic data [22]. Figure 3.2 shows the fitting of a FR based on historical data.

In this research, we will start by fitting De Romphs' FR as given in equation (2.6), and thereby fit the parameters u_0, α, β, q_c and q_j on different parts of a highway. We will use the De Romphs' FR because it has clear parameters, and its clear relation between vehicle speed and traffic density. If this method

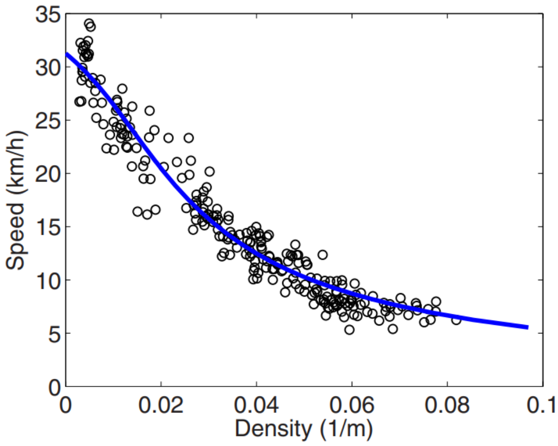

Figure 3.2: An example of a FR, fitted onto historical data. This image was taken from [7], page 249, and the FR was here derived from a model of traffic flows at intersections.

yield bad results, then I will look into other FRs like the model introduced in [7], which derives a FR from a model of traffic flows at intersections. [20] and [2] show an overview of FRs, and how they developed over the years. The latest FR is one with varying capacity, developed by Kerner and Rehborn [9].

In this research, we will start with the Smulders FR. To fit the Smulders FR using pytorch, we have to write it as a function of ReLU's and other implemented functions. We know the fundamental relation as:

$$u_{\text{Sm}}(q) = \begin{cases} u_0(1 - \frac{q}{q_j}), & \text{for } q < q_c \\ \gamma(\frac{1}{q} - \frac{1}{q_j}), & \text{for } q \geq q_c \end{cases}$$

There is some difficulty fitting this speed-density function, coming from the variable congestion density and the $1/q$ in the bottom equation. That is why we will not fit the speed-density function, but the flux-density function. We know from the definition of flux that $f = qu$. Furthermore, continuity in $q = q_c$ means that $\gamma = u_0 q_c$. This gives the following equation for flux:

$$f_{\text{Sm}}(q) = \begin{cases} u_0 q - u_0 \frac{q^2}{q_j} & \text{for } q < q_c \\ u_0 q_c - \frac{u_0 q_c q}{q_j} & \text{for } q \geq q_c \end{cases}$$

We will try to rewrite this into one equation using the ReLU function.

$$ReLU(x) = \begin{cases} 0 & x < 0 \\ x & x \geq 0 \end{cases}$$

$$ReLU(x - A) = \begin{cases} 0 & x < A \\ x - A & x \geq A \end{cases}$$

$$ReLU(A - x) = \begin{cases} A - x & x \leq A \\ 0 & x > A \end{cases}$$

Now we will rewrite both parts of $f_{\mathsf{Sm}}(q)$ as follows:

$$\begin{aligned}
u_0 q - \frac{u_0 q^2}{q_j} &= u_0(q - q_c + q_c) - \frac{u_0 q(q - q_c + q_c)}{q_j} \\
&= u_0 q_c - u_0(q_c - q) - \frac{u_0 q q_c}{q_j} + \frac{u_0 q(q_c - q)}{q_j} \\
&= u_0 q_c - u_0(q_c - q) - \frac{u_0(q - q_c + q_c) q_c}{q_j} + \frac{u_0 q(q_c - q)}{q_j} \\
&= u_0 q_c - u_0(q_c - q) - \frac{u_0 q_c^2}{q_j} + \frac{u_0(q_c - q) q_c}{q_j} + \frac{u_0 q(q_c - q)}{q_j} \\
&= u_0 q_c - \frac{u_0 q_c^2}{q_j} + (-u_0 + \frac{u_0 q_c}{q_j} + \frac{u_0 q}{q_j})(q_c - q) \\
&= u_0 q_c - \frac{u_0 q_c^2}{q_j} + (-u_0 + \frac{u_0 q_c}{q_j} + \frac{u_0 q}{q_j}) ReLU(q_c - q) \text{ for } q \leq q_c \\
u_0 q_c - \frac{u_0 q_c q}{q_j} &= u_0 q_c - \frac{u_0 q_c(q - q_c + q_c)}{q_j} \\
&= u_0 q_c - \frac{u_0 q_c^2}{q_j} - \frac{u_0 q_c}{q_j}(q - q_c) \\
&= u_0 q_c - \frac{u_0 q_c^2}{q_j} - \frac{u_0 q_c}{q_j} ReLU(q - q_c) \text{ for } q \geq q_c
\end{aligned}$$

We can see that both parts of the flux-density relation of the Smulders FR consist of $u_0 q_c - \frac{u_0 q_c^2}{q_j}$ and two ReLU terms which are zero in the opposite domain. This means that we can write the flux-density relation of the Smulders FR in one equation as:

$$f_{\mathsf{Sm}}(q) = \frac{u_0}{q_j}(q_c q_j - q_c^2 + (q_c + q - q_j) ReLU(q_c - q) - q_c ReLU(q - q_c)) \tag{3.1}$$

3.3. Machine learning

3.3.1. The usage of machine learning in traffic models

Examples of machine learning (ML) are function fitting, neural networks, clustering algorithms and others. Over the past years, different ML applications have been in the center of research in different fields, and modeling traffic has been no exception. In this research, we plan to use neural networks to find the relation between the FR on some piece of road and the FR of that piece of road with a closed lane. Furthermore, we will need some fitting technique to find the FR of some piece of road based on historical data, as demonstrated in figure 3.2.

There is a lot of research that combines machine learning techniques with traffic modelling [21, 1]. For example, computer vision algorithms are used for self-driving cars or lane assisting. Other traffic forecasting models with a longer time-frame can also use machine learning algorithms to find patterns in historical traffic data, and use that to make predictions not unlike weather predictions. An example can be found on the Keras website [10], which uses a Python implementation of an LSTM to predict the traffic situation on a single road. Furthermore, [21] shows a big list of models that uses neural networks to directly predict traffic.

3.3.2. Using machine learning techniques to predict the effect of road work

Almost all of the road work on highways in the Netherlands is planned months in advance, but for the majority of this work, it is not clear how much effect this work will have on the traffic situation. For example, there is no model (yet) that estimates how much total time will be lost because of these adjustments. This research will be about some predictive model that can estimate the effect of highway alterations like the closing of a lane, lowering the maximum speed or closing up whole roads.

There appears to be a research gap at the intersection of traffic modelling and the impact of road work. While there have been enough studies about the effect of road work on the safety of roads [19] and on the total global warming gas emissions [11], we have not found any study that predicts the effect of planned road work on traffic. Furthermore, there is enough research about different ways to model traffic [21], but none that specifically look into the effect road alterations have.

Using historical traffic data and information about previous road alterations, we can study how a FR changes when some alteration occurs. More precisely, we can study the way the parameters of FRs may change when these road alterations occur, when given inputs that describe the road alteration. Examples of these inputs can be the amount of lanes or the maximum speed before and after an alteration. We can use historical data to train a neural network that can predict these changes in parameters. This historical data should contain traffic speed data of pieces of road where alterations have occured, before and during this alteration. This neural network is then applicable to other pieces of road, so we can finally be able to predict the effect of an alteration on pieces of road where there has never been such an alteration. A diagram of this algorithm is shown in figure 3.3.

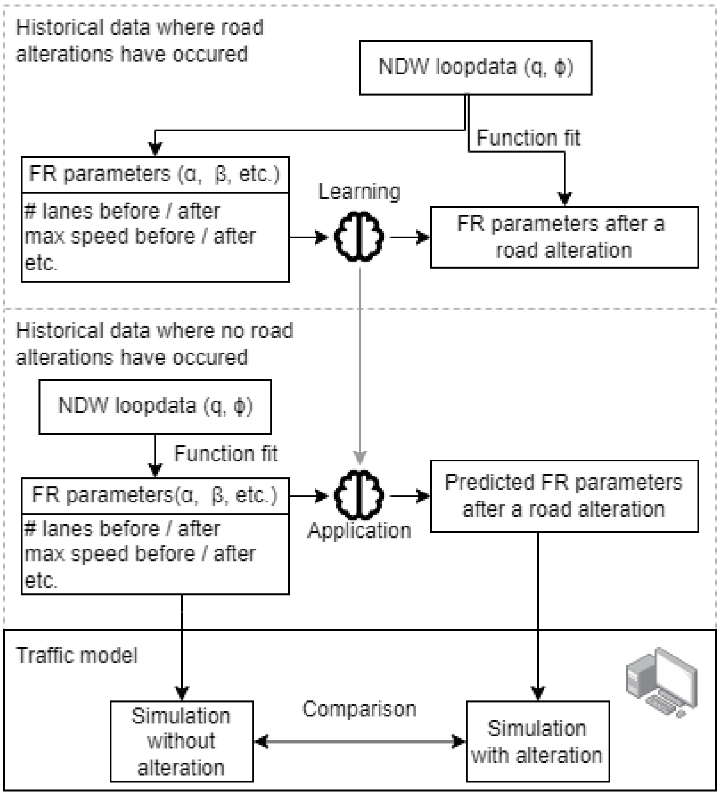

Figure 3.3: This is a basic diagram showing how machine learning can be applied to predict the effect of road alterations. A neural network is used to learn the way that the FR parameters change after a road alteration like a lane closing. This is applied to another FR, which results in a predicted FR. In the last step, a macroscopic traffic model is run once with the normal FR and once with the predicted FR. Finally, the results of these simulations are compared.

This approach to predicting the effect of alterations to the road situation is different from a normal macroscopic traffic model because it uses machine learning to express the effect of the alterations on the FR for that piece of road, while still using a macroscopic traffic model for an accurate traffic model. Because of this, it is possible to quickly make rough predictions about the effect of alterations. Eventually this should help planning road work in a more sophisticated way.

3.4. Research questions

The main objective of this research is to model the effect of alterations to the road situation using a combination of traffic models and ML algorithms. Estimating the effect of these alterations will improve the planning process of road work. This has lead to the research question:

> How can mathematical models, specifically the combination of traffic models and machine learning algorithms, be used to improve estimates of the effect of road work on traffic?

To answer this question, I want to look into different subquestions.

1. What is a measure of performance of different traffic models and ML algorithms in terms of their ability to accurately estimate the effects of road work on traffic?
2. How can macroscopic traffic models be used as a framework to estimate the effect of road work on traffic using ML?
3. How can we use ML algorithms to identify and predict the impact of road work based on historical data?
4. How can the insights gained from the traffic model be used to improve the efficiency of road work planning processes?

By answering these questions, we aim to construct a functional model that can simulate basic changes to the road situation of a normal Dutch highway. The required data consists of historical traffic data of the NDW, and a list of road work activities.

References

[1] Nour Alqudah and Qussai Yaseen. "Machine Learning for Traffic Analysis: A Review". In: *Procedia Computer Science* 170 (2020). The 11th International Conference on Ambient Systems, Networks and Technologies (ANT) / The 3rd International Conference on Emerging Data and Industry 4.0 (EDI40) / Affiliated Workshops, pp. 911–916. ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2020.03.111. URL: https://www.sciencedirect.com/science/article/pii/S1877050920305494.

[2] Arnaud Banos et al. "Multiscale Modeling: Application to Traffic Flow". In: Jan. 2017, pp. 37–62. ISBN: 9781785481574. DOI: 10.1016/B978-1-78548-157-4.50002-9.

[3] Jaume Barcelo. *Fundamentals of Traffic Simulation*. Jan. 2010. ISBN: 978-1-4419-6141-9. DOI: 10.1007/978-1-4419-6142-6.

[4] Bjørnar Dolva Bergersen. "Numerical Solutions of Traffic Flow on Networks". In: (2014).

[5] K. Fox. "Is micro-simulation a waste of time?" In: (2008).

[6] Sergei K. Godunov and I. Bohachevsky. "Finite difference method for numerical computation of discontinuous solutions of the equations of fluid dynamics". In: *Matematičeskij sbornik* 47(89).3 (1959), pp. 271–306. URL: https://hal.science/hal-01620642.

[7] Dirk Helbing. "Derivation of a fundamental diagram for urban traffic flow". In: *The European Physical Journal B* 70 (2009), pp. 229–241.

[8] Ansgar Jüngel. "Modeling and Numerical Approximation of Traffic Flow Problems". In: *Lecture Notes (preliminary version)* (2002).

[9] B. S. Kerner and H. Rehborn. "Experimental features and characteristics of traffic jams". In: *Phys. Rev. E* 53 (2 Feb. 1996), R1297–R1300. DOI: 10.1103/PhysRevE.53.R1297. URL: https://link.aps.org/doi/10.1103/PhysRevE.53.R1297.

[10] Arash Khodadadi. *Traffic forecasting using graph neural networks and LSTM*. 2021. URL: https://keras.io/examples/timeseries/timeseries_traffic_forecasting/ (visited on 12/28/2021).

[11] Philippe Lepert and Francois Brillet. "The overall effects of road works on global warming gas emissions". In: *Transportation Research Part D: Transport and Environment* 14.8 (2009), pp. 576–584.

[12] Guopeng Li, Victor L. Knoop, and J. W. C. van Lint. "Estimate the limit of predictability in short-term traffic forecasting: An entropy-based approach". In: *Transportation Research Part C: Emerging Technologies* 138 (2022), p. 103607. ISSN: 0968-090X. DOI: https://doi.org/10.1016/j.trc.2022.103607. URL: https://www.sciencedirect.com/science/article/pii/S0968090X22000535.

[13] M. J. Lighthill and G. B. Whitham. "On Kinematic Waves. II. A Theory of Traffic Flow on Long Crowded Roads". In: *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences* 229.1178 (1955), pp. 317–345. ISSN: 00804630. URL: http://www.jstor.org/stable/99769 (visited on 07/12/2023).

[14] J.W.C. van Lint. *Data gedreven modelleren van verkeer en vervoer*. Delft, The Netherlands: Delft University of Technology, 2021.

[15] D. Ngoduy, S. P. Hoogendoorn, and Henk J. Van Zuylen. "Comparison of Numerical Schemes for Macroscopic Traffic Flow Models". In: *Transportation Research Record* 1876.1 (2004), pp. 52–61. DOI: 10.3141/1876-06. eprint: https://doi.org/10.3141/1876-06. URL: https://doi.org/10.3141/1876-06.

[16] S. Paul and V. Knoop. *Traffic Flow Theory and Simulation*. Delft, The Netherlands: Delft University of Technology, 2016.

[17] Paul I Richards. "Shock waves on the highway". In: *Operations research* 4.1 (1956), pp. 42–51.

[18] J. Ngoumou S. Heigl D. Hubber. *Godunov methods in GANDALF*. 2015. URL: https://gandalfcode.github.io/gandalf-school/talks/GodunovMethods.pdf (visited on 10/28/2015).

[19] Athanasios Theofilatos et al. "Meta-analysis of the effect of road work zones on crash occurrence". In: *Accident Analysis & Prevention* 108 (2017), pp. 1–8.

[20] Femke van Wageningen-Kessels et al. "Genealogy of traffic flow models". In: *EURO Journal on Transportation and Logistics* 4.4 (2015), pp. 445–473. ISSN: 2192-4376. DOI: https://doi.org/10.1007/s13676-014-0045-5. URL: https://www.sciencedirect.com/science/article/pii/S2192437620301114.

[21] Eleni I. Vlahogianni, Matthew G. Karlaftis, and John C. Golias. "Short-term traffic forecasting: Where we are and where we're going". In: *Transportation Research Part C: Emerging Technologies* 43 (2014). Special Issue on Short-term Traffic Flow Forecasting, pp. 3–19. ISSN: 0968-090X. DOI: https://doi.org/10.1016/j.trc.2014.01.005. URL: https://www.sciencedirect.com/science/article/pii/S0968090X14000096.

[22] Nationaal Dataportaal Wegverkeer. *NDW open data*. 2023. URL: https://opendata.ndw.nu/ (visited on 10/05/2023).

A

Python code for a Godunov scheme

A.1. godunovfunctions.py

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt


# functions
# Basic Fundamental Relation class
class FR:
    def __init__(self) -> None:
        self.q_max = 0
        self.f_max = 0
        pass

    def f(self, q):
        pass

    def f_der(self, q):
        pass

    def find_max(self):
        # Finds the unique solution u to f'(u) = 0
        pass


# Linear fundamental relation
class Linear(FR):
    def __init__(self) -> None:
        self.qmax = 0.5
        self.fmax = self.f(0.5)
        pass

    def f(self, q):
        return q - q * q

    def f_der(self, q):
        return 1 - 2 * q


# Plot functions
def plot_density(x, q):
    assert len(q) == len(x)
    plt.plot(x, q)
    plt.show()
```

A.2. test.py

```
from godunovfunctions import *
from tqdm import tqdm

# Settings
dt = 0.01  # Also known as k
xmin = -30
xmax = 30
xlen = 500
T = 15
periodic_BC = False

# Set up variables
# x = np.arange(
#     start=  -1,
#     stop=   1+dx,
#     step=   dx,
# )
x = np.linspace(start=xmin, stop=xmax, num=xlen)
dx = (xmax - xmin) / (xlen - 1)  # Also known as h

q = np.zeros_like(x)

# Set initial values
q[: int(len(q) / 2)] = 0.5
q[int(len(q) / 2) :] = 0.25
# q = 0.25 + 0.25 * np.exp(-0.01 * x**2)  # "Formation of a traffic jam"

# Choose FR
fr = Linear()

# Time loop
timesteps = int(T / dt)
for timestep in tqdm(range(timesteps)):
    f = fr.f(q)
    f_der = fr.f_der(q)

    # Find all q* (or actually, find all f(q*))
    f_q_star = np.zeros_like(q)
    for i in range(xlen):
        # When looking at q*[i], we need info from q[i] and q[i]+1
        # Edge case at i_max: in this case, q[i_max] will be constant
        # Four cases:
        if f_der[i] >= 0 and f_der[(i + 1) % xlen] >= 0:
            f_q_star[i] = f[i]
        elif f_der[i] < 0 and f_der[(i + 1) % xlen] < 0:
            f_q_star[i] = f[(i + 1) % xlen]
        elif f_der[i] >= 0 and f_der[(i + 1) % xlen] < 0:
            s = (f[(i + 1) % xlen] - f[i]) / (q[(i + 1) % xlen] - q[i])
            if s >= 0:
                f_q_star[i] = f[i]
            else:
                f_q_star[i] = f[(i + 1) % xlen]
        elif f_der[i] < 0 and f_der[(i + 1) % xlen] >= 0:
            f_q_star[i] = fr.fmax

    # Now that q* is known, we can calculate the new q values
    new_q = np.zeros_like(q)
    if periodic_BC:
        for i in range(xlen):
            new_q[i] = q[i] - dt / dx * (f_q_star[i] - f_q_star[(i - 1) % xlen])
    else:
        new_q[0] = q[0]
        new_q[-1] = q[-1]
        for i in range(1, xlen - 1):
            new_q[i] = q[i] - dt / dx * (f_q_star[i] - f_q_star[i - 1])

    q = new_q

plot_density(x, q)
```