Continuous Modeling and Optimization Approaches for Manufacturing Systems

Inauguraldissertation
zur Erlangung des akademischen Grades
eines Doktors der Naturwissenschaften

der Universität Mannheim

vorgelegt von

Dipl.-Math. Patrick Schindler
aus Worms

Mannheim 2014

Dekan:	Professor Dr. Heinz Jürgen Müller, Universität Mannheim
Referent:	Professor Dr. Simone Göttlich, Universität Mannheim
Korreferent:	Professor Dr. Axel Klar, Technische Universität Kaiserslautern

Tag der mündlichen Prüfung: 3. April 2014

Abstract

This thesis is concerned with two macroscopic models that are based on hyperbolic partial differential equations (PDE) with discontinuous flux functions.
The first model describes the material flow of an entire production line with finite buffers. We consider different solutions of the model, present the novel DFG-method (Discontinuous Flux Godunov), and compare the results with other established numerical methods. Additionally, we investigate a restricted optimization problem with respect to partial differential equations with discontinuous flux functions and consider two different solution approaches that are based on the adjoint method and the mixed integer problem (MIP). Further, we extend the model and its optimization problem to network structures.

The second model describe the material flow on conveyor belts with obstacle interactions. We introduce a novel two dimensional model with a discontinuous and a non-local flux function. We consider a finite volume method and the discontinuous Galerkin method for solving this model. Finally, we validate the model with real data and present a numerical study with respect to the introduced solution methods.

Zusammenfassung

Die vorliegende Dissertation untersucht im wesentlichen zwei makroskopische Modelle welche auf hyperbolische partielle Differentialgleichungen (engl. PDE) mit unstetigen Flussfunktionen basieren.
Das erste Modell beschreibt vereinfacht den Materialfluss in einer Produktionslinie mit finiten Puffern. Wir untersuchen mögliche Lösungen des Modells, präsentieren das neuartige DFG - Verfahren (engl. Discontinuous Flux Godunov) und vergleichen die Resultate mit anderen gängigen numerischen Methoden. Zusätzlich untersuchen wir restringierte Optimierungsprobleme bezüglich der partiellen Differentialgleichung mit unstetigem Fluss und betrachten zwei Lösungsansätze basierend auf den Adjungiertenverfahren und der gemischt ganzzahligen Optimierung (engl. MIP). Außerdem erweitern wir das Modell und deren Optimierungsproblem auf Netzwerkstrukturen.

Das zweite Modell beschreibt den Materialfluss auf Fließbändern mit Hindernis-Interaktionen. Wir führen ein neues zweidimensionales Modell mit unstetiger und nicht lokaler Flussfunktion ein. Es werden ein Finites Volumen Verfahren und das Discontinuous Galerkin Verfahren zur Lösung betrachtet. Abschließend zeigen wir eine Validierung des Modells mit Realdaten und präsentieren eine weitere numerische Studie bezüglich der vorgestellten Lösungsmethoden.

Acknowledgements

First of all, I would like to show my greatest appreciation to Prof. Dr. Simone Göttlich who provided me with in-depth discussions, valuable comments and constructive feedback. Without her guidance and persistent help, this thesis would not have been possible.

I would also like to offer my special thanks to Prof. Dr. Axel Klar for his useful suggestions, practical advice, great expertise and being co-referee.

Moreover, I want to thank to Prof. Dr. Oliver Kolb, Dr. Veronika Schleper, Simon Hoher, Sebastian Kühn and Peter Schillen for their wide-ranging discussions, insightful comments and their contribution to this work.

Special thanks also to Prof. Dr. Andreas Neuenkirch for favourably answering the question for being third examiner.

Also, I would also like to express my gratitude to my family for their moral support and warm encouragements.

Finally, I would like to express my gratitude to Ann-Christin who provided me with her deep understanding, tolerance and patience.

Thanks to the financial support of DAAD (research grant no. 57049018).

Contents

Introduction

In order to manufacture products with certain standards, for example, quality, delivered quantity, requests of customers, the entire production process needs to be planned and controlled in detail. However, the planning of a manufacturing system is a wide field of complex tasks that contains, e.g., constructions and verifications of single modules of production units, controlling of the entire product or material flow, and more. Therefore, applications of simulation tools based on mathematical models are helpful in planning, evaluating, and controlling of manufacturing plants and production processes.

In this work, mainly two approaches are introduced for the modeling of production processes using the fact that a high number of products (goods) can be considered as a continuous material flow: firstly, an entire production network including finite buffers and deterministic machine failures; and secondly, a material flow model on conveyor belts with obstacles including congestion formations.

Mathematical models based on continuous equations, in general partial differential equations (PDEs), or in some cases conservation laws, have a broad range of applications, for example, traffic flow [5, 6, 19, 61, 64, 68], pedestrian flow [20, 21], or gas and fluid dynamics [8, 10].
Indeed, simulations of manufacturing systems are a mighty tool to organize industrial processes. Many mathematical models are discrete and based on consideration of individual parts, for example, discrete event simulators [3] and mixed integer models [52, 94, 101]. The drawback of these approaches however is the enormous computational effort for a high amount of parts. Continuous models use an averaged quantity as density (parts per length), and the dynamic is prescribed by a material flux (parts per time). Various continuous models prescribing production processes are investigated in the last decades, for example, [3, 4, 25, 26, 44, 46]. These models describe production lines or networks as a coupling of several individual production units consisting of one processor with a buffer in front. However, many of these models use the assumption of an unlimited buffers which are not realistic in various applications. Whereas, few models [4, 59] are able to prescribe production processes with limited buffers by using discontinuous conservation laws. Utilization of discontinuities seems to be justified for different applications in production processes. We illustrate this with an example of a buffer in a production line. In general, a buffer can be prescribed by two different states, namely the buffer capacity is reached or not reached.

Therefore, unprocessed goods can be stacked in the buffer if the maximal buffer capacity is not reached. In that case, the production process works straightforward. However, if the buffer is full, then we observe a tailback in the production process similar to traffic flow problems.

In this work, we study a model based on a discontinuous conservation laws and investigate an extension to a novel network model. In the latter case, the transport of quantities are prescribed by a conservation law on each edge of the network. Thereby, the considered quantities flow together at the intersection vertices by certain rules and move further to other edges of the network. Generally, this process is representable by certain coupling conditions on the network vertices that fulfills mass conservation. Already various models including conservation laws on network topologies are investigated, e.g., traffic flow [19, 61, 62, 64, 68], gas and water pipe lines [8, 22], telecommunication networks [27]. Furthermore, these network models use conservation laws with continuous flux functions. So far, however, there has been discussion about network coupling conditions for discontinuous conservation laws.

In general, a numerical computation and solving of discontinuous flux PDEs is a challenging task, since the most numerical solvers require a continuous flux function. A common way of solving such equations is the usage a regularized equation with a continuous flux and solve it by a suitable numerical method. Depending on the refinement of the regularization, a common numerical method needs a high number of iterations that yields a enormous computational effort. Obviously, there is a need for an alternative solution scheme that computes the discontinuous conservation law in an efficient way without any regularization.

Another important application in manufacturing products is decision making with aid of optimization problems, for example, minimizing the buffers, fulfilling a demand, finding an optimal time interval for a maintenance, and much more. In general, the optimization of production processes tries to find an optimal state in a system with respect to an objective function. There are different approaches that try to provide optimization problems with PDE constraints. On the one hand, an adjoint equation system of the discretized model can be derived, i.e., we derive a first order optimality system of the discretized version of the continuous model. Optimization issues for continuous PDEs based on adjoint approaches are investigated, for instance [99, 100]. On the other hand, the discretized model can be transformed into a linear mixed integer problem (or short MIP). Reformulation of continuous conservation laws to MIP can be found, for example, in [25, 37, 51]. However, far too little attention has been paid to optimization approaches with respect to discontinuous conservation laws.

Another application in due of production processes is the simulation of the ma-

terial flow in a more accurate way, i.e., a detailed modeling of certain sectors of a manufacturing system, e.g., the conveyor belt, machine processes of production units, and many more. Within this work, we are interested in finding a novel continuous model that describes material flow on conveyor belts. Many models in order to simulate production processes are discrete and based on ordinary differential equation systems, e.g., [90, 91, 103]. However, discrete models would be too time consuming for a large amount of parts. Therefore, a derivation of a continuous model represent a good compromise.

Parts of this work will be or have been published in the following journals and proceedings:

- S. Hoher, P. Schindler, S. Göttlich, V. Schleper, and S. Röck, *System Dynamic Models and Real-time Simulation of Complex Material Flow Systems*, In H. A. ElMaraghy, editor, Enabling Manufacturing competitiveness and economic sustainability, Part 3, pages 316-321. Springer, 2012.

- S. Göttlich, A. Klar, and P. Schindler, *Discontinuous conservation laws for production networks with finite buffers*, Discontinuous conservation laws for production networks with finite buffers, SIAM J. Appl. Math., 73(3):1117-1138, 2013.

- S. Göttlich, S. Hoher, P. Schindler, V. Schleper, and A. Verl, *Modeling, simulation and validation of material flow on conveyor belts* - accepted to applied mathematical modeling, 2013.

- S. Göttlich, and P. Schindler, *Optimal inflow control of production systems with finite buffers* - submitted, 2013.

Chapter 1

Mathematical Modeling in 1D

Manufacturing systems can be prescribed by a large number of mathematical models. These approaches help us to study and analyze the dynamic of production systems. However, they can help us to plan and optimize production processes. Therefore, we are interested in finding simulation and optimization tools. The focus of this chapter is on models, which prescribe deterministic machine failures and maintenance procedures in a production line. As a rule, these models are time-dynamic. There are various approaches to prescribe such a behavior. On the one hand, there exist models based on the computation of individual parts. These approaches are classified as microscopic models. The drawback of this approach however is an enormous computation time for a large number of parts. On the other hand, alternative approaches are fluid models based on partial differential equations (PDE). These models are characterized by aggregate quantities such as product density and material flow, see [4, 26, 28, 44, 46] and many more. The computation time of fluid models is invariant of the number of parts. Hence, this is a clear advantage in comparison to microscopic models. The use of such models is widely found in traffic flow applications [19, 61, 64, 68]. In the last years, traffic flow models based on PDEs are extended to applications in production and manufacturing systems. We refer to [3, 4, 25, 26, 44, 46, 58] for an overview.

In this chapter, we present a phenomenological model for production lines with break-downs and limited buffers. As long as the maximum buffer limit is not reached, the production in process is straightforward. If the buffer is full, however, then we observe a bottleneck situation causing congestions similar to the traffic flow problems. In the beginning of this chapter, a basic microscopic model in one dimensional is introduced to prescribe the dynamic of production processes with break-downs. The underlying fluid model of this microscopic approach is a conservation law with a discontinuous flux function. In general, problems with discontinuous flux functions are divided in two classes: discontinuities in the quantity or density, e.g. [16, 29–32, 41, 59, 81, 82, 102] or discontinuities in the space variable [1, 7, 42, 76, 96, 97] and references therein. In our case, the flux contains one discontinuity in the quantity.

The computation of approximate solutions of the discontinuous flux conservation

law requires special numerical methods. Generally, finite volume approaches yield good approximations for conservation laws. However, these approaches work only for continuous flux functions. One option is to regularize the discontinuous flux to a continuous function and use a finite volume method to solve them, for example, the regularized flux Godunov method (RFG). Nevertheless, this approach requires a high number of iterations for an accurate approximation to the original problem. Indeed, this results in high computational effort. Another option to solve efficiently discontinuous conservation laws is a wave front tracking method based on the exact solutions of Riemann problems. If we connect the ideas of the finite volume and the wave front tracking, we derive a new method for computing discontinuous flux conservation laws. This method is called discontinuous Godunov method (DFG) and it is a fast and accurate method to solve discontinuous flux conservation laws.
Another important task is an optimal control of production lines. There are several optimization approaches with PDE restrictions. In this thesis, we deal with two approaches for an optimization with restriction to a conservation law in discontinuous flux. Both approaches are based on the *first discretize then optimize* procedure. The first approach uses an adjoint system to compute efficiently a steepest descent direction for an iterative method. Adjoint equation approaches are often used for continuous optimization, see [63, 75, 99]. The other approach is a reformulation of the DFG method to a mixed integer program (MIP). The nonlinearity of this problem is transformed into linear constraints that include binary variables. Furthermore, a connection of the adjoint system within the MIP model is shown.
This chapter is structured as follows: In Section 1.1 we introduce a one dimensional microscopic model based on an ordinary differential equation system. The following section provides the concept of continuous modeling with conservation laws. Therefore, the microscopic model of Section 1.1 is used to derive a conservation law with discontinuous flux. Additionally, several cases of Riemann solutions and the appearance of zero waves are discussed. In Section 1.3, the numerical methods for discontinuous flux conservation laws, RFG, DFG, and the front tracking method, are presented. An application in optimization with an adjoint equation and MIP model approach is investigated in Section 1.4. Finally, we show the numerical results in Section 1.5.

1.1 Microscopic Model

We introduce a phenomenological model to prescribe the material flow in a manufacturing system with break-downs. Such a manufacturing system is organized as a production line consisting of machines wherein each machine is responsible for certain production steps and products are moved between machines. If such a machine is occupied or stopped, incoming goods cannot be processed and must

be stored in a buffer. As a rule, such buffers have a limited capacity. In case of a machine break-down, it could happen that a buffer reaches its maximal capacity. As a consequence, the preceding machine must be stopped sequentially. In such a situation, we observe a tailback of goods, which is similar to traffic flow.

We present a microscopic model to prescribe such a behavior. The meaning of microscopic is the individual characterization of each object in the system. The approach is based on the following assumptions:

- The dynamic of each good in a manufacturing system are prescribed individually.
- The machine break-downs are deterministic.
- The states of goods are reduced to the degree of completion (DOC) and the time.
- The processing sequence is directed by the FIFO principle (first in, first out).

Each good is assigned to an index $i \in \mathbb{Z}$. The state of degree of completion (DOC) of a good i at a time t is a function $x_i(t)$. Also we assume that the goods are ordered by the DOC state, i.e., $x_i(t) < x_{i+1}(t)$. The production process of each good i is characterized by a production velocity $v_i(t)$. Additionally, we assume that the velocity $v_i(t)$ also depends on the state of the good x_{i+1}. The dynamic of $x_i(t)$ is given by an ordinary differential equation system (ODE)

$$\frac{d}{dt}x_i(t) = v_i(t) \quad \text{for all } i, \tag{1.1}$$

where $v_i(t)$ is the actual processing velocity for a good i at the time t.

$$v_i(t) = \begin{cases} a & \text{if} \quad x_{i+1} - x_i > H_0, \\ 0 & \text{if} \quad x_{i+1} - x_i = H_0. \end{cases} \tag{1.2}$$

The velocity depends on the distance $x_{i+1} - x_i$ and a positive constant $a > 0$. If the distance is larger than the minimal distance H_0, the goods will be processed by a production velocity a. Otherwise, if the distance $x_{i+1} - x_i$ is equal to H_0, the goods will be stopped immediately.
In case of a break-down situation, a certain good i at a DOC state $x_i(t)$ cannot be processed anymore. To obtain a break-down in the model, the production velocity is set to zero, i.e., $v_i(t) = 0$. The incoming goods $i - 1$ with velocity $v_{i-1} > 0$ reduces its distance to i if the minimal distance is reached. Then the velocity v_{i-1} becomes immediately zero. Hence, the free flow state turns into a blocking state. An illustration is given in Figure 1.1.
If a machine is repaired, the velocity v_i is set to the original production velocity a. Then the blocking state turns into the free flow state.

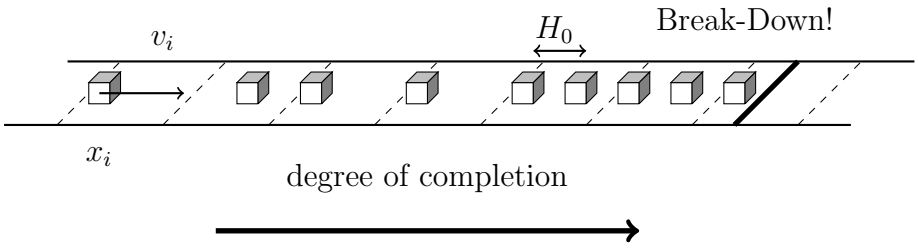


Figure 1.1: Illustration of the microscopic model. Each good i moves with a velocity v_i. The distance of both goods never deceeds H_0.

1.2 Continuous Modeling

In the following, we consider a large number of goods in a manufacturing system. Generally, the microscopic model yields a large ODE system and the computation time becomes very high. This problem can be avoided if we homogenize the quantity (goods) to a density. The result is a macroscopic or continuous approach.
Continuous models based on differential equations are nowadays widely used to describe production systems, see [3, 4, 25, 26, 44, 46, 58] for an overview.

1.2.1 The Conservation Law

There are two different approaches to describing the movement of several particles moving in a certain direction x. It can be done either by describing the movement of each part or by considering the evolution of a part density. The density of goods on space x and time t is given by $\rho(x,t) \in \mathbb{R}^+$. The amount of goods in a spatial interval $[x_1, x_2]$ is given by

$$\int_{x_1}^{x_2} \rho(x,t)dx.$$

The material flux $f(x,t)$ prescribes the amount of goods crossing each point x in one time unit. The amount of goods passing through the point x during the time interval $[t_1, t_2]$ is given by

$$\int_{t_2}^{t_1} f(x,t)dt.$$

In general, the particles in the system cannot be lost. This means, that the amount of goods in a arbitrary interval $[x_1, x_2]$ at a certain time t_2 minus the amount of goods in the same interval at an earlier time t_1 is equal to the difference

of inflowing goods at location x_1 minus outgoing goods at location x_2 during the time interval $[t_1, t_2]$, i.e.,

$$\int_{x_1}^{x_2} \rho(x, t_2)dx - \int_{x_1}^{x_2} \rho(x, t_1)dx = \int_{t_2}^{t_1} f(x_1, t)dt - \int_{t_2}^{t_1} f(x_2, t)dt. \tag{1.3}$$

If ρ and f is smooth enough, (1.3) yields

$$\int_{t_2}^{t_1} \int_{x_1}^{x_2} \partial_t \rho(x, t) + \partial_x f(x, t)dxdt = 0. \tag{1.4}$$

Because (1.4) holds for all $t_1, t_2 > 0$ and all intervals $[x_1, x_2]$, we obtain a conservation law, a hyperbolic partial differential equation (PDE):

$$\partial_t \rho(x, t) + \partial_x f(x, t) = 0. \tag{1.5}$$

However, (1.5) requires continuous differentiable functions ρ and f. This might be a strong restriction. The integral form of (1.3) holds, even when ρ and f are discontinuous.
Multiplying (1.5) by test functions $\Phi(x, t)$ and integrating with respect to the whole space and time domain yields

$$\int_0^\infty \int_{-\infty}^\infty [\partial_t \rho(x, t) + \partial_x f(x, t)]\Phi(x, t)dxdt = 0. \tag{1.6}$$

In particular, Φ has a compact support, meaning it is identically zero outside of some bounded region of the x-t-plane. If we assume that Φ is a smooth function, then we can integrate by parts in (1.6) to obtain the following weak solution formulation. For more details, we refer to [14, 79].

Definition 1.2.1 (Weak Solution)**.** *A function $\rho(x, t)$ is called a weak solution of* (1.5) *if it holds*

$$\int_0^\infty \int_{-\infty}^\infty [\rho\partial_t\Phi + f\partial_x\Phi]dxdt = -\int_\infty^\infty \rho(x, 0)\Phi(x, 0)dx \tag{1.7}$$

for all smooth functions Φ with compact support.

The equation (1.5) requires the unknown functions ρ and f. Hence, the degree of freedom can be reduced to one if f depends explicitly on ρ.

1.2.2 Connection of the Microscopic Model and the Conservation Law

In the following, we find the relation between the flux f and the density ρ in consideration of the microscopic model. Derivations of continuous models from

certain microscopic models are already investigated, e.g., [3, 5]. The following computation orientates to [3].
We assume that a function $z(x,t)$ exists such that $\rho(x,t) = -\partial_x z(x,t)$. Set $\rho(x,t) = -\partial_x z(x,t)$ and integrate (1.5) once with respect to x. This yields

$$\partial_t z(x,t) - f(x,t) = 0. \tag{1.8}$$

We construct an approximation of ρ, z, and f based on the microscopic model in Section 1.1 which fulfills (1.8) arbitrary for large number of goods. This is a motivation to get a PDE model for the microscopic model in Section 1.1.
In the following, we consider the microscopic model and introduce the total volume Y of all goods in the system. Also, Y is bounded and do not become infinity. Furthermore, N denotes the number of all goods. Additionally, we define the ratio of the total volume between the total amount of goods, i.e.,

$$\Delta y := \frac{Y}{N}$$

We define a function $Z(x,t)$ based on the solution of the ODE system (1.1). Moreover, Z is called N-curve, see [84]. The N-curve $Z(x,t)$ at a DOC state x is given by the number of goods which have passed the DOC state x at time t, i.e.,

$$Z(x,t) = \sum_{i=1}^{N} \Delta y \cdot H(x_i(t) - x),$$

where H denotes the usual Heaviside function

$$H(x) = \begin{cases} 1 & \text{if} \quad x > 0, \\ 0 & \text{if} \quad x \leq 0. \end{cases}$$

Note that z of equation (1.8) is a N-curve as well.
The flux at a DOC state x is given by the time derivative of $Z(x,t)$, i.e.,

$$F(x,t) = \frac{d}{dt} Z(x,t) = \Delta y \sum_{i=1}^{N} \delta(x_i(t) - x) \cdot v_i(t),$$

where δ terms the dirac-distribution. The density at a state x is computed by the negative spatial derivative of $Z(x,t)$, i.e.,

$$R(x,t) = -\frac{d}{dx} Z(x,t) = \Delta y \sum_{i=1}^{N} \delta(x_i(t) - x).$$

Multiplying R with arbitrary test function Φ and integration with respect to x over $\mathbb{R}$ yields

$$\int_{-\infty}^{\infty} \Phi(x) R(x,t) dx = \sum_{i=1}^{N} \Phi(x_i(t)) \Delta y.$$

We rewrite this into a Riemann sum for an integral as

$$\int_{-\infty}^{\infty} \Phi(x)R(x,t)dx = \sum_{i=1}^{N} \Phi(x_i(t))(\Delta x_i(t) \cdot \rho(x_i(t),t)),$$

where $\Delta x_i(t)$ denotes the distance of $x_{i+1}(t)$ and $x_i(t)$ and the function $\rho(x,t)$ is given at $x = x_i(t)$, i.e.,

$$\Delta x_i(t) = x_{i+1}(t) - x_i(t), \qquad \rho(x_i(t),t) = \frac{\Delta y}{x_{i+1}(t) - x_i(t)}.$$

Therefore, R approximate the function ρ arbitrary in a weak sense for a high number of goods, i.e.,

$$\int_{-\infty}^{\infty} \Phi(x)R(x,t)dx \approx \int_{-\infty}^{\infty} \Phi(x)\rho(x,t)dx.$$

By the same way, we find a function f which approximates F in a weak sense for large N, i.e.,

$$\begin{aligned}\int_{-\infty}^{\infty} \Phi(x)F(x,t)dx = \sum_{i=1}^{N} \Phi(x_i(t))\Delta y \cdot v_i(t) &= \sum_{i=1}^{N} \Phi(x_i(t))(\Delta x_i(t) \cdot f(x_i(t),t)) \\ &\approx \int_{-\infty}^{\infty} \Phi(x)f(x,t)dx,\end{aligned}$$

where $f(x,t)$ is given in $x = x_i(t)$ by

$$\begin{aligned}f(x_i(t),t) &= v_i(t) \cdot \frac{\Delta y}{x_{i+1}(t) - x_i(t)} \\ &= v_i(t) \cdot \rho(x_i(t),t).\end{aligned}$$

The velocity $v_i(t)$ which is defined in (1.2) has an alternative form

$$\begin{aligned}v_i(t) = a \cdot H(x_{i+1} - x_i - H_0) &= a \cdot H(\frac{\Delta y}{H_0} - \frac{\Delta y}{x_{i+1} - x_i}) \\ &= a \cdot H(\rho_{max} - \rho(x_i(t),t)),\end{aligned}$$

where $\rho_{max} = \frac{\Delta y}{H_0}$. Then the flux $f(x,t)$ for $x = x_i(t)$ is

$$f(x_i(t),t) = a\rho(x_i(t),t) \cdot H(\rho_{max} - \rho(x_i(t),t)). \tag{1.9}$$

At this point, the functions $\rho(x,t)$ and $f(x,t)$ is given only for $x = x_i(t)$. The next step is to define ρ, f for all $x \in \mathbb{R}$ by a suitable interpolation. Especially, f is representable as a ρ-dependent function. Furthermore, it is possible to construct an N-curve $z(x,t)$ from ρ. Finally, we show that $z(x,t)$ fulfills (1.8) arbitrary for

large N.
The density ρ is given for single points $x_i(t)$. We extend the function $\rho(x,t)$ for all x by an interpolation approach, i.e.,

$$\rho(x,t) = \rho(x_i(t),t) \quad \text{if } x_i(t) \le x < x_{i+1}(t), \text{ for all } i.$$

Also the interpolation of the density ρ preserves the total mass, i.e., the spatial integral of ρ,

$$\int_{-\infty}^{\infty} \rho(x,t)dx = \sum_{i=1}^{N} \Delta x_i(t)\rho(x_i(t),t) = \Delta y N = Y,$$

yields the total volume Y. Thus, an interpolation of the flux $f(x,t)$ is given by

$$f(x,t) = a\rho(x,t) \cdot H(\rho_{max} - \rho(x,t)).$$

By definition, the N-curve $z(x,t)$ is the negative antiderivative of $\rho(x,t)$, i.e.,

$$z(x,t) = -\int_{x_0}^{x} \rho(s,t)ds, \quad \text{or} \quad \rho(x,t) = -\partial_x z(x,t),$$

for an $x_0 < x_1(0)$. Especially, $z(x,t)$ is a continuous function. Now we evaluate the time derivative of $z(x,t)$ for an x with $x_i(t) \le x < x_{i+1}(t)$, i.e.,

$$\partial_t z(x,t) = \frac{d}{dt}\int_{x_0}^{x} -\rho(s,t)ds = \frac{d}{dt}\left(\sum_{k=1}^{i-1} -\Delta x_k(t)\rho(x_k(t),t) - \rho(x_i(t),t)(x - x_i(t))\right)$$
$$= -\frac{d}{dt}\rho(x_i(t),t)(x - x_i(t)) = \frac{\Delta y}{\Delta x_i(t)^2}(v_{i+1}(t) - v_i(t))(x - x_i(t)) + \frac{\Delta y}{\Delta x_i(t)}v_i(t).$$

Finally, this results

$$\partial_t z(x,t) = \frac{\Delta y}{\Delta x_i(t)^2}(v_{i+1}(t) - v_i(t))(x - x_i(t)) + \frac{\Delta y}{\Delta x_i(t)}v_i(t). \tag{1.10}$$

- **Case 1:** The distance of two neighboring goods is $x_{i+1} - x_i > H_0$. This yields that the goods i and $i+1$ moves with velocity a, i.e., $v_i = v_{i+1} = a$. Insert $v_i(t)$ and $v_{i+1}(t)$ in equation (1.10). In that case, $\partial_t z(x,t)$ simplifies to

$$\partial_t z(x,t) = a\frac{\Delta y}{\Delta x_i(t)} = a\rho(x,t), \quad \text{for } x_i(t) \le x < x_{i+1}(t).$$

- **Case 2:** The distance of the goods i and $i+1$ is $x_{i+1} - x_i = H_0$. If the velocity of the succeeding good $i+1$ becomes zero, i.e., $v_{i+1}(t) = 0$, then the good i stops as well, i.e., $v_i(t) = 0$. By applying $v_i(t)$, $v_{i+1}(t)$ in equation (1.10) yields

$$\partial_t z(x,t) = 0, \quad \text{for } x_i(t) \le x < x_{i+1}(t).$$

- **Case 3:** The distance of the goods i and $i+1$ is larger than H_0 and the succeeding good $i+1$ stops, i.e., $x_{i+1} - x_i > H_0$ and $v_{i+1} = 0$. Thus, $v_i(t) = a$ until $x_i(t)$ reaches the minimal distance H_0. Moreover, (1.10) yields

$$\partial_t z(x,t) = a\rho(x_i(t),t) - \frac{x - x_i(t)}{x_{i+1} - x_i(t)} \cdot a\rho(x_i(t),t), \quad \text{for } x_i(t) \le x < x_{i+1}(t).$$

In all cases $\partial_t z$ coincides with f in $x = x_i(t)$ for all i, i.e.,

$$\partial_t z(x_i(t),t) = f(x_i(t),t), \quad \forall i.$$

Especially in case 1 and 2 holds $\partial_t z(x,t) = f(x,t)$ for all x. In case 3, $\partial_t z(x,t)$ is a piecewise linear interpolation of $f(x_i,t)$ with sampling points x_i. Furthermore, $\partial_t z(x,t)$ is an approximation of $f(x,t)$ for large N. Integration over the difference $|\partial_t z(x,t) - f(x,t)|$ with respect to x yields

$$\begin{aligned}\int_{-\infty}^{\infty} |\partial_t z(x,t) - f(x,t)|dx &\overset{\text{Case 1,2}}{=} \sum_{v_i=a, v_{i+1}=0} \int_{x_i(t)}^{x_{i+1}(t)} |\partial_t z(x,t) - f(x,t)|dx \\ &= \sum_{v_i=a, v_{i+1}=0} \rho(x_i(t),t)\Delta x_i(t)^2 \le N \cdot \rho(x_i(t),t)\frac{1}{N^2} = \mathcal{O}(\frac{1}{N})\end{aligned}$$

Moreover, (1.8) is arbitrary fulfilled in a weak sense for $N \to \infty$, i.e.,

$$\partial_t z(x,t) - f(x,t) = 0.$$

Also $f(x,t)$ is a function which depends on the density $\rho(x,t)$ and is representable as

$$f(\rho) = a\rho H(\rho_{max} - \rho).$$

1.2.3 The Flow Model

The models of interest rely on conservation laws with discontinuous flux functions representing production units with finite buffers. The evolution of the part density $\rho(x,t) \in [0, \rho_{max}]$ satisfies for all $x \in [0,1]$ the equation

$$\partial_t \rho + \partial_x f(\rho) = 0, \quad \rho(x,0) = \rho_0(x), \tag{1.11}$$

where the relation between flux and density is given by

$$f(\rho) = H(\rho_{max} - \rho)\tilde{f}(\rho). \tag{1.12}$$

Since $H(\cdot)$ denotes the Heaviside function and $\tilde{f}(\rho)$ is a smooth concave function. In the following the solution of (1.11) with discontinuous flux (1.12) is defined as a limit process of weak solutions of a regularized model. A regularized model is a modification of (1.11) with a continuous flux, which approximates the discontinuous flux (1.12).

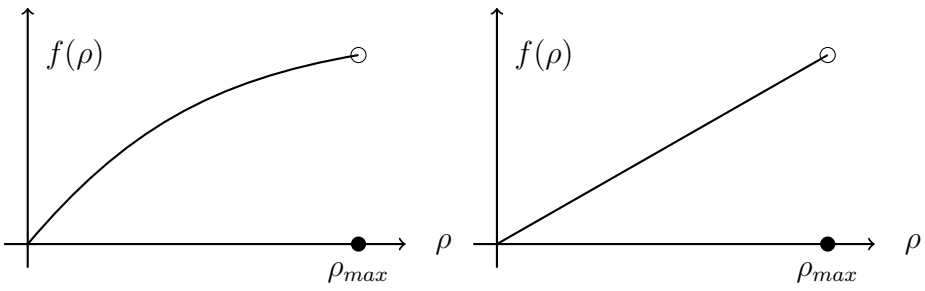


Figure 1.2: An example for the flux function in (1.12) (left picture) and the linearized flux function (1.13) (right picture).

Definition 1.2.2 (Regularized Flux)**.** *A flux f_δ is a regularized flux of f if it holds*

- *$f_\delta(\rho)$ is continuous for all $\rho \in [0, \rho_{max}]$.*
- *$f_\delta(\rho_{max}) = 0$.*
- *f_δ is concave, i.e., $f_\delta(\lambda\rho_1 + (1-\lambda)\rho_2) \geq \lambda f_\delta(\rho_1) + (1-\lambda) f_\delta(\rho_2)$ for all $\lambda \in [0,1]$, $\rho_1, \rho_2 \in [0, \rho_{max}]$.*
- *f_δ is an arbitrary approximation of f, i.e.,*

$$\int_0^{\rho_{max}} |f_\delta(\rho) - f(\rho)| d\rho = \mathcal{O}(\delta), \quad \delta \to 0.$$

Definition 1.2.3 (Weak Solution)**.** *A function $\rho(x,t)$ is a weak solution of* (1.11) *with discontinuous flux* (1.12) *if it holds*

$$\lim_{\delta \to 0} \int_0^\infty \int_{-\infty}^\infty |\rho_\delta(x,t) - \rho(x,t)| dx dt = 0,$$

where ρ_δ is a weak solution of the continuity equation with a regularized flux f_δ, i.e.,

$$\int_0^\infty \int_{-\infty}^\infty [\rho_\delta \partial_t \Phi + f_\delta \partial_x \Phi] dx dt = -\int_\infty^\infty \rho_0(x) \Phi(x,0) dx$$

for all smooth functions Φ with compact support.

For simplicity, we restrict this model to a linear ramp-up situation and describe the procedure for

$$f(\rho) = a\rho H(\rho_{max} - \rho) \tag{1.13}$$

with a constant velocity $a > 0$, see Figure 1.2 (right picture).
In our setting, we can use a simpler approach. The solutions of (1.13) are defined as the limit solutions of a regularized problem, which is defined as follows. The flux $f(\rho) = a\rho H(\rho_{max} - \rho)$ is approximated by the continuous function f_δ for $\delta > 0$ with

$$f_\delta(\rho) = \min\{a\rho, \frac{1}{\delta}(\rho_{max} - \rho)\} \quad \text{for } \delta > 0. \tag{1.14}$$

The flux function (1.14) is shown in Figure 1.3.

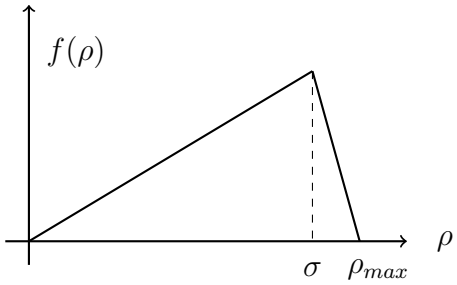


Figure 1.3: Regularized flux function (1.14).

In the limit $\delta \to 0$, we get the convergence towards the desired flux function (1.13). In accordance with the traffic flow literature, we remark that

$$\sigma := \underset{\rho\in[0,\rho_{max}]}{\arg\max} f_\delta(\rho) = \frac{\rho_{max}}{1 + a\delta}. \tag{1.15}$$

Note that, the above mentioned model with the flux function (1.14) is well investigated and intensively discussed in [27]. This knowledge will be used in Chapter 2.1 used to extend the model (1.13) to general networks. But we before we do so, let us discuss more details of equation (1.12) and (1.13) .

1.2.4 Riemann Problems

The Riemann problem for equation (1.13) is given by the initial data

$$\rho(x, 0) = \begin{cases} \rho_l & \text{if} \quad x < 0, \\ \rho_r & \text{if} \quad x > 0. \end{cases} \tag{1.16}$$

We note that the Riemann problem has been solved in more generality and explained in detail in [102], where the case of a general nonlinear flux function with discontinuity lies at any point in the domain not just at the right boundary. As in [4, 102] the following cases are distinguished.

(A.) $0 \leq \rho_l, \rho_r < \rho_{max}$: The solution to the Riemann problem is a shock wave with speed $s = a$. This case is classical.

(B.) $0 \leq \rho_l < \rho_r = \rho_{max}$: Looking at the regularized problem as $\delta \to 0$, the solution to the Riemann problem is a shock wave traveling with speed $s = \frac{f(\rho_l)}{\rho_l - \rho_{max}}$.

(C.) $0 \leq \rho_r < \rho_l = \rho_{max}$: By applying the Rankine Hugoniot condition $s = \frac{f(\rho_r) - f(\rho_l)}{\rho_r - \rho_l}$ with $f(\rho_{max}) = 0$ directly to the discontinuous conservation law, one would get negative valued velocities for s. Considering the regularization f_δ of $a\rho \cdot H(\rho_{max} - \rho)$ one obtains a solution consisting of two dispersing shock waves with intermediate state σ. The speed of the two waves is $s = -\frac{1}{\delta}$ and $s = a$. For small δ the solution of the conservation law with f_δ approximates the classical shock wave solution to the Riemann problem with speed $s = a$.

Zero Waves

We note that, in particular in case (B.) and (C.), these Riemann problems do not describe the dynamical picture in all situations. In certain situations solutions of Riemann problems cannot be considered separately. One has to investigate so called double Riemann problems, see [41, 102]. Consider, for example, a situation with initially 2 discontinuities; as in case (B.) and in (C.), compare [102].

$$\rho(x,0) = \begin{cases} \rho_l & \text{if} \quad x < 0, \\ \rho_{max} & \text{if} \quad 0 < x < 1, \\ \rho_r & \text{if} \quad x > 1, \end{cases}$$

with $\rho_l, \rho_r < \rho_{max}$. In this case the propagation of the discontinuities is not described by a separate analysis of the Riemann problems. Considering the above Riemann problems separately one obtains, that the left discontinuity is traveling as a shock wave with negative speed $s = \frac{f(\rho_l)}{\rho_l - \rho_{max}}$ (case (B.)) and the right discontinuity is traveling with speed $s = a$ (case (C.)) in the final state.
However, this is not the limit solution of the regularized problem with initial values (1.16) as $\delta \to 0$. For fixed δ, this solution is a backward going shock wave for the left discontinuity and a combination of two shock waves for the right discontinuity, where one of them is propagating with speed $s = a$ to the right, the other one with speed $s = -1/\delta$ to the left. However, this picture is correct, only as long as the two waves do not interact. Once they interact, (as in once the shock wave with speed $s = -1/\delta$ arrives at the left discontinuity), we are in a situation like in case (A.), since the density there is reduced below ρ_{max}. This means, from now on the regularized solution moves to the right with speed a.

Since for $\delta \to 0$ the speed of propagation of the backwards going wave starting at the right discontinuity is infinity, the solution at the left discontinuity behaves immediately as in case (A.) and propagates with speed $s = a$. That means, the limit solution of the regularized problem as $\delta \to 0$ is propagating without changing its shape with speed $s = a$. See Figure 1.4 for the time evolution and compare the corresponding figure in [102].

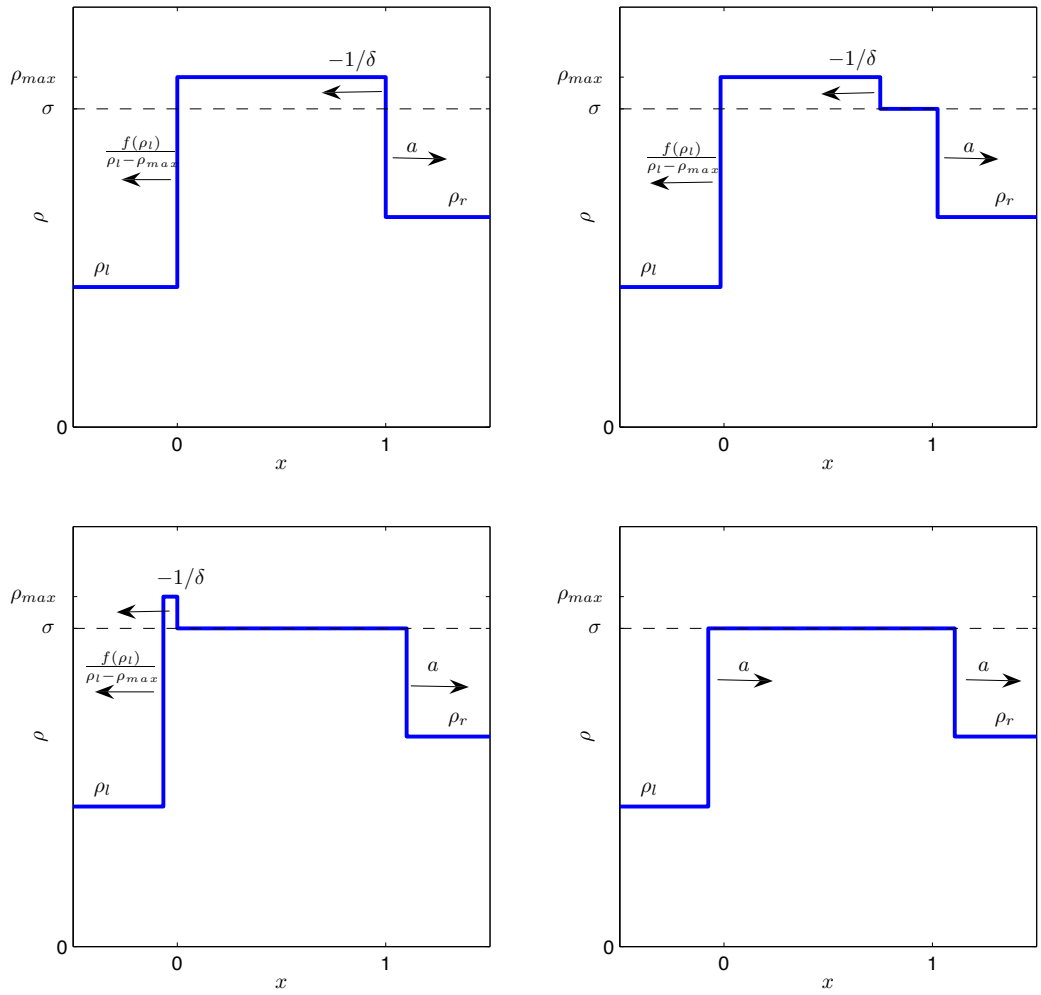


Figure 1.4: Double Riemann problem and zero shock waves.

Boundary Conditions

Boundary conditions have also to be discussed carefully. Considering a boundary value problem with a situation as in case (B.) with a boundary at $x_B > 0$. One can prescribe the outgoing flux at this boundary. If such a predefined outflow

f_{out} ranges from 0 to $a\rho_{max}$, by mass conservation, the shock wave in case (B.) is moving with the speed $s = \frac{f(\rho_l)-f_{out}}{\rho_l-\rho_{max}}$ ranging from $s = \frac{f(\rho_l)}{\rho_l-\rho_{max}}$ to $s = a$.

1.3 Numerical Methods

In this section we introduce three numerical methods for solving (1.11) with discontinuous flux (1.12). At first, we present the regularized Flux Godunov method. The main components of this method are the regularization of the discontinuous flux and a numerical solver based on the Godunov method. For the second method, we briefly review the weak solutions of (1.13) and discuss the corresponding wave-front tracking algorithm. The latter will be used to set up the numerical framework for the discontinuous flux Godunov (DFG) method recently introduced in [48]. The DFG method is a finite volume approach supplemented with a problem-adapted numerical flux allowing for a sharp tracking of shocks. Therefore this formulation will be essential for the numerical consideration of our optimal control problem introduced in 1.4 and 1.4.3.

1.3.1 Regularized Flux Godunov

A conventional way for solving (1.11) is the regularization of the discontinuous flux (1.12) and the use of classical schemes for hyperbolic conservation laws, see [4]. Therein, the flux discontinuity is connected with a linear ramp-down of slope $-1/\delta$, cf. Figure 1.3. The regularized flux of (1.12) is defined as

$$f_\delta(\rho) = \min\{a\rho, \frac{1}{\delta}(\rho_{max} - \rho)\} \qquad \text{for } \delta > 0.$$

Obviously, f_δ fulfills the assumptions of Definition 1.2.2, however, f_δ approximates f arbitrary for small δ. Now it is possible to solve the regularized conservation law with conventional methods, e.g. Lax-Friedrichs, Godunov. In this thesis, the regularized flux conservation law is solved by the Godunov method. A detailed description of the Godunov method can be found in [79].

The spatial domain is discretized to a equidistant grid

$$0 = x_0 < x_1 < x_2 < x_{N-1} < x_N = 1.$$

Moreover, the spatial step size is defined as $\Delta x := x_i - x_{i-1}$. Analogously, discretize the time to a grid

$$0 = t_0 < t_1 < t_2 < t_3 < ...$$

with step size $\Delta t := t_n - t_{n-1}$. An approximation of the density $\rho(x,t)$ is given as a set of discrete cells, i.e., $\rho(x,t_n) = \rho_i^n$ for $x \in [x_{i-1}, x_i]$.

One option to apply boundary conditions is to extend the computational domain with additional cells, called ghost cells. We define additional cells on the left and right boundary, i.e., $[x_{-1}, x_0]$, $[x_N, x_{N+1}]$ are ghost cells with values ρ_0^n, ρ_N^n. Let f_{in} be the inflow profile. Then the left ghost cell $[x_1, x_0]$ is set to the value $\rho_0^n = f_{in}/a$.
The right boundary condition is defined as follows. We set the right ghost cell to value $\rho_{N+1}^n = \rho_N^n$ for a free flow boundary. Alternatively or additionally, the right ghost cell can set to value $\rho_{N+1} = \rho_{max}$ for simulation of a zero flux condition.

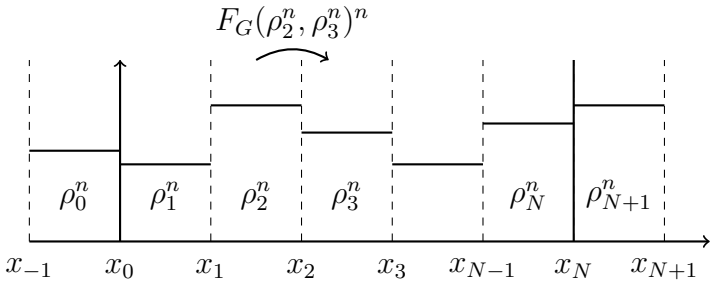


Figure 1.5: Illustration of the finite volume method.

For numerical computations, we consider the explicit Godunov scheme in conservative form,

$$\rho_i^{n+1} = \rho_i^n - \frac{\Delta t}{\Delta x}(F_G(\rho_i^n, \rho_{i+1}^n) - F_G(\rho_{i-1}^n, \rho_i^n)), \quad i = 1, ..., N, \tag{1.17}$$

$$F_G(\rho_i^n, \rho_{i+1}^n) = \begin{cases} \min_{w \in [\rho_i^n, \rho_{i+1}^n)} f_\delta(w) & \text{if} \quad \rho_i^n \leq \rho_{i+1}^n, \\ \max_{w \in [\rho_{i+1}^n, \rho_i^n)} f_\delta(w) & \text{if} \quad \rho_i^n \geq \rho_{i+1}^n. \end{cases} \tag{1.18}$$

Additionally it is sufficient to hold the CFL condition which depends explicitly of δ. In more detail, it yields:

$$\Delta t \leq \delta \Delta x \qquad \text{for sufficient small } \delta \tag{1.19}$$

As already discussed in Section 1.2.4, this regularization also implies fast back traveling shock waves in the solution. In fact, there are two types of fast traveling waves for the regularized version. The zero waves described in Subsection 1.2.4 and the backward traveling waves in the description of the Riemann problem (Case (B.)), if ρ_l is near to ρ_{max}. Due to the CFL condition, which is $\Delta t \leq |\delta| \Delta x$ in the case of zero waves, explicit solvers are forced to use very small time-steps for a reasonable resolution and are thus computationally expensive.
Obviously, there is a need for an alternative solution scheme that computes the discontinuous flux function in an efficient way without any regularization.

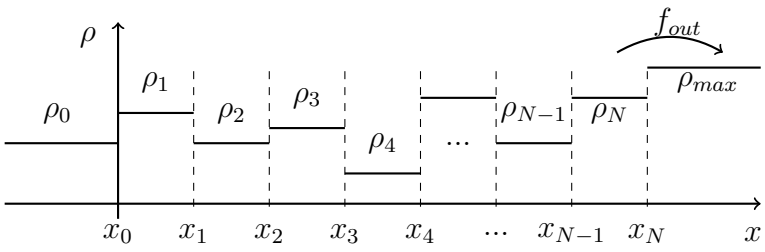


Figure 1.6: Choice of the initial data for a multiple Riemann problem.

1.3.2 Wave Front Tracking Algorithm

We compute approximate solutions of (1.11) with discontinuous flux function (1.12) by the wave front tracking approach; see [14, 41, 42, 69, 76]. The primary idea is to approximate the initial data by step functions, i.e., piecewise constant functions. This yields a multiple Riemann problem. Single Riemann problems are considered and solved in 1.2.4. The result is one ore more traveling wave fronts, which can collide. All wave interactions leads to new Riemann problems. At first, we construct a multiple Riemann problem. Discretize the spatial domain in a grid $0 = x_0 < x_1 < x_2 < x_{N-1} < x_N = 1$ and define the step size $\Delta x := x_i - x_{i-1}$. The problem can be formulated as a multiple Riemann problem:

$$\rho_0(x) = \sum_{i=1}^{N} \rho_i \chi_i(x), \quad \chi_i(x) = \begin{cases} 1 & \text{if} \quad x \in (x_{i-1}, x_i), \\ 0 & \text{otherwise.} \end{cases} \tag{1.20}$$

The initial data at the boundaries is set as follows (see also Figure 1.6):

$$\begin{aligned} x < x_0 : &\quad \rho_0(x) = \rho_0, \quad f(\rho_0) = a\rho_0, \\ x > x_N : &\quad \rho_0(x) = \rho_{max}, \quad f(\rho_{max}) = f_{out}. \end{aligned}$$

Then, the wave front-tracking algorithm works as follows. We choose a starting time defined as $[t_N] = 0$, i.e., no interaction between two waves has happened so far. Since the initial data is piecewise constant given by the multiple Riemann problem (1.20), different wave fronts will evolve over time. The key idea of the wave front-tracking algorithm is to track each wave propagation individually. Next, we describe the tracking procedure and present in Figures 1.7 and 1.8 some useful illustrations.
According to the cases (A.), (C.), the shock front at position x_i moves with positive velocity, if there is no interaction between any backward traveling shock waves. Generally, the positive shock velocity is computed via

$$s_{i+1}^{+} := \frac{f(\rho_i) - f(\rho_{i+1})}{\rho_i - \rho_{i+1}},$$

cf. case (A.) in Section 1.3.2. Additionally, if $f_{out} > f(\rho_N)$, the shock moves in positive direction. Thus, there it exists no backward traveling shock wave. If $f_{out} < f(\rho_N)$, a shock with negative speed appears in the solution. This shock starts at location x_N and moves with velocity s_N^-:

$$s_N^- := \frac{f_{out} - f(\rho_N)}{\rho_{max} - \rho_N}.$$

If the negative shock wave interacts with the shock wave of velocity s_N^+, we get a new single Riemann problem with states $\rho_l = \rho_{N-1}$, $\rho_r = \rho_{max}$, $f(\rho_r) = f_{out}$. Therefore a new shock wave appears. Generally, the shock velocity is determined by

$$s_i^- := \frac{f_{out} - f(\rho_i)}{\rho_{max} - \rho_i}, \quad i = 1, \dots, N.$$

Note that the shock with velocity s_i^- moves slower than s_i^+, i.e., $s_i^- \leq s_i^+$.
We define the time $[t_{i-1}]$ for the interaction of wave s_i^- with wave s_i^+. The slower traveling shock changes its velocity to s_{i-1}^-. For an illustration, see Figure 1.7.

Remark 1.3.1. *In the case of a zero wave, i.e., $\rho_i = \rho_{max}$, the shock speed is $s = -\infty$. This yields that $[t_{i-1}] = [t_i]$.*

For evaluating the time of wave interaction $[t_{i-1}]$ it is necessary to have knowledge about the previous waves. Therefore we assume that $[t_i]$ is known. Then, the intersection of characteristic curves is

$$s_i^+([t_{i-1}] - [t_i]) = \Delta x + s_i^-([t_{i-1}] - [t_i])$$

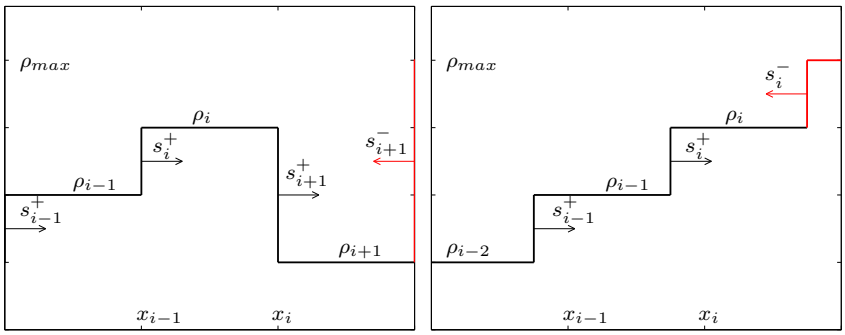
which in turn recursively leads to

$$[t_{i-1}] = \frac{\Delta x}{a - s_i^-} + [t_i] = \frac{\Delta x(\rho_{max} - \rho_i)}{a\rho_{max} - f_{out}} + [t_i] = \sum_{k=i}^{N} \frac{\Delta x(\rho_{max} - \rho_k)}{a\rho_{max} - f_{out}}. \tag{1.21}$$

In the next section we explain how the information induced by the wave front-tracking algorithm can be used to derive a suitable and efficient numerical scheme.

1.3.3 Discontinuous Flux Godunov

Numerical schemes which are able to deal with discontinuous flux functions have been developed for example in [81, 102] and [82]. In [82] an implicit method based on the analysis in [16] has been developed.
The scheme described in [102] is more closely related to the presented approach. This scheme is able to treat the general case with discontinuities located anywhere, not only on the right boundary of the density domain. It is able to deal



(a) Time $[t_{i+1}]$ (b) Time $[t_i]$

(c) Time $[t_{i-1}]$ (d) Time $[t_{i-2}]$

Figure 1.7: Solution of a multiple Riemann problem for several times. Note that the location of the interacting shock wave at time $[t_{i-1}]$ is computed by $s_i^+[t_{i-1}] + x_{i-1}$.

with the zero waves by including the solution of the above mentioned double Riemann problems. Thus, this algorithm avoids the stability problems due to the zero waves. However, the waves in Case (B.) might still require a strong restriction on the time step. In the present case, a related, but much simpler algorithm can be set up due to the much simpler situation compared to [102]; and, in particular, due to the fact that the discontinuity is at the right end point. The algorithm is based on the fact that for the present situation, except for Riemann problems as in Case (B.), the propagation is always given by the linear flux function. In Case (B.), the evolution of the density is given by mass conservation and the propagation of the discontinuity is determined recursively.

Based on our theoretical considerations, we introduce a finite volume scheme to solve (1.13) This scheme was presented for the first time in [48]. Applying the idea of front-tracking combined with the finite volume approach will lead to a scheme called discontinuous flux Godunov (DFG).

As before, the spatial domain is divided into N cells $[x_{i-1}, x_i]$ with constant width Δx. The solution is assumed to be piecewise constant on each grid cell. The cell-

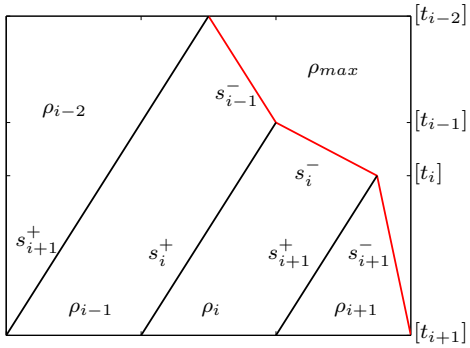


Figure 1.8: Time evolution of multiple Riemann solutions in the (x,t)-plane, cf. the detailed description in Figure 1.7.

averaged solution is given by ρ_i^n where n denotes the time index. The update ρ_i^{n+1} is the cell average of the solution of the multiple Riemann problem evaluated at interfaces between adjacent cells.
The following three-stage-algorithm, referred to as the *Reconstruct-Evolve-Average* or *REA algorithm* as in [79], is used to compute the time evolution for ρ_i^{n+1}:

- *Reconstruct* a piecewise constant function $\rho(x,t_n) = \rho_i^n$ for all $x \in [x_{i-1}, x_i]$ from the cell average ρ_i^n at time t_n.
- *Evolve* the conservation law exactly using initial data $\rho(x,t_n)$, thereby obtaining $\rho(x,t_{n+1})$ at time $t_{n+1} = t_n + \Delta t$.
- *Average* the solution $\rho(x,t_{n+1})$ to determine new cell average values

$$\rho_i^{n+1} = \frac{1}{\Delta x} \int_{x_{i-1}}^{x_i} \rho(x,t_{n+1}) dx.$$

Here, the evolution step can be performed by solving local Riemann problems at each cell interface setting $\rho_l = \rho_{i-1}^n$ and $\rho_l = \rho_i^n$. We solve Riemann solutions of regularized flux functions according to (1.3) and consider the limit case $\delta \to 0$. Similar approaches are also used for conservation laws with discontinuous flux in [41, 81, 102]. Especially the work of [102] uses *REA algorithms* for the construction of finite volume methods. The main advantage of this approach is that zero waves with infinite speed can be reproduced without any hard restriction of the CFL condition, i.e., $\Delta t \leq a \Delta x$ instead of $\Delta t |\delta| \Delta x$. Since three different cases (A.)-(C.) may arise as Riemann solutions, the front-tracking approach can be used to perform the evolution step:

(1.) Solve each local Riemann-Problem and determine the shock velocity s.

(2.) Detect shock interactions with different velocities. Once an interaction occurs, a new Riemann Problem emerges. Repeat step (1.) until the final time horizon has been reached. Note that shocks can appear with infinite velocity. The interaction of an infinite speed shock happens immediately.

After the evolution step, the front-tracking solution is averaged to obtain the cell average values ρ_i^{n+1}.
Let $F(x,t)$ be the numerical flux $f(\rho(x,t))$ at position x and time t. Then, integration of (1.11) over the domain $[t_n, t_n + \Delta t] \times [x_{i-1}, x_i]$ yields

$$\rho_i^{n+1} = \rho_i^n - \lambda F_i^n + \lambda F_{i-1}^n, \quad F_i^n := \frac{1}{\Delta t} \int_{t_n}^{t_n+\Delta_t} F(x_i, t) dt$$

where $\lambda = \frac{\Delta t}{\Delta x}$. Let us explain the choice of F_i^n. At first we consider the flux at point x_N. If $a\rho_N^n < f_{out}$, then the shock moves in positive direction, i.e., $F(x_N, t) = a\rho_N^n$. Otherwise, the shock moves with the negative speed s_N^-. Thus the flux becomes $F(x_N, t) = f_{out}$. In particular, it holds $F(x_N, t) = \min\{a\rho_N^n, f_{out}\}$.
Next, we consider the flux $F(x_i, t)$ at point x_i. The front-tracking approach provides different opportunities. In fact, two configurations are possible:

- The backward traveling shock wave s_i^- never reaches the point x_i at time $t \in (t_n, t_n + \Delta t)$. Then, the solution is just a shock wave between ρ_i^n, ρ_{i+1}^n with positive speed $s_{i+1}^+ = a > 0$ and the flux at x_i is $F(x_i, t) = a\rho_i$:

$$F_i^n := \frac{1}{\Delta t} \int_{t_n}^{t_n+\Delta_t} F(x_i, t) dt = a\rho_i^n.$$

- The backward traveling shock wave passes the location x_i at time $[\hat{t}_i] \in (t_n, t_n + \Delta t)$. Analogous to the previous situation, the flux at x_i for time $t \in (t_n, [\hat{t}_i])$ becomes $F(x_i, t) = a\rho_i$. For $t > \hat{t}_i$, the density is $\rho(x_i, t) = \rho_{max}$ with flux $F(x_i, t) = f_{out}$:

$$F(x_i, t) = \begin{cases} a\rho_i & \text{if} \quad t < [\hat{t}_i], \\ f_{out} & \text{if} \quad t > [\hat{t}_i], \end{cases}$$

  where the time $[\hat{t}_i]$ can be computed as:

$$[\hat{t}_i] = [t_i] - \frac{a[t_i]}{s_i^-} + t_n.$$

This leads to the following numerical flux:

$$\begin{aligned} F_i^n &= \frac{1}{\Delta t}\int_{t_n}^{t_n+\Delta_t} F(x_i,t)dt = \frac{1}{\Delta t}\Big[a\rho_i^n([\hat{t}_i]-t_n) + f_{out}(t_n+\Delta t-[\hat{t}_i])\Big] \\ &= \frac{1}{\Delta t}\Big[(a\rho_i^n - f_{out})[t_i] + (\rho_{max}-\rho_i^n)a[t_i]\Big] + f_{out} \\ &= \frac{1}{\Delta t}(a\rho_{max}-f_{out})[t_i] + f_{out} \overset{(1.21)}{=} \sum_{k=i+1}^{N}\frac{\Delta x}{\Delta t}(\rho_{max}-\rho_k^n) + f^{out} \\ &= \frac{\rho_{max}-\rho_{i+1}^n}{\lambda} + \sum_{k=i+2}^{N}\frac{1}{\lambda}(\rho_{max}-\rho_k^n) + f^{out} = \frac{\rho_{max}-\rho_{i+1}^n}{\lambda} + F_{i+1}^n. \end{aligned}$$

The backward traveling shock wave passes x_i if and if only the wave velocity s_i^- is negative. This is valid for $a\rho_i^n > f_{out}$. Hence,

$$\int_{t_n}^{t_n+\Delta_t} a\rho_i^n dt > \int_{t_n}^{[\hat{t}_i]} a\rho_i^n dt + \int_{[\hat{t}_i]}^{t_n+\Delta_t} f_{out}\,dt \quad \text{with} \quad a\rho_i^n > \frac{\rho_{max}-\rho_{i+1}^n}{\lambda} + F_{i+1}^n.$$

Summarizing, the discontinuous flux Godunov method (DFG) is defined as:

$$\text{(PDE):} \qquad \rho_i^{n+1} = \rho_i^n - \lambda[F_i^n - F_{i-1}^n], \qquad i = 1,\dots,N$$

supplemented with the numerical flux:

$$\text{(FLUX):} \qquad F_{i-1}^n = \min\{a\rho_{i-1}^n, \frac{\rho_{max}-\rho_i^n}{\lambda} + F_i^n\}, \qquad i = 2,\dots,N.$$

The solution is also given by the initial data and the boundary values. For the inflow, we choose the left boundary value of the density $\rho(0,t)$.

$$\text{(INFLOW):} \qquad F_0^n = a\rho_0^n \qquad \forall n = 1,\dots,N_T-1.$$

It is also necessary to prescribe outflow boundary conditions. We introduce a variable f_{out}^n that limits the outflow in a following way:

$$\text{(OUTFLOW):} \qquad F_N^n = \min\{a\rho_N^n, f_{out}^n\} \qquad \forall n = 1,\dots,N_T-1.$$

The DFG-method satisfies the following numerical properties:

Lemma 1.3.2 (Monotonicity)**.** *Let the CFL condition hold, i.e.,* $(1-a\lambda) \geq 0$. *Then, the discontinuous flux Godunov (DFG) method is a monotone numerical scheme with respect to* ρ_i^n *for all* i*, i.e., increasing the value of any* ρ_i^n *leads to non-decreasing values* ρ_i^{n+1}.

Proof. We consider the following case distinction for the numerical flux F_{i-1}^n :
Case 1: Assume that the flux satisfies $F_{i-1}^n = \frac{\rho_{max}-\rho_i^n}{\lambda} + F_i^n$. Then, the DFG-scheme simplifies to

$$\rho_i^{n+1} = \rho_i^n - \lambda F_i^n + \lambda \frac{\rho_{max} - \rho_i^n}{\lambda} + \lambda F_i^n = \rho_{max}$$

and ρ_i^{n+1} is constant for all ρ_i^n.
Case 2: Assume that the flux satisfies $F_{i-1}^n = a\rho_{i-1}^n$. We also assume that there exits a constant $K \geq 0$ such that

$$F_k^n = \frac{\rho_{max} - \rho_{k+1}^n}{\lambda} + F_{i+1}^n, \quad k < i + K, \tag{1.22}$$

$$F_k^n = a\rho_k^n, \quad k = i + K. \tag{1.23}$$

Note if $K = 0$, then $F_{i-1}^n = a\rho_{i-1}^n$, $F_i^n = a\rho_i^n$. We obtain the recursion

$$\begin{aligned}\rho_i^{n+1} &= \rho_i^n - \lambda\left(a\rho_{i+K}^n + \sum_{k=1}^{K} \frac{\rho_{max} - \rho_{i+k}^n}{\lambda}\right) + \lambda a\rho_{i-1}^n \\ &= \sum_{k=0}^{K-1} \rho_{i+k}^n - K\rho_{max} + (1 - a\lambda)\rho_{i+K}^n + a\lambda\rho_{i-1}^n.\end{aligned}$$

Due to the CFL condition and (1.22)+ (1.23), ρ_i^{n+1} is non-decreasing for all ρ_i^n. Thus, the DFG-method is monotone. □

We have shown that the DFG-method belongs to the class of monotone methods. These methods also imply the following properties:

1. The scheme is monotonicity preserving, i.e., $\rho_i^n \leq \bar{\rho}_i^n$ for all i implies $\rho_i^{n+1} \leq \bar{\rho}_i^{n+1}$ for all i,n.

2. L_1-contraction:

$$\|\rho^{n+1} - \bar{\rho}^{n+1}\|_{L_1} \leq \|\rho^n - \bar{\rho}^n\|_{L_1}.$$

3. Total Variation Diminishing (TVD):

$$\|\rho^{n+1}\|_{BV} \leq \|\rho^n\|_{BV}.$$

We refer to [79] for more details.

1.4 Optimization

Mathematical models with optimization issues play an important role for many applications. In consideration of manufacturing systems, it is useful to minimize incurred costs, reduce machine capacity utilization, or fulfill demands. In Subsection 1.4.1, we consider an optimal distribution of material flow for known dates of maintenance. Therefore, two approaches are presented. The first approach is based an adjoint equations. The other approach is an reformulation of the DFG method to a Mixed Integer Program (MIP). Furthermore, we show a connection between both approaches. In Subsection 1.4.6, we extend the MIP model to find the optimal date of a maintenance.

1.4.1 The Inflow Control Problem

The Inflow control problem consists of a minimization of inflow such that the constraints given by the discontinuous conservation law is ensured. In other words: We assume that for a predefined outflow the optimal inflow into the production system is determined such that congestions are avoided. It is up to optimization to find the optimal time-dependent inflow $\vec{u}^n = \vec{F_0}^n$ regarding the fact that a certain supply S, i.e.,

$$\text{(SUPPLY):} \qquad \Delta t \sum_{n=1}^{N_T} F_0^n = S,$$

with box constraints

$$\text{(BOX):} \qquad F_0^n \leq W \quad \forall n,$$

must be fulfilled. Mathematically, we solve this problem using a *first discretize-then optimize* approach, i.e., we directly apply the numerical discretization to the optimal control problem. We prefer the discrete optimization approach due to the almost linear nature of the problem. Similar to [51], one can show that there exits a direct connection between adjoint variables and dual variables in this setting.

$$\begin{aligned} &\min_{\vec{u}^n} J(\rho_i^n) = \sum_{n=1}^{N_T} \sum_{i=1}^{N} C_i^n \rho_i^n \\ &\text{subject to} \\ &\text{(PDE), (FLUX), (OUTFLOW), (SUPPLY), (BOX),} \end{aligned} \tag{1.24}$$

where C_i^n are positive weights and $\vec{u}^n = \vec{F_0}^n$ the controls. In the following sections, we formally derive discrete adjoint equations and a mixed-integer program as well to solve (1.24). We also focus on the equivalence of both approaches.

1.4.2 Optimality system

We formally derive the first order optimality system of the discrete problem. Therefore, we transform the pde-restricted problem (1.24) into an unrestricted one. We denote by ϕ_i^n the Lagrange multiplier for the discretized partial differential equation and ψ for the inflow condition. Then, the discrete Lagrangian function reads:

$$\begin{aligned} L(\vec{\rho_i}^n, \vec{u}^n, \vec{\phi_i}^n, \psi) &= \sum_{n=1}^{N_T}\sum_{i=1}^{N} C_i^n \rho_i^n + \sum_{n=1}^{N_T-1}\sum_{i=1}^{N} \phi_i^n \Big(\frac{\rho_i^{n+1}-\rho_i^n}{\Delta t} + \frac{F_i^n - F_{i-1}^n}{\Delta x}\Big) \\ &+ \psi\Big(\sum_{n=1}^{N_T-1} F_0^n - S\Delta t^{-1}\Big). \end{aligned}$$

We formally deduce the first order optimality system from (1.25) by assuming sufficient regularity conditions.

- The state equations (forward) equations result from the derivatives with respect to the Lagrange multipliers ϕ_i^n and ψ, i.e., they are immediately given by the constraints (PDE), (FLUX), (OUTFLOW) and (SUPPLY).

- The adjoint (or backward) equations are

$$\phi_i^{n-1} = C_i^n + \phi_i^n - \lambda \sum_{j=1}^{N} (\partial_{\rho_i^n} F_j^n - \partial_{\rho_i^n} F_{j-1}^n)\phi_j^n, \tag{1.25}$$

  for $i = 1, \ldots, N$ and $n = 2, \ldots, N_T - 1$ where the initial values obey

$$\phi_i^{N_T-1} = C_i^{N_T}, \quad i = 1, \ldots, N. \tag{1.26}$$

  Obviously, we need the derivatives $\partial_{\rho_i^n} F_j^n$ of the non-smooth numerical flux function in (1.25). Therefore, it is necessary to smooth the min $-$expression by a smooth approximation $\min_\epsilon \approx \min$. We choose

$$\min_\epsilon(\alpha, \beta) := \begin{cases} \alpha & \text{if} \quad \alpha \le \beta, \\ \frac{-\epsilon^2}{\alpha-\beta+\epsilon} + \beta + \epsilon & \text{if} \quad \alpha > \beta. \end{cases} \tag{1.27}$$

  with $\alpha = a\rho_{i-1}^n$ and $\beta = \frac{\rho_{max}-\rho_i^n}{\lambda} + F_i^n$. Then, the derivatives of the numerical flux can be represented as

$$\begin{aligned} \partial_{\rho_i^n} F_j^n &= \partial_1 \min_\epsilon\{a\rho_j^n, \frac{\rho_{max}-\rho_{j+1}^n}{\lambda} + F_{j+1}^n\} \cdot a\delta_{i,j} \\ &+ \partial_2 \min_\epsilon\{a\rho_j^n, \frac{\rho_{max}-\rho_{j+1}^n}{\lambda} + F_{j+1}^n\} \cdot (-\delta_{i,j+1}\frac{1}{\lambda} + \partial_{\rho_i^n} F_{j+1}^n) \end{aligned} \tag{1.28}$$

for $i = 1, \ldots, N$ where $\delta_{i,j}$ denotes the *Kronecker delta*, i.e., $\delta_{i,j} = 1, i = j$ or $\delta_{i,j} = 0, i \neq j$. Note that the flux F_j^n does not depend on the density ρ_i^n for $j > i$. The derivatives are then

$$\partial_{\rho_i^n} F_j^n = 0, \quad j > i.$$

The outflow F_N^n only depends on the density values ρ_N^n. Hence,

$$\partial_{\rho_i^n} F_N^n = \partial_1 \min_\epsilon \{a\rho_N^n, f_{out}^n\} \cdot a\delta_{i,N}.$$

- Considering the gradient equation, we end up with:

$$\lambda\phi_1^n + \psi = 0. \tag{1.29}$$

An optimal solution of the first order optimality system can be found by projected gradient methods where the solution of the gradient equation (1.29) is computed by a serial realization of the state and adjoint equations. The idea is to start with a feasible solution of (1.24) and seek a control $u_{(k)}^n = F_0^n$ that minimizes $L(\vec{\rho_i}^n, \vec{u}^n, \vec{\phi_i^n}, \psi)$ iteratively for each level k. To ensure that $\sum_{n=1}^{N_T-1} u_{(k+1)}^n = S\Delta t^{-1}$, the steepest descent direction $d_{(k)}$ must fulfill the condition

$$0 = \sum_{n=1}^{N_T-1} (u_{(k)}^n + \sigma_{(k)} d_{(k)}^n) - S\Delta t^{-1} = \sigma_{(k)} \sum_{n=1}^{N_T-1} d_{(k)}^n,$$

where $\sigma_{(k)} > 0$ is the step size of the corresponding gradient descent method. For instance, the step size $\sigma_{(k)}$ can be computed by the Armijo rule; see [93]. Furthermore, we select the steepest descent direction as

$$d_{(k)}^n = -\lambda\phi_1^n - \psi, \tag{1.30}$$

where $n = 1, ..., N_T - 1$ and $\sum_{n=1}^{N_T-1} d_{(k)}^n = 0$. Thus, the adjoint ψ can be computed as follows

$$\psi = \frac{1}{N_T - 1} \sum_{n=1}^{N_T-1} \lambda\phi_1^n. \tag{1.31}$$

Now we summarize the previous computational steps to a solution algorithm.

Solution Algorithm

Initial values: $\rho_i^1, u_{(0)}$ with $\sum_{n=1}^{N_T-1} u_{(0)}^n = S$, f_{out}

1. Solve ρ_i^n for $n = 2, ..., N_T$ by the forward simulation.

2. Solve the adjoint system ϕ_i^n for $i = 1, ..., N$ and $n = 1, ..., N_T - 1$.

3. Compute the adjoint ψ.

4. Compute the descent direction $d_{(k)}$,

$$d_{(k)} = -\lambda\phi_1^n - \psi,$$

5. Update the control $u_{(k+1)} = u_{(k)} + \sigma_{(k)}d_{(k)}$.

6. If $\|d_{(k)}\| \geq \varepsilon$ Go to 1, otherwise STOP.

Instead of considering the second-order optimality system to check that there really exists a local minimum, we follow another way. The idea is to consider an alternative optimization approach which can be solved to global optimality. Having such a tool at hand, we show the connection between the proposed optimization models.

1.4.3 Mixed Integer Programming Model

Mixed-integer programming (MIP) models can be used to solve a special class of pde-constrained optimization problems. Since the optimal inflow problem (1.24) has a nearly linear structure, it is closely related to the problems mentioned in [25, 37, 51]. Usually a MIP model consists of a linear cost functional combined with linear constraints and floating and integer variables. Generally, a mixed integer model (MIP) has the following form

$$Z(X) = \min\{c^T x : x \in X\},$$

where X describes the set of feasible solutions

$$X = \{x \in \mathbb{R}_+^{n-p} \times \{0,1\}^p : Ax \geq b\}.$$

The only nonlinearity appearing in (1.24) is the flux function. Here, we apply a standard linearization (see [37]) by introducing binary variables $\xi_i^n \in \{0,1\}$.

$$\begin{aligned}
&\text{(FLUX1):} \quad && a\rho_{i-1}^n - \xi_{i-1}^n\mathcal{M} \leq F_{i-1}^n,\\
&\text{(FLUX2):} \quad && F_{i-1}^n \leq a\rho_{i-1}^n,\\
&\text{(FLUX3):} \quad && \frac{\rho_{max} - \rho_i^n}{\lambda} + F_i^n - (1 - \xi_{i-1}^n)\mathcal{M} \leq F_{i-1}^n,\\
&\text{(FLUX4):} \quad && F_{i-1}^n \leq \frac{\rho_{max} - \rho_i^n}{\lambda} + F_i^n.
\end{aligned}$$

where $i = 2, \ldots, N$, $n = 1, \ldots, N_T - 1$. Additionally $\mathcal{M}$ is a large number, i.e., $\mathcal{M} \gg a\rho_{max}$. All other equations in (1.24) are already discretized in space and

time and can therefore be directly interpreted as constraints of a MIP. Summarizing, this leads to

$$\min J(\rho_i^n) = \sum_{n=1}^{N_T} \sum_{i=1}^{N} C_i^n \rho_i^n \tag{1.32}$$

subject to

(PDE), (FLUX1) - (FLUX4), (OUTFLOW), (SUPPLY), (BOX).

MIP problems are solved using common software packages, e.g. CPLEX [71]. Note that increasing the number of binary variables, the computation time of the MIP may blow up. One possibility to reduce the computational effort provides the following lemma. We introduce an extension of the MIP model by including new constraints such that the original problem is solved faster.

Lemma 1.4.1. *Let the MIP model given by* (1.32)*. Then, the additional constraint for all* $n = 1, \dots, N_T$

$$\xi_{i-1}^n \leq \xi_i^n, \quad i = 2, ..., N, \tag{1.33}$$

ensures an eligible restriction on the binary variables.

Proof. Consider the binary variable $\xi_i^n = 0$. Then, the inequalities (FLUX1) - (FLUX3) yield

$$F_i^n = a\rho_i^n.$$

and the numerical flux will be

$$F_{i-1}^n = \min\{a\rho_{i-1}^n, \frac{\rho_{max} - \rho_i^n}{\lambda} + a\rho_i^n\}. \tag{1.34}$$

We assume that $0 \leq \rho_i^n \leq \rho_{max}$ for all indices i, n. It holds that

$$\begin{aligned} 0 &\leq \min\{\rho_{max} - \rho_{i-1}^n, \rho_{max} - \rho_i^n\} \\ &= \rho_{max} - \max\{\rho_{i-1}^n, \rho_i^n\} \\ &\leq \rho_{max} + (a\lambda - 1)\rho_i^n - a\lambda\rho_{i-1}^n. \end{aligned}$$

finally resulting in

$$a\rho_{i-1}^n \leq \frac{\rho_{max} - \rho_i^n}{\lambda} + a\rho_i^n. \tag{1.35}$$

Combining the results (1.34) and (1.35), the flux F_{i-1}^n simplifies to

$$F_{i-1}^n = \min\{a\rho_{i-1}^n, \frac{\rho_{max} - \rho_i^n}{\lambda} + a\rho_i^n\} = a\rho_{i-1}^n.$$

The inequalities (FLUX1) - (FLUX3) lead to $\xi_{i-1}^n = 0$. The choice of $\xi_i^n = 1$ as a starting point works analogously. □

The interpretation of Lemma 1.4.1 can be also done looking at the discussion of the Riemann problems in Subsection 1.2.4. The only combination of binaries that is not allowed is $\xi_{i-1}^n = 1$ and $\xi_i^n = 0$. This corresponds to a sequence of congestion followed by a free flow regime, i.e., this is impossible.

1.4.4 Comparison of Optimization Approaches

So far, we have presented two solution approaches for (1.24) which may at first sight seem different. In this section, we connect both optimization approaches. We formally compare the adjoint variables (1.25) with the dual variables of the relaxed MIP (1.32). We transform the relaxed MIP into its dual problem and explain in a second step the relevant similarities to (1.25).

Definition 1.4.2 (Linear Program)**.** *A linear problem (LP) has the following form: Find the vector $x \in \mathbb{R}^n_+$ that solves*

$$\begin{array}{ll} \min & c^T x \\ s.t. & Ax \geq b \\ & x \geq 0 \end{array} \tag{1.36}$$

with given vectors $c \in \mathbb{R}^n$, $b \in \mathbb{R}^m$ and a given matrix $A \in \mathbb{R}^{m \times n}$

Definition 1.4.3 (Dual Program)**.** *A dual problem of* (1.36) *has the following form: Find the vector $\varphi \in \mathbb{R}^m_+$ that solves*

$$\begin{array}{ll} \max & b^T \varphi \\ s.t. & A^T \varphi \leq c \\ & \varphi \geq 0 \end{array} \tag{1.37}$$

with given vectors $c \in \mathbb{R}^n$, $b \in \mathbb{R}^m$ and a given matrix $A \in \mathbb{R}^{m \times n}$

Let the binary variables are treated as real-valued. Then, the relaxed MIP with $\rho_i^n, F_i^n \in \mathbb{R}$ reads for all indices n, i:

$$\min \quad \sum_{n=1}^{N_T} \sum_{i=1}^{N} C_i^n \rho_i^n$$

s.t. (PDE) constraints:

$$\begin{array}{lll} \Phi_i^n: & \rho_i^{n+1} - \rho_i^n + \lambda F_i^n - \lambda F_{i-1}^n & = 0 \\ \Phi_i^0: & \rho_i^1 & = \frac{1}{\Delta x} \int_{x_{i-1}}^{x_i} \rho_0(x) dx \end{array}$$

(FLUX) constraints:

$$\begin{array}{lll} \varphi_{1,i}^n: & a\rho_i^n - F_i^n & \geq 0 \\ \varphi_{2,i}^n: & F_{i+1}^n - F_i^n - \lambda^{-1} \rho_{i+1}^n & \geq -\rho_{max} \lambda^{-1} \\ \varphi_{2,N}^n: & -F_N^n & \geq -f_{out}^n \end{array} \tag{1.38}$$

(SUPPLY) constraints:

$$\begin{array}{lll} \Psi: & \sum_{n=1}^{N_T - 1} F_0^n & = S \Delta t^{-1} \\ \Psi^n: & -F_0^n & \geq -W \end{array}$$

where the corresponding dual variables are introduced in an extra column. This leads to the dual program of the form:

$$\begin{aligned}\max\ & \sum_{i=1}^{N} \frac{1}{\Delta x} \int_{x_{i-1}}^{x_i} \rho_0(x)dx\Phi_i^0 \\ & - \sum_{n=1}^{N_T-1} \sum_{i=1}^{N-1} (\rho_{max}\lambda^{-1})\varphi_{2,i}^n - f_{out}^n \varphi_{2,N}^n - W\Psi^n + S\Delta t^{-1}\Psi\end{aligned}$$

s.t.

$$\begin{array}{lll} \rho_i^n: & \Phi_i^{n-1} - \Phi_i^n + a\varphi_{1,i}^n - \lambda^{-1}\varphi_{2,i-1}^n & = C_i^n \\ \rho_i^1: & \Phi_i^0 - \Phi_i^1 & = C_i^1 \\ \rho_i^{N_T}: & \Phi_i^{N_T-1} & = C_i^{N_T} \end{array} \tag{1.39}$$

$$\begin{array}{lll} F_i^n: & \lambda\Phi_i^n - \lambda\Phi_{i+1}^n - \varphi_{1,i}^n + \varphi_{2,i-1}^n - \varphi_{2,i}^n & = 0 \\ F_N^n: & \lambda\Phi_N^n - \varphi_{1,N}^n - \varphi_{2,N}^n + \varphi_{2,N-1}^n & = 0 \\ F_0^n: & -\lambda\Phi_1^n + \Psi - \Psi^n & = 0 \end{array}$$

with $\Phi_i^n, \Psi \in \mathbb{R}, \Psi^n, \varphi_{k,i}^n \in \mathbb{R}^+, k = 1, 2$ and where again the dual variables are denoted in a separate column. We set $\varphi_{2,0}^n := 0$ as well.
The aim is now to compare the dual variables of the dual MIP (1.39) with the adjoint variables (1.25).
Let us assume that ρ_i^n, F_i^n are a feasible solution of the primal problem (1.38). We are interested in finding a feasible (not necessarily optimal) solution of the dual relaxed problem (1.39). By applying the complementary slackness theorem it is possible to obtain an (optimal) solution to the dual when only an (optimal) solution to the primal is known. In other words: If a MIP solution ρ_i^n, F_i^n is optimal for the primal problem, then the dual slack variables $\varphi_{1,i}^n, \varphi_{2,i}^n$ fulfill the complementary slackness conditions (1.40). In case of no optimality, we have no restriction for the dual states $\varphi_{1,i}^n, \varphi_{2,i}^n$ with respect to the primal state, i.e., the dual variables can be chosen freely. We intend to pick those dual variables $\varphi_{1,i}^n, \varphi_{2,i}^n$ such that the complementary slackness condition is satisfied:

$$\begin{aligned} \varphi_{1,k}^n(F_k^n - a\rho_k^n) &= 0 \\ \varphi_{2,k}^n(F_k^n - F_{k+1}^n + \lambda^{-1}\rho_{k+1}^n - \rho_{max}\lambda^{-1}) &= 0 \end{aligned} \tag{1.40}$$

for $k = i-1, i$. In fact, due to the theoretical investigations, we have to analyze three different scenarios: freeflow, blocking and release of congestions.

Case 1: Freeflow
Let $F_k^n = a\rho_k^n$ and $F_k^n < \frac{\rho_{max}-\rho_{k+1}^n}{\lambda} + F_{k+1}^n$ for $k = i-1, i$. Thus, due to (1.40), the dual variables must be

$$\varphi_{1,k}^n \geq 0, \quad \varphi_{2,k}^n = 0, \quad k = i-1, i,$$

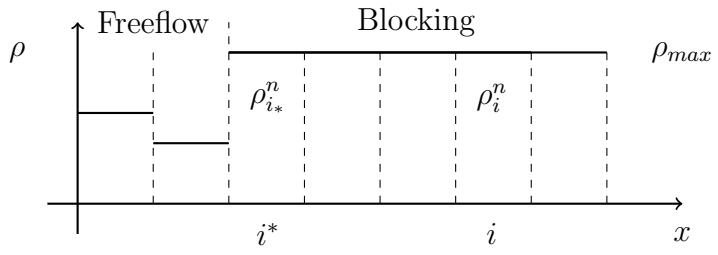


Figure 1.9: Blocked and freeflow regions

and the constraints of the dual problem turn into

$$\Phi_i^{n-1} - \Phi_i^n + a\varphi_{1,i}^n = C_i^n$$
$$\lambda\Phi_i^n - \lambda\Phi_{i+1}^n - \varphi_{1,i}^n = 0.$$

Rearranging the last two equations, we end up with

$$\Phi_i^{n-1} = C_i^n + \Phi_i^n - a\lambda(\Phi_i^n - \Phi_{i+1}^n).$$

The equation for Φ_i^{n-1} is similar to ϕ_i^{n-1} in (1.25) since $\partial_{\rho_i^n} F_j^n = a$.

Case 2: Blocking
Let $F_k^n < a\rho_k^n$ and $F_k^n = \frac{\rho_{max}-\rho_{k+1}^n}{\lambda} + F_{k+1}^n$ for $k = i^*, \dots, i$ with $i^* \leq i$ and additionally $F_{i^*-1}^n = a\rho_k^n$ and $F_{i^*-1}^n < \frac{\rho_{max}-\rho_{i^*}^n}{\lambda} + F_{i^*}^n$, see Figure 1.9. Then, the complementary slackness condition (1.40) reveals

$$\varphi_{1,k}^n = 0, \quad \varphi_{2,k}^n \geq 0, \quad \varphi_{2,i^*-1}^n = 0, \quad \forall k = i^*, \dots, i,$$

$$\lambda\Phi_i^n - \lambda\Phi_{i+1}^n + \varphi_{2,i-1}^n - \varphi_{2,i}^n = 0.$$

This yields

$$\begin{aligned}\varphi_{2,i-1}^n &= \varphi_{2,i}^n + \lambda\Phi_{i-1}^n - \lambda\Phi_i^n \\ &= \sum_{k=i^*}^{i-1} (\lambda\Phi_k^n - \lambda\Phi_{k+1}^n) = -\lambda\Phi_i^n + \lambda\Phi_{i^*}^n.\end{aligned}$$

This result can be plugged into the constraint

$$\Phi_i^{n-1} - \Phi_i^n - \lambda^{-1}\varphi_{2,i-1}^n = C_i^n,$$

leading to

$$\Phi_i^{n-1} = C_i^n + \Phi_{i^*}^n. \tag{1.41}$$

Again, we compare the dual variables Φ_i^{n-1} with the adjoints ϕ_i^{n-1} in (1.25), but with the crucial difference that the partial derivatives of $\partial_{\rho_i^n} F_j^n$ are more involved, cf. (1.27) and (1.28). If $\alpha \leq \beta$, the derivative of the smoothed minimum function $\min_\epsilon$ is

$$\partial_1 \min_\epsilon(\alpha, \beta) = 1, \quad \partial_2 \min_\epsilon(\alpha, \beta) = 0$$

or otherwise, for $\alpha > \beta$,

$$\partial_1 \min_\epsilon(\alpha, \beta) = \frac{\epsilon^2}{(\alpha - y + \epsilon)^2}, \quad \partial_2 \min_\epsilon(\alpha, \beta) = \frac{-\epsilon^2}{(\alpha - \beta + \epsilon)^2} + 1.$$

Using the Taylor expansion to simplify the expressions $\partial_1 \min_\epsilon(\alpha, \beta), \partial_2 \min_\epsilon(\alpha, \beta)$ for $\alpha > \beta$ yields

$$\begin{aligned}
\partial_1 \min_\epsilon(\alpha, \beta) &= \frac{\epsilon^2}{\alpha - \beta} + \mathcal{O}(\epsilon^4) = \mathcal{O}(\epsilon^2) \\
\partial_2 \min_\epsilon(\alpha, \beta) &= 1 - \frac{\epsilon^2}{\alpha - \beta} + \mathcal{O}(\epsilon^4) = 1 + \mathcal{O}(\epsilon^2)
\end{aligned}$$

Then, the partial derivatives of the numerical flux $\partial_{\rho_i^n} F_j^n$ can be expressed as

$$\partial_{\rho_i^n} F_j^n = \partial_1 \min_\epsilon \{a\rho_j^n, \frac{\rho_{max} - \rho_{j+1}^n}{\lambda} + F_{j+1}^n\} \cdot a\delta_{i,j} = \mathcal{O}(\epsilon^2)$$

for $i = j$ and

$$\partial_{\rho_i^n} F_j^n = \partial_2 \min_\epsilon \{a\rho_j^n, \frac{\rho_{max} - \rho_{j+1}^n}{\lambda} + F_{j+1}^n\} \cdot (-\delta_{i,j+1}\frac{1}{\lambda} + \partial_{\rho_i^n} F_{j+1}^n) = -\frac{1}{\lambda} + \mathcal{O}(\epsilon^2)$$

for $i^* \leq j < i$. If $j < i^*$, the derivatives are $\partial_{\rho_{i^*}^n} F_j^n = 0$. A small computation shows that the dual variables Φ_i^{n-1} in (1.41) and the adjoint variables ϕ_i^{n-1} from (1.25) only differ $\mathcal{O}(\epsilon^2)$:

$$\begin{aligned}
\phi_i^{n-1} &= C_i^n + \phi_i^n - \lambda \sum_{j=1}^{N} (\partial_{\rho_i^n} F_j^n - \partial_{\rho_i^n} F_{j-1}^n)\phi_j^n \\
&= C_i^n + \phi_i^n - (1 + \mathcal{O}(\epsilon^2))\phi_{i^*}^n - \lambda \sum_{j=i^*+1}^{i-1} \left(\frac{-1}{\lambda} - \frac{-1}{\lambda} + \mathcal{O}(\epsilon^2) \right) \phi_j^n - (1 + \mathcal{O}(\epsilon^2))\phi_i^n \\
&= C_i^n + \phi_{i^*}^n + \mathcal{O}(\epsilon^2).
\end{aligned}$$

Case 3: Release
Let $F_k^n = a\rho_k^n$ and $F_k^n = \frac{\rho_{max} - \rho_{i+1}^n}{\lambda} + F_{k+1}^n$ for $k = i-1, i$. Then, the complementary slackness conditions (1.40) does not contain any information about the dual variables, i.e.,

$$\varphi_{1,k}^n \geq 0, \quad \varphi_{2,k}^n \geq 0, \quad k = i-1, i.$$

Note that the choice of $\varphi_{2,k}^n$ is not unique. Looking at the objective function of (1.39), we see that the maximization problem forces small values of $\varphi_{2,k}^n$. In the ideal case, $\varphi_{2,k}^n = 0$ and Case 3 (Release) immediately reduces to Case 1 (Freeflow).

1.4.5 Connection between the MIP and the relaxed LP

In general, solution routines of MIP models are very expensive and time consuming. Typical methods are *Branch and Bound* and *Branch and Cut* approaches, see [85]. In contrast to MIP models, Linear Programs (LP) are easier to solve. In consideration of the optimal inflow problem, the relaxed LP and the MIP model have similar structures. The crucial difference of both problems is the usage of binary variables in the flux constraints. Under certain assumptions, it is sufficient to solve only the relaxed LP for solving the inflow problem.

Theorem 1.4.4. *Let ρ_i^n, F_i^n be an optimal solution of the relaxed mixed integer model* (1.38) *with the objective function*

$$J(\rho_i^n) = \sum_{n=1}^{N_T} \sum_{i=1}^{N} \rho_i^n.$$

Then F_0^n is an optimal inflow of the MIP model (1.32) *for all $n = 1, ..., N_T - 1$.*

Proof. Let F_0^n be components of the optimal solution of the relaxed mixed integer model (1.38). At first, we construct a feasible solution $\bar{\rho}_i^n$, $\bar{F}_i^n$ of the MIP model (1.32) by the forward simulation with input parameter F_0^n and ρ_i^1, i.e.,

$$\begin{aligned} \bar{\rho}_i^1 &:= \rho_i^1, \quad i = 1, ..., N, \\ \bar{F}_0^n &:= F_0^n, \quad n = 1, ..., N_T - 1, \end{aligned}$$

$$\begin{aligned} \text{(PDE):} \qquad & \bar{\rho}_i^{n+1} := \bar{\rho}_i^n - \lambda[\bar{F}_i^n - \bar{F}_{i-1}^n], \\ \text{(FLUX):} \qquad & \bar{F}_i^n := \min\{a\bar{\rho}_i^n, \frac{\rho_{max} - \bar{\rho}_{i+1}^n}{\lambda} + \bar{F}_{i+1}^n\}, \\ \text{(OUTFLOW):} \qquad & \bar{F}_N^n := \min\{a\bar{\rho}_N^n, f_{out}^n\}, \end{aligned}$$

for $i = 1, ..., N$ and $n = 1, ..., N_T - 1$.
Obviously, $\bar{F}_0^n$ fulfills the linear constraints (SUPPLY) and (BOX). Hence, $\bar{\rho}_i^n$, $\bar{F}_i^n$ is a feasible solution of the mixed integer model (1.32).
The next step is to show, that the objective values of the relaxed model and the MIP model coincide, i.e.,

$$J(\rho_i^n) = \sum_{n=1}^{N_T-1} \sum_{i=1}^{N} \rho_i^n = \sum_{n=1}^{N_T-1} \sum_{i=1}^{N} \bar{\rho}_i^n = J(\bar{\rho}_i^n).$$

Then $\bar{\rho}_i^n$, $\bar{F}_i^n$ is a feasible optimal solution of the mixed integer model.
We prove inductively the following relation

$$\sum_{n=1}^{m} F_i^n \leq \sum_{n=1}^{m} \bar{F}_i^n, \quad i = 1, ..., N. \tag{1.42}$$

In due of the inequality constraint of the relaxed MIP, the solution variable F_i^n have the following estimations

$$F_i^n \leq a\rho_i^n, \tag{1.43}$$

$$F_i^n \leq \frac{\rho_{max} - \rho_{i+1}^n}{\lambda} + F_{i+1}^n, \quad i < N, \tag{1.44}$$

$$F_N^n \leq f_{out}^n. \tag{1.45}$$

Induction start: Let $m = 1$, it is necessary to show $F_i^1 \leq \bar{F}_i^1$ for all $i = 1, ..., N$.
By using the inequalities (1.43) and (1.45), we can estimate

$$F_N^1 \underset{(1.45)}{\overset{(1.43)}{\leq}} \min\{a\rho_N^1, f_{out}^1\} = \min\{a\bar{\rho}_N^1, f_{out}^1\} \overset{\text{(OUTFLOW)}}{=} \bar{F}_N^1.$$

By induction with respect of i and the inequalities (1.43) and (1.44), we get

$$\begin{aligned} F_i^1 &\overset{(1.43),(1.44)}{\leq} \min\{a\rho_i^1, \frac{\rho_{max} - \rho_{i+1}^1}{\lambda} + F_{i+1}^1\} \\ &\leq \min\{a\bar{\rho}_i^1, \frac{\rho_{max} - \bar{\rho}_{i+1}^1}{\lambda} + \bar{F}_{i+1}^1\} \overset{\text{(FLUX)}}{=} \bar{F}_i^1. \end{aligned}$$

Induction hypothesis: The statement

$$\sum_{n=1}^{m} F_i^n \leq \sum_{n=1}^{m} \bar{F}_i^n, \quad i = 1, ..., N \tag{1.46}$$

has been proven. We show that

$$\sum_{n=1}^{m+1} F_i^n \leq \sum_{n=1}^{m+1} \bar{F}_i^n, \quad i = 1, ..., N.$$

Induction step: $m \to m + 1$
The densities are representable by (PDE):

$$\rho_i^{m+1} \overset{\text{(PDE)}}{=} \rho_i^1 - \lambda \sum_{n=1}^{m} F_i^n + \lambda \sum_{n=1}^{m} F_{i-1}^n, \tag{1.47}$$

$$\bar{\rho}_i^{m+1} \overset{\text{(PDE)}}{=} \bar{\rho}_i^1 - \lambda \sum_{n=1}^{m} \bar{F}_i^n + \lambda \sum_{n=1}^{m} \bar{F}_{i-1}^n. \tag{1.48}$$

One evaluates $\bar{F}_i^{m+1}$ in three cases.
Case 1: Let $\bar{F}_N^{m+1} = f_{out}^{m+1}$. Inequality (1.45) yields $F_N^{m+1} \leq f_{out}^{m+1} = \bar{F}_N^{m+1}$.
By induction hypothesis (1.46) we get

$$\sum_{n=1}^{m+1} F_N^n \leq \sum_{n=1}^{m+1} \bar{F}_N^n$$

Case 2: Now let $i = 1, ..., N$ and $\bar{F}_i^{m+1} = a\bar{\rho}_i^{m+1}$
Inequality (1.43) yields the following statement

$$\begin{aligned}\sum_{n=1}^{m+1} F_i^n = \sum_{n=1}^{m} F_i^n + F_i^{m+1} &\overset{(1.43)}{\leq} \sum_{n=1}^{m} F_i^n + a\rho_i^{m+1} \\ &\overset{(1.47)}{=} \sum_{n=1}^{m} F_i^n + a\rho_i^1 - a\lambda \sum_{n=1}^{m} F_i^n + a\lambda \sum_{n=1}^{m} F_{i-1}^n \\ &= a\rho_i^1 + (1 - a\lambda) \sum_{n=1}^{m} F_i^n + a\lambda \sum_{n=1}^{m} F_{i-1}^n\end{aligned}$$

The CFL condition yields $(1 - a\lambda) \geq 0$. By Induction hypothesis (1.46) and (1.48), one obtains

$$\begin{aligned}\sum_{n=1}^{m+1} F_i^n &\overset{(1.46)}{\leq} a\bar{\rho}_i^1 + (1 - a\lambda) \sum_{n=1}^{m} \bar{F}_i^n + a\lambda \sum_{n=1}^{m} \bar{F}_{i-1}^n \\ &\overset{(1.48)}{=} \sum_{n=1}^{m} \bar{F}_i^n + a\bar{\rho}_i^{m+1} = \sum_{n=1}^{m+1} \bar{F}_i^n.\end{aligned}$$

Case 3: Now let $i = 1, ..., N-1$ and $\bar{F}_i^{m+1} = \frac{\rho_{max} - \bar{\rho}_{i+1}^{m+1}}{\lambda} + \bar{F}_{i+1}^{m+1}$.
For the proof of the statement $\sum_{n=1}^{m+1} F_i^n \leq \sum_{n=1}^{m+1} \bar{F}_i^n$, it is necessary that the inequality $\sum_{n=1}^{m+1} F_{i+1}^n \leq \sum_{n=1}^{m+1} \bar{F}_{i+1}^n$ holds. This statement is already proved for $i = N$ by the cases 1 and 2. By an additional induction, we can prove this for $i = N-1, N-2, ..., 2, 1$ by the cases 2 and 3.
Thus, (1.44) and (1.47) yields the following estimation

$$\begin{aligned}\sum_{n=1}^{m+1} F_i^n = \sum_{n=1}^{m} F_i^n + F_i^{m+1} &\overset{(1.44)}{\leq} \sum_{n=1}^{m} F_i^n + \frac{1}{\lambda}(\rho_{max} - \rho_{i+1}^{m+1}) + F_{i+1}^{m+1} \\ &\overset{(1.47)}{\leq} \sum_{n=1}^{m} F_i^n + \frac{1}{\lambda}(\rho_{max} - \rho_{i+1}^1) - \sum_{n=1}^{m} F_i^n + \sum_{n=1}^{m} F_{i+1}^n + F_{i+1}^{m+1} \\ &= \frac{1}{\lambda}(\rho_{max} - \rho_{i+1}^1) + \sum_{n=1}^{m+1} F_{i+1}^n \leq \frac{1}{\lambda}(\rho_{max} - \bar{\rho}_{i+1}^1) + \sum_{n=1}^{m+1} \bar{F}_{i+1}^n.\end{aligned}$$

Now we expand the previous inequality by $0 = \sum_{n=1}^{m} \bar{F}_i^n - \sum_{n=1}^{m} \bar{F}_i^n$ and use (1.48). This yields

$$\begin{aligned}\sum_{n=1}^{m+1} F_i^n &\leq \frac{1}{\lambda}(\rho_{max} - \bar{\rho}_{i+1}^1 + \lambda \sum_{n=1}^{m} \bar{F}_{i+1}^n - \lambda \sum_{n=1}^{m} \bar{F}_i^n) + \bar{F}_{i+1}^{m+1} + \sum_{n=1}^{m} \bar{F}_i^n \\ &= \frac{1}{\lambda}(\rho_{max} - \bar{\rho}_{i+1}^{m+1}) + \bar{F}_{i+1}^{m+1} + \sum_{n=1}^{m} \bar{F}_i^n \overset{(1.48)}{=} \bar{F}_i^{m+1} + \sum_{n=1}^{m} \bar{F}_i^n.\end{aligned}$$

Finally, we get

$$\sum_{n=1}^{m+1} F_i^n \leq \sum_{n=1}^{m+1} \bar{F}_i^n, \quad i = 0, ..., N,$$

and we finish the proof of (1.42). Now we consider the sum of the density for the time-step m

$$\begin{aligned}\sum_{i=1}^{N} \rho_i^m &\overset{(1.47)}{=} \sum_{i=1}^{N} \rho_i^1 + \lambda \sum_{n=1}^{m-1} F_0^n - \lambda \sum_{n=1}^{m-1} F_N^n \\ &\overset{(1.42)}{\geq} \sum_{i=1}^{N} \bar{\rho}_i^1 + \lambda \sum_{n=1}^{m-1} \bar{F}_0^n - \lambda \sum_{n=1}^{m-1} \bar{F}_N^n \\ &\overset{(1.48)}{=} \sum_{i=1}^{N} \bar{\rho}_i^m.\end{aligned} \tag{1.49}$$

The next step is the evaluation of the objective function J for the constructed solution $\bar{\rho}_i^n$. Moreover, the objective value of the optimal relaxed MIP solution can be estimated by (1.49), i.e.,

$$J(\rho_i^n) = \sum_{n=1}^{N_T-1} \sum_{i=1}^{N} \rho_i^n \overset{(1.49)}{\geq} \sum_{n=1}^{N_T-1} \sum_{i=1}^{N} \bar{\rho}_i^n = J(\bar{\rho}_i^n). \tag{1.50}$$

However, $\bar{\rho}_i^n$, $\bar{F}_i^n$ is a feasible solution of the MIP model and also of the relaxed MIP model. Furthermore, the solution of the relaxed MIP model ρ_i^n, F_i^n is optimal by assumption. Hence, the objective value of ρ_i^n cannot be larger than the objective value of $\bar{\rho}_i^n$. In consideration of (1.50), the objective values of ρ_i^n and $\bar{\rho}_i^n$ must be equal, i.e.,

$$J(\rho_i^n) = \sum_{n=1}^{N_T-1} \sum_{i=1}^{N} \rho_i^n = \sum_{n=1}^{N_T-1} \sum_{i=1}^{N} \bar{\rho}_i^n = J(\bar{\rho}_i^n).$$

Finally, $\bar{\rho}_i^n$, $\bar{F}_i^n$ is a feasible optimal solution of the mixed integer model.

□

Remark 1.4.5. *The proof of Theorem 1.4.4 works only for the presented linear objective function*

$$J(\rho_i^n) = \sum_{n=1}^{N_T} \sum_{i=1}^{N} \rho_i^n.$$

If we consider an arbitrary objective function, e.g., $J(\rho_i^n) = \sum_{n=1}^{N_T} \sum_{i=1}^{N} C_i^n \rho_i^n$, *the inequality* (1.49) *cannot estimate* (1.50) *and the proof does not work.*

1.4.6 The Maintenance Problem

In case of a maintenance, it is necessary to shut down a machine in progress for a certain time interval. Thus, the stopped machine can be repaired or be checked for the maintenance. The task is to find an optimal time interval for a machine shutdown, where the duration of a maintenance is known. Therefore, we look for an time interval to stop a machine efficiently such the capacity of all machines in a production line is reduced. After the maintenance the production is continued. The optimization problem is based on the discrete formulation in 1.3.3. The maintenance optimization approach is an extension of the MIP model in 1.4.3. The maintenance problem for supply-chains in due of MIP modeling is already investigated in [37].
In the following, we consider a maintenance only for the last machine in a production line. In our model, a shutdown process can be simulated if the flux is set to zero at the local point x_N for a time interval. The problem is defined as

$$\begin{aligned} &\min_j J(\rho_i^n) = \sum_{n=1}^{N_T} \sum_{i=1}^{N} C_i^n \rho_i^n \qquad (1.51)\\ &\text{subject to}\\ &(\text{PDE}), (\text{FLUX}) \text{ ,(INFLOW)}\\ &F_N^n = \begin{cases} 0 & \text{for all } n \in [j, j+N_{off}],\\ a\rho_N^n & \text{for all } n \notin [j, j+N_{off}], \end{cases} \end{aligned}$$

where j is the discrete start time of the maintenance. The length of the time interval is given by $N_{off} \in \mathbb{N}$. Moreover, N_{off} is the number of discrete time-steps for a machine shutdown. As an extension of the MIP formulation of 1.4.3, we introduce additional binary variables θ^j. If θ^j is one, the maintenance interval starts at the discrete time j. We assume, that the time interval is unique and has only one starting point j. Thus, if a j exists such that $\theta^j = 1$ then $\theta^n = 0$ for all $j \neq n$. This yields the constraint

$$\text{(SHUTDOWN 1):} \qquad \sum_{j=1}^{N_T - N_{off}} \theta^j = 1.$$

For a j the flux $F_N^n = 0$ if $n \in [j, j + N_{off}]$. Otherwise, $F_N^n = a\rho_N^n$. This yields the following constraints of the MIP:

$$\begin{aligned}&\text{(SHUTDOWN 2):} \qquad F_N^l \leq (1-\theta^j)\mathcal{M} \quad l \in \{j, ..., j + N_{off} - 1\},\\ &\text{(SHUTDOWN 3):} \qquad a\rho_N^l - (1-\theta^j)\mathcal{M} \leq F_N^l \quad l \notin \{j, ..., j + N_{off} - 1\},\end{aligned}$$

for all $j = 1, ..., N_T - N_{off}$.

$$\text{(SHUTDOWN 4):} \qquad F_N^n \leq a\rho_N^n \quad n = 1, ..., N_T - 1.$$

Finally, this leads to

$$\begin{aligned}&\min J(\rho_i^n) = \sum_{n=1}^{N_T}\sum_{i=1}^{N} C_i^n \rho_i^n\\ &\text{subject to}\\ &\text{(PDE), (INFLOW),}\\ &\text{(FLUX 1) - (FLUX 4), (SHUTDOWN 1) - (SHUTDOWN 4).}\end{aligned}$$

1.5 Numerical Results

Finally, we present computational results of the 1D model and their optimization issues. In particular, we cover the following aspects:

- In Subsection 1.5.1 we compare the DFG methods against the Wave Front tracking method.
- In Subsection 1.5.2 we give a validation of the DFG method by comparison with the Godunov Method for the regularized problem (RFG). Additionally, we highlight the computational efficiency of the DFG method and the RFG method.
- In Subsection 1.5.3 we compare the numerical results of the microscopic model against the continuous model.
- In Subsection 1.5.4 we investigate the results of adjoint approach and the MIP model for the inflow problem.
- In Subsection 1.5.5 we consider an computational example for the optimal date for a maintenance.

All computations are performed on the same platform, namely a 3.0 GHz Dual-core computer with 8 GB RAM. The algorithms are implemented in MATLAB [83]. The MIP and the LP models are solved using the commercial solver ILOG CPLEX [71].

1.5.1 Wave Front Tracking Algorithm vs. DFG method

At first, we test the DFG method against the classical front-tracking algorithm. The latter is a grid-independent method, i.e., there is no CFL condition, that tracks all propagating waves and their interactions, see Subsection 1.2.4.
We consider the initial boundary value problem for solving (1.11) with (1.13):

$$\rho_0(x) = 0.4\sin(\pi x) + 0.4$$
$$f(\rho(0,t)) = a\rho(0,t) = 0.2.$$

To compare the approximate solutions of both methods we choose a time horizon of $T = 1$ as well as $a = 1$ and $\rho_{max} = 1$. In the time interval $0.5 \leq t < 0.8$, the outflow $f_{out}(t)$ is as large as possible, i.e., freeflow regime. Otherwise the outflow is blocked.

$$f_{out}(t) = \begin{cases} 1 & \text{if} \quad t \in [0.5, 0.8), \\ 0 & \text{otherwise.} \end{cases}$$

For the wave front tracking algorithm, we divide the spatial domain into $N = 20$ cells. The initial data for the multiple Riemann problem is given by the cell integrals over $\rho_0(x)$, i.e.,

$$\rho_i = \frac{1}{\Delta x} \int_{x_{i-1}}^{x_i} \rho_0(x)\, dx \quad \text{for all } i = 1, \ldots, N.$$

The inflow $f(\rho(0,t))$ can be translated into a Riemann problem of the form $\rho_r = \rho_1$ and $\rho_l = \rho_0 = 0.2$. The multiple Riemann problem is shown in Figure 1.10.

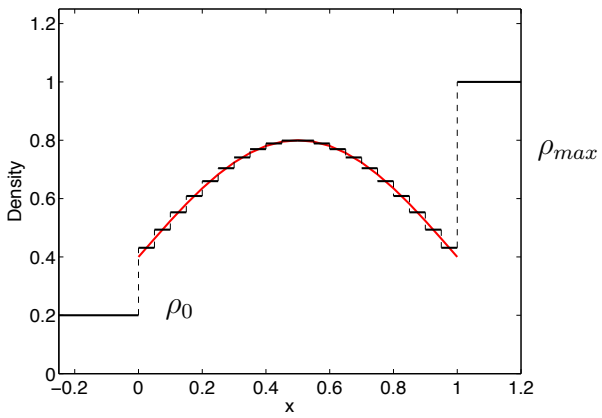
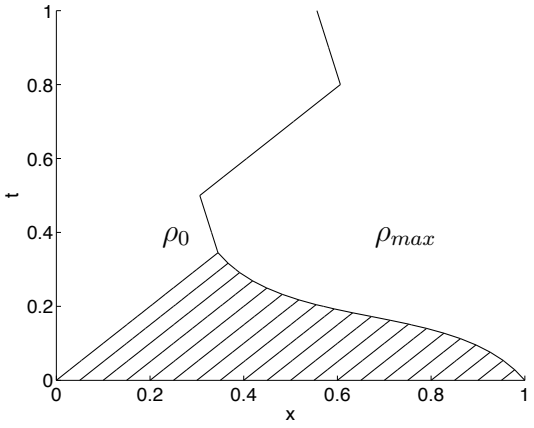
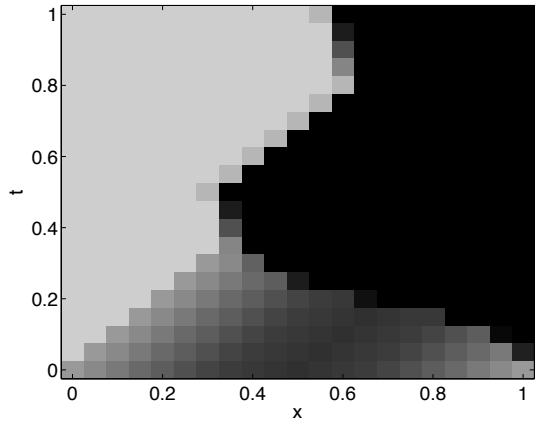


Figure 1.10: Initial data for the multiple Riemann problem (black), continuous initial data $\rho_0(x)$ (red).

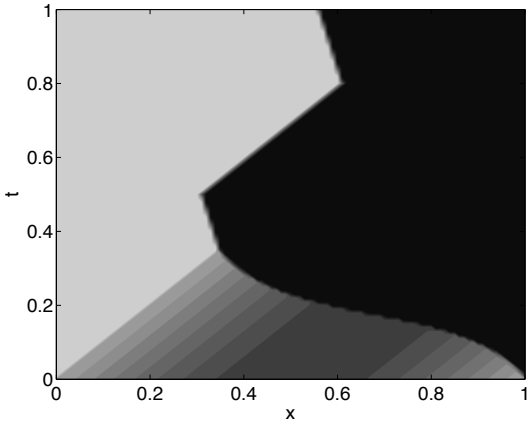
In Figure 1.11, we observe that the density is transported in a forward direction with velocity a. Since the outflow is zero for $0 \leq t < 0.5$, a congestion occurs resulting in a backward traveling shock wave with maximal density ρ_{max}. The shape of the latter shock wave is according to the nonlinear initial condition $\rho_0(x)$. For the time $0.5 \leq t < 0.8$, the congestion is released, i.e., the outflow is not zero anymore. All density ρ moves with velocity a in a positive direction. After time $t \geq 0.8$, the outflow is blocked again and a new jam arises.



(a) Wave Front Tracking



(b) DFG, $\Delta x = 1/20$



(c) DFG, $\Delta x = 1/100$

Figure 1.11: Shock motion of a multiple Riemann problem: Wave Front Tracking (above), DFG method with colored density values (bottom).

1.5.2 DFG method vs. RFG method

In the following, we present numerical results to validate and compare the Discontinuous Flux Godunov Method to a classical Godunov method for the regularized

problem. I.e., on the one hand, we apply the DFG Method directly to the discontinuous conservation law (1.11), (1.13) and on the other hand, we present solutions of the standard Godunov scheme for the regularized problem (1.14) calling it the Regularized-Flux-Godunov method (RFG). As already mentioned, the RFG scheme will need very small time steps in the limit $\delta \to 0$ to produce qualitatively good solutions. We compare the solution of both numerical methods for different scenarios. In particular, we are interested in the accuracy, efficiency and performance of the numerical approaches.
For an example, we present a comparison of computing times between the DFG Method for the discontinuous problem and the RFG scheme for the regularized one. We stick to the unit interval and fix Δx and $a = 1$. Then, the CFL condition leads to time step sizes of range $\Delta t = \Delta x$ for the DFG method and accordingly $\Delta t = |\delta|\Delta x$ for the RFG scheme, if possible Case (B.) waves are not too fast. We assume an inflow at the left boundary of $f(\rho(0,t)) = 0.4$ for all times $t > 0$, an outflow at the right boundary of $f(\rho(1,t)) = 0$ for $t \leq 2$ (blocking) and $f(\rho(1,t)) = a \cdot \rho(1,t)$ afterwards (release). The system is empty at the beginning, i.e., $\rho(x,0) = 0$. The total time-horizon is $T = 3$. Additionally to the CPU times, we compute the L1-error as the difference between the density either computed by the DFG method $\rho_{DFG}(x,t)$ or the RFG scheme $\rho_{RFG}(x,t)$:

$$\int_0^T \int_0^1 |\rho_{DFG}(x,t) - \rho_{RFG}(x,t)| \, dx \, dt.$$

The computation times are listed in Table 1.1.

Level	Grid size Δx	DFG	RFG $\delta = 0.1$	RFG $\delta = 0.01$	L1-error $\delta = 0.1$	L1-error $\delta = 0.01$
1	0.1	0.0004	0.0602	0.5847	0.1893	0.2033
2	0.05	0.0014	0.2384	2.3054	0.1280	0.1254
3	0.01	0.0376	5.7347	56.9094	0.0800	0.0443
4	0.001	2.8894	573.5482	5710.9	0.0668	0.0156

Table 1.1: CPU times in seconds and error comparison of the DFG and the RFG method with different regularization parameters and space grid sizes.

Let us switch to an analysis of our numerical approaches, see Figure 1.12. We compare our numerical solutions with Riemann problems discussed in Section 1.2.4 for the discontinuous problem (1.13) and analytical solutions presented in [27] and [4] for the regularized model (1.14). The computational setting above indicates different regimes ($\delta = 0.1$ and $\Delta x = 10^{-2}$):

1. The simulation starts and a forward traveling shock with speed $s = a$ is running through the system, cf. Figure 1.12 at time $t = 0.5$. The DFG

method yields an exact representation of the shock while the RFG scheme smears the initial discontinuity.

2. The system is blocked, cf. Figure 1.12 at time $t = 1.25$. The resulting solution is a shock wave traveling with $s = \frac{f(\rho_l)}{\rho_l - \rho_{\max}}$, i.e., $s = \frac{2}{3}$. Here, both numerical schemes yield the same numerical solution due to the choice of the parameters λ and δ. This is usually not valid for all configurations of parameters and different cases of Riemann problems.

3. The system is released, cf. Figure 1.12 at time $t = 2.05$, but the numerical solutions differ widely: the DFG method treats the influence of the zero wave correctly, whereas the RFG method makes a mistake induced by the regularization $f_\delta = \min\{\rho, \frac{1}{\delta}(1-\rho)\}$. For the RFG method we observe the wave which is traveling with speed $s = -\frac{1}{\delta}$ to the left, compare the discussion in Subsection 1.2.4. We note that, the difference between the dashed and the blue line will vanish for $\delta \to 0$.

4. The congestion starts to clear, cf. Figure 1.12 at time $t = 2.1$. The two waves computed by the Godunov scheme with regularization interact. Now, both numerical methods lead to forward traveling shocks with $s = a$. However, the shock locations are different due to the different history of the two solutions.

1.5.3 Microscopic Model vs. Continuous Model

We compare the results of the microscopic model in Section 1.1 against the continuous model in Section 1.2. In this scenario, the parts move with velocity $a = 1$. The spatial domain is restricted to the unit interval $[0, 1]$. At starting time $t = 0$, no part is located in the system. Therefore, the parts spawn at the left boundary $x = 0$ to each time $t = 0, 0.25, 0.5, ...$ a.s.o. Also, the minimal distance is set to $H_0 = 0.1$. The right boundary is blocked for $t \leq 2$, i.e., parts cannot pass $x = 1$. Furthermore, they change immediately their velocity to zero. After $t > 2$, the parts can pass the right boundary, i.e., the velocity of parts at $x = 1$ changes their velocity to $a = 1$.

Now we transfer this setting to the continuous model. However, we select the ratio of the total volume and the total amount of parts $\Delta Y = 0.1$, see Section 1.1. Thus, one obtains a maximal density $\rho_{max} = 1$ and a inflow density $\rho(0, t) = 0.4$. Additionally, the outflow is $f(\rho(1, t)) = 0$.
The microscopic model is based on an ODE-system, which is computed by the explicit Euler method for a step size 10^{-3}. The continuous model is computed by the DFG method with step sizes $\Delta t = \Delta x = 0.01$. The results are shown in Figure 1.13. The parts move with constant velocity into the domain. The first

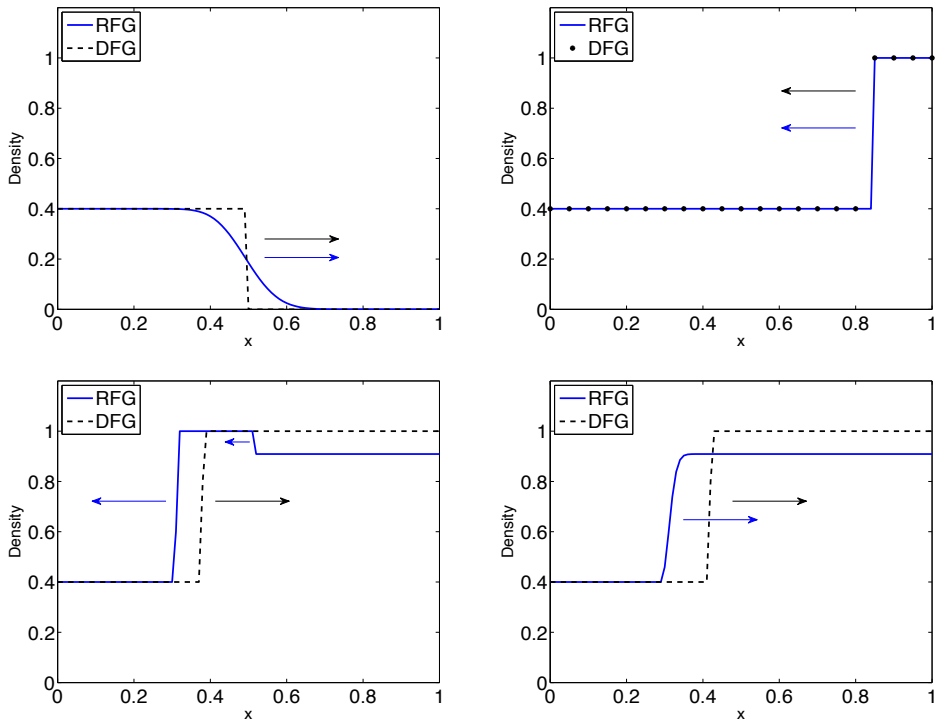


Figure 1.12: Comparison of solutions at times $t = 0.5$ and $t = 1.25$ (first row), $t = 2.05$ and $t = 2.1$ (second row).

part stops immediately if it reaches the point $x = 1$ for $t \leq 2$. The succeeding parts changes their velocity to zero if the distance to the predecessor becomes H_0. Thus, we recognize a tailback situation. Afterwards $t > 2$, all parts move with velocity $a = 1$.

1.5.4 Optimal Inflow

Our main objective, however, is to find the optimal inflow $f(\rho(0,t))$ such that congestions are avoided. To do so, we try to keep the buffers as small as possible, i.e., we minimize the sum over the density $\sum_{i,n} \rho_i^n$, cf. Section 1.4.1. We investigate the following scenario. Let us assume a total supply of $S = 4$, a production velocity of $a = 1$ and a final time $T = 15$. The production system is mapped onto the unit interval $[0, 1]$ and the last machine is stopped for maintenance in time $t \in [4, 8) \cup [11, 15)$.

$$f_{out}(t) = \begin{cases} 0 & \text{if} \quad t \in [4,8) \cup [11,15), \\ 1 & \text{otherwise.} \end{cases}$$

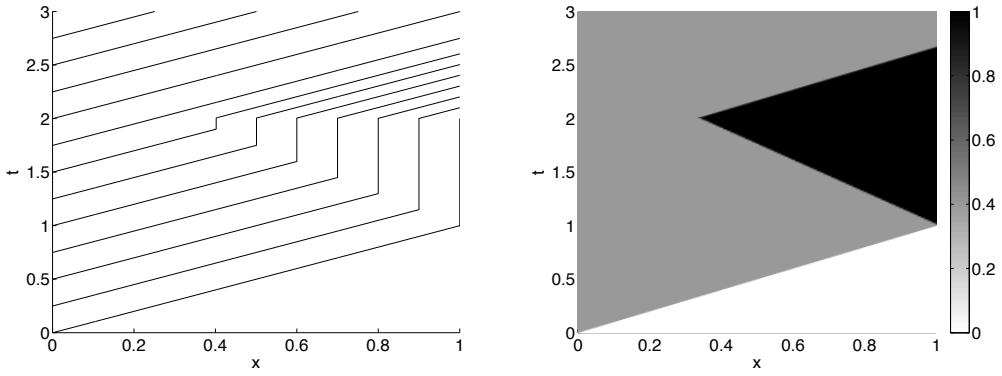


Figure 1.13: Trajectories of the microscopic model (left), part density computed by the DFG method (right)

We restrict the inflow to the upper bound $W = 0.5$. For the space-time grid we take $N = 10$ cells and $N_T = 150$ time points. The adjoint approach is implemented with a smoothness parameter $\epsilon = 10^{-2}$. The termination criterion is selected for a tolerance $TOL = 10^{-6}$ with

$$\Delta t \sum_{n=1}^{N_T} (\Delta t d_{(k)}^n)^2 < TOL,$$

where $d_{(k)}^n$ is the steepest descent direction to the k-th iteration. All optimization results are plotted in Figure 1.14, 1.15 and 1.16.

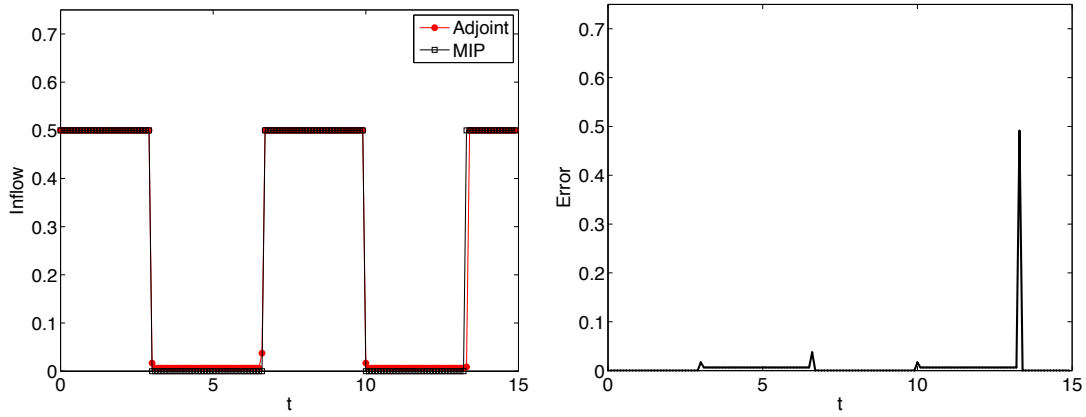


Figure 1.14: Optimal inflow profile of the adjoint approach and the MIP model (left). Absolute error of both models (right).

Figure 1.14 indicates that both methods yield the same result. More precisely, the results are independent of the underlying solution technique. The projected

gradient method used for the adjoint approach as well as the Branch-and-Bound solver [71] behaves in the same way. Only small differences can be identified, i.e., there is a delay of 10 discrete time steps in the second interval of maintenance.

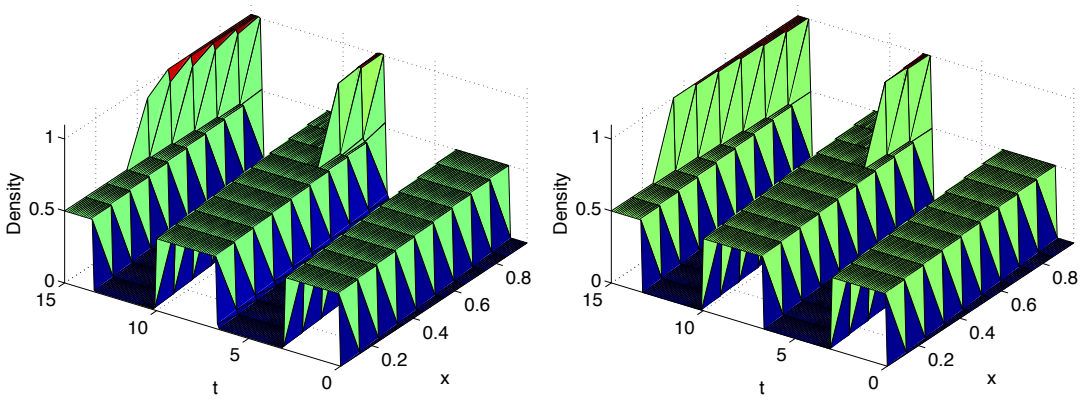


Figure 1.15: Density profile: Adjoint calculus (left), MIP model (right).

The corresponding optimal densities are shown in Figure 1.15. Since the optimal inflow tries to prevent congestions 1.14 by reducing the inflow to zero before the maintenance intervals are scheduled we naturally observe a similar behavior in the evolution of the density. Looking at Figure 1.16, we see that there is no outflow during the maintenance intervals. Hence the whole system is blocked. Due to the relation $F_N^n \leq f_{out}(t)$, the outflow is adjusted as soon as the system is released again. As already shown, the density variables of the adjoint approach differ slightly from the density variables of the MIP model due to the smooth approximation of the min-function for the adjoint approach. This effect is particularly apparent in the the optimal cost functional values, i.e., $J^*(\rho_i^n) = 393.21$ for the adjoint approach and $J^*(\rho_i^n) = 396$ for the MIP model.

Computation times

We repeat the previous example with a different number of discrete cell points N and measure the computation times. The goal is to compare the efficiency of the different optimization approaches from a practical point of view. Table 1.2 and Table 1.3 stress the usability of the adjoint approach. Finer resolutions lead to a moderate increase of the computation times. In contrast, the MIP model without any acceleration takes approximately one hour to solve the $N = 20$ instance. By additionally including constraint (1.33), the solution of the MIP model can be speed up significantly. There is a certain threshold where the adjoint approach dominates the improved MIP.

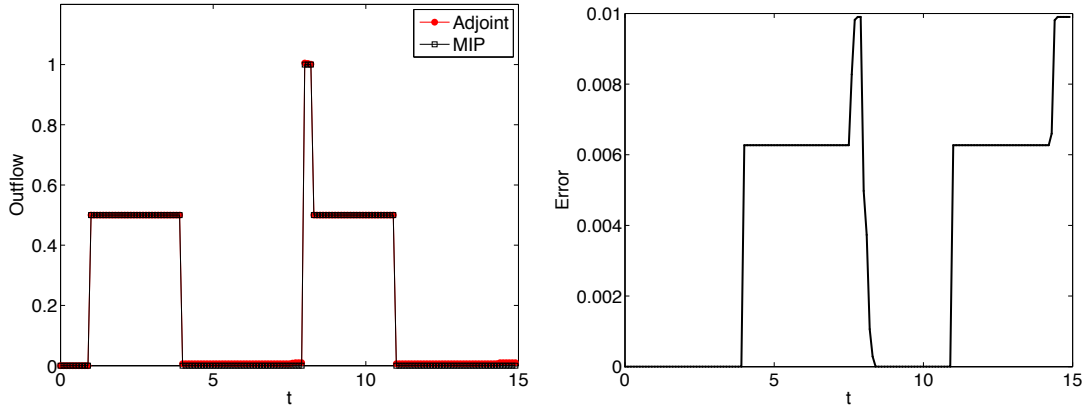


Figure 1.16: Outflow profile F_N^n of the adjoint approach and the MIP model (left). Absolute error of both models (right).

N	Adjoint	MIP (CPLEX)	improved MIP (CPLEX)
10	4.11s	0.45s	0.53s
15	16.95s	36.71s	1.99s
16	22.07s	188.17s	2.52s
20	44.26s	3612.53s	96.01s

Table 1.2: Computation times in seconds for the MIP, improved MIP and adjoint approach.

Number of	$N = 10$	$N = 15$	$N = 16$	$N = 20$
Variables (MIP)	3160	6990	7936	12320
Binaries (MIP)	1500	3375	3840	6000
Constraints (MIP)	9011	20266	23057	36021
Constraints (improved MIP)	10361	23416	26657	41721

Table 1.3: Number of Variables and Constraints of the MIP model.

Accuracy of the gradient

The adjoint approach is useful to compute gradient informations efficiently. In this test case, we compare the gradients of the adjoint approach (1.29) with finite differences. The start control vector is feasible and constant, i.e., $u^n = F_0^n = 4\frac{1}{\Delta t \cdot N_T}$ for all $n = 1, \ldots, N_T$. We formally denote the solution operator of the forward problem for a fixed u by $G(u) = (\rho, F)$. Thus, the gradient can be approximated by central finite differences

$$\partial_{u^n} J(G(u), u) \approx \frac{1}{2\delta}(J(G(u+\delta), u+\delta) - J(G(u-\delta), u-\delta)),$$

for $\delta \to 0$. A numerical comparison of the gradients for $\delta = 10^{-4}$ is given in Figure 1.17. The error of magnitude is 10^{-8} and therefore satisfactory.

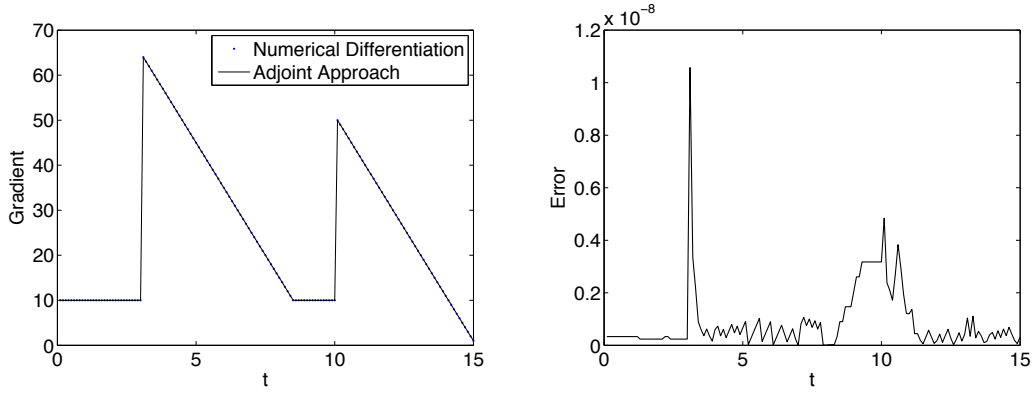


Figure 1.17: Gradients computed by numerical differentiation and adjoint approach (left). Error of both gradients (right).

1.5.5 Machine shut-down for maintenance

The last machine of a production line needs a maintenance over 3 time-units. The total simulation time is $T = 15$. The production inflow is set to

$$f(\rho(0,t)) = 0.5\sin(0.2\pi t) + 0.5.$$

The initial state of the density is $\rho(x,0) = 0.5$.
Also, the spatial interval [0,1] is discretized into $N = 10$ cells. The time interval is divided into $N_T = 150$ points. The duration of the maintenance is set to $N_{off} = 30$. The optimization task is to minimize the density on the whole time and space, i.e., $\sum_{i,n} \rho_i^n$.

The results are shown in Figure 1.18. The maintenance starts in time $t = 7.5$ and ends in time $t = 10.5$, consider Figure 1.18 (right). During the maintenance, the outflow becomes zero and the result is a tailback in form of a back traveling shockwave. However, the choice of that date keeps the buffers as small as possible. The objective value is minimized to $J^* = 940.13$. Especially, the propagation of the tailback is plotted in Figure 1.19. During the times $t = 7.5$ to $t = 10.5$, the outflow F_N^n is zero. This situation causes a back traveling shockwave, cf. Figure 1.19 (left). After time $t = 10.5$, the congestion is released, cf. Figure 1.19 (right). In contrast to an unoptimized solution, we select the maintenance date to $t = 5.5$. The corresponding objective value leads to $J = 1113.58$ and the resulting outflow is plotted in Figure 1.20. The corresponding tailback is quite larger than in the optimized solution.

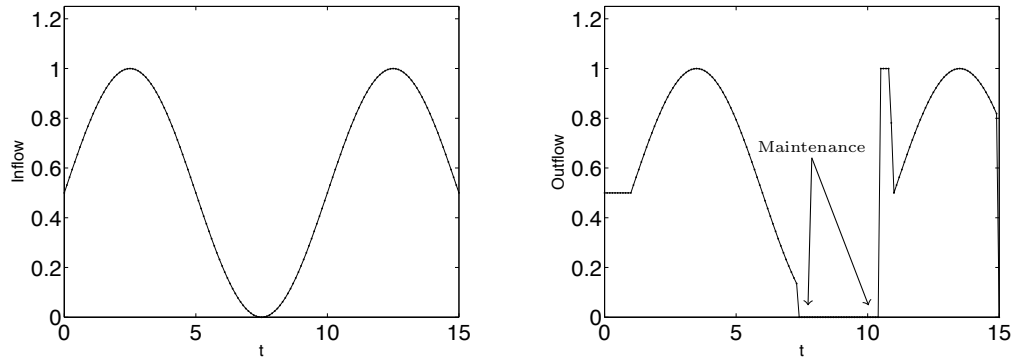


Figure 1.18: Inflow profile is given (left), Outflow profile computed by the MIP model (right)

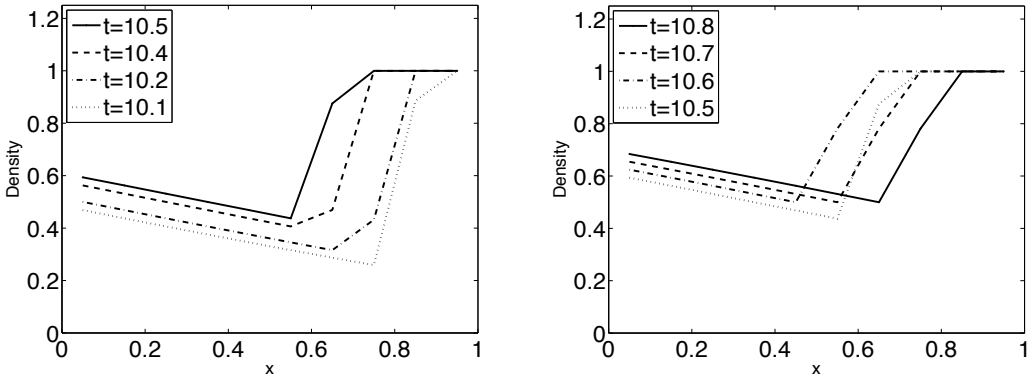


Figure 1.19: Propagation of the tailback by maintenace (left), release of the tailback (right)

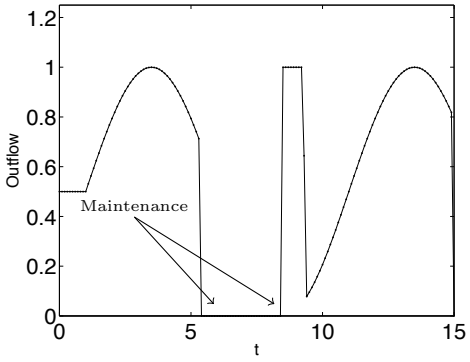


Figure 1.20: Outflow profile; maintenance starts at $t = 5.5$.

Chapter 2

Network Extension

A manufacturing system is organized as a production line consisting of machines where each machine is responsible for certain production stages. Often enough, the material flow is distributed to several production units. In particular, this is useful if certain production stages take a long time, and thus the production stage is accomplished simultaneously by several machines for reducing production time. This scenario leads to the concept of production and supply-chain networks.

In this chapter, the model that is already introduced in Chapter 1 is extended to network topologies. In a mathematical sense, conservation laws are considered on a network structure, and therefore it is necessary to find suitable coupling conditions on the network junctions. In the last decade, similar models in different applications are investigated and extended to network structures; for example, continuous traffic flow models, gas and water networks. We refer to [8, 19, 27, 35, 38, 39, 60, 62, 64, 68, 98] for an overview. Network models for supply and manufacturing systems which are based on conservation laws are found in [26, 28, 37, 44, 45, 58, 75].

In applications, optimization issues for production networks play an important role. Therefore, we introduce an optimization model for the network extension which is based on a mixed integer program model (MIP). Continuous network models and their reformulation to discrete optimization problems are also investigated, for example, [33, 37, 43, 106]. The main drawback of MIPs is the enormous computation time for a huge system with a high number of variables. One option for reducing the computation time and its complexity is preprocessing or presolving approach. Thereby, the preprocessing routine is called before the MIP solving process (e.g. branch and bound) starts. Preprocessing routines for general MIPs can be found in [2, 13, 23, 73, 92, 95]. In particular, known PDE structures and informations is useful to obtain efficient preprocessing routines for MIP models with PDE constraints, e.g. [33]. In this work, we present preprocessing routines for the underlying network model for discontinuous conservation laws.

In Section 2.1, we introduce a network coupling for the model of Section 1.2. The coupling approach for the regularized model based on the approach of Coclite-Garavello-Piccoli (CGP) [19]. This approach is briefly introduced in Subsection

2.1.1. In Subsection 2.1.2, we derive a network model for a transport equation with a discontinuous flux function. Then, a solution algorithm for the network extension is presented in Section 2.2. Afterwards, in Section 2.3, we derive an optimization approach based on the mixed integer programming model. For decreasing computation times of the MIP model, we introduce presolving techniques in Section 2.4. Finally, in Section 2.5, numerical results are presented.

2.1 Network Model Approach

In this section, we establish coupling conditions for a network model for the discontinuous conservation law of Section 1.2

$$\partial_t \rho + \partial_x f(\rho) = 0, \quad \rho(x,0) = \rho_0(x), \tag{2.1}$$

with flux function

$$f(\rho) = a\rho H(\rho_{max} - \rho). \tag{2.2}$$

At first, we give a definition of a network and introduce the basic notations used throughout of this chapter.

Definition 2.1.1 (Network defintion)**.** *A network is given as a directed graph $G = (V, E)$.*

- *V denotes the set of all vertices or junctions in a network. E is the set of edges.*
- *The function $\alpha : E \to V$ maps each edge to its starting point, and the function $\omega : E \to V$ maps each edge to its endpoint.*
- *The set of all incoming edges of $v \in V$ is denoted by $\delta_v^{in} := \{e \in E : \omega(e) = v\}$, and $\delta_v^{out} := \{e \in E : \alpha(e) = v\}$ is referred to as the set of all out coming edges of v for all junctions in V, cf. Figure 2.1.*
- *Let $\tilde{E} \subset E$ be an arbitrary subset of E. The set of all incoming edges in $\tilde{E}$ is defined as $\delta^{in}(\tilde{E}) := \{e \in E \setminus \tilde{E} : \exists \tilde{e} \in \tilde{E}$ with $\alpha(\tilde{e}) = \omega(e)\}$. The set of all outgoing edges of $\tilde{E}$ is defined as $\delta^{out}(\tilde{E}) := \{e \in E \setminus \tilde{E} : \exists \tilde{e} \in \tilde{E}$ with $\omega(\tilde{e}) = \alpha(e)\}$, cf. Figure 2.2.*
- *The set of all incoming edges of the network is given by $E^{in} := \{e \in E : \delta^{in}(\{e\}) = \emptyset\}$. Each element of $E^{in} \subset E$ is called inflow edge.*
- *The set of all outgoing edges of the network is given by $E^{out} := \{e \in E : \delta^{out}(\{e\}) = \emptyset\}$. Each element of $E^{out} \subset E$ is called outflow edge.*
- *Each edge is modeled by an interval $[a_e, b_e]$ with a length $L_e := |b_e - a_e|$.*

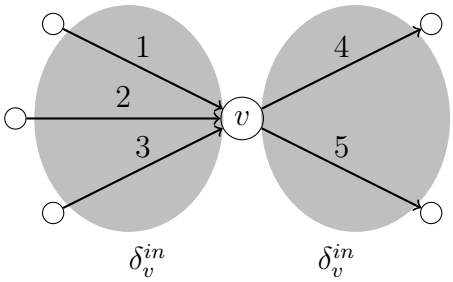


Figure 2.1: Junction v with incoming edges $e \in \delta_v^{in} = \{1, 2, 3\}$ and outgoing edges $e \in \delta_v^{out} = \{4, 5\}$.

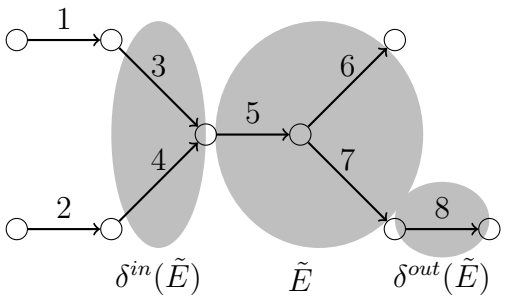


Figure 2.2: Given is a set of edges $\tilde{E} = \{5, 6, 7\}$. The incoming edges of $\tilde{E}$ is $\delta^{in}(\tilde{E}) = \{3, 4\}$. The outgoing edges of $\tilde{E}$ is $\delta^{out}(\tilde{E}) = \{8\}$

To simplify the notation, we choose $\rho_{max} = 1$ and $a = 1$. We review the coupling conditions for the regularized type of the conservation law, see [19, 27]. We approximate the flux f as before by the continuous function f_δ for $\delta > 0$ with

$$f_\delta(\rho) = \min\{\rho, \frac{1}{\delta}(1 - \rho)\} \quad \text{for } \delta > 0. \tag{2.3}$$

Here, the density value for the maximal flow is given by

$$\sigma := \arg\max_{\rho \in [0,1]} f_\delta(\rho) = \frac{\rho_{max}}{1 + a\delta}. \tag{2.4}$$

We consider the network problem

$$\begin{aligned} &\partial_t \rho_e(x,t) + \partial_x f_\delta(\rho_e(x,t)) = 0 \quad \forall e \in E, x \in (a_e, b_e), \quad t \geq 0, \\ &\rho_e(x,0) = \rho_{e,0}(x) \quad \forall x \in (a_e, b_e). \end{aligned} \tag{2.5}$$

For a definition of weak network solutions and Riemann problems at junctions fulfilling the equality of fluxes, we refer to [68]. We denote the Riemann initial data with $\rho_{e,0} = \rho_{e,0}(b_e)$ for incoming edges and $\rho_{e,0} = \rho_{e,0}(a_e)$ for outgoing edges for a single junction. Assuming a unique solution for the problem at the junction, we denote the solution at the junction, i.e., at $x = b_e$ for incoming and at $x = a_e$ for outgoing edges, by

$$(\bar{\rho}_1, ..., \bar{\rho}_{n+m}).$$

2.1.1 The Approach of Coclite-Garavello-Piccoli (CGP)

We consider a junction with n incoming edges and m outgoing edges labeled by $e = 1, \ldots, n$ and $e = n + 1, \ldots, m + n$, respectively (cf. Figure 2.3).

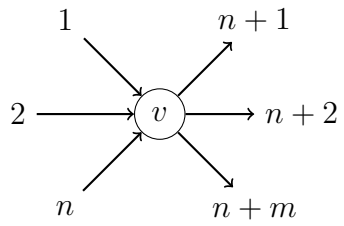


Figure 2.3: A junction with n incoming and m outgoing edges.

Given the constant initial values $\rho_{e,0}$, we need to determine a unique solution $\bar{\rho}_e$ satisfying the coupling condition. In particular for the piecewise differentiable flux function (1.14) we refer to [27]. The values of $\bar{\rho}_e$ are restricted as follows:

$$\begin{aligned} \bar{\rho}_e &\in [\sigma, 1], & \rho_{e,0} &\geq \sigma, & e &= 1, ..., n, \\ \bar{\rho}_e &\in \{\rho_{e,0}\} \cup (\tau(\rho_{e,0}), 1], & \rho_{e,0} &\leq \sigma, & e &= 1, ..., n, \\ \bar{\rho}_e &\in [0, \sigma], & \rho_{e,0} &\leq \sigma, & e &= n+1, ..., n+m, \\ \bar{\rho}_e &\in [0, \tau(\rho_{e,0})) \cup \{\rho_{e,0}\}, & \rho_{e,0} &\geq \sigma, & e &= n+1, ..., n+m, \end{aligned} \tag{2.6}$$

where for each $\rho \neq \sigma, \rho \in [0, 1]$ the value $\tau(\rho)$ is the unique number $\tau(\rho) \neq \rho$ such that $f(\rho) = f(\tau(\rho))$. Thus $\rho < \sigma \Rightarrow \tau(\rho) > \sigma$ and vice versa. Here

$$\tau(\rho) = \begin{cases} \frac{1}{\delta}(1 - \rho) & \rho > \sigma \\ 1 - \delta\rho & \rho \leq \sigma \end{cases} \tag{2.7}$$

Next, we look for suitable coupling conditions for (2.1) with linear flux (2.2). We proceed as for the Riemann problems in Section 1.2. The coupling conditions for the discontinuous problem are obtained by using the CGP-approach for the regularized problem (2.3) and considering the limit solutions for small δ. We review the approach in the regularized case considering only two types of junctions, the first one having two incoming and one outgoing edge and the second one having one incoming and two outgoing edges, see Figure 2.4.

Coupling conditions for two incoming edges and one outgoing edge

We consider a junction with two incoming edges $n = 2$ and one outgoing edge $m = 1$. The initial densities on edges $e = 1, 2, 3$ are given by $\rho_{1,0}, \rho_{2,0}, \rho_{3,0}$. The corresponding fluxes are denoted as $\gamma_{e,0} = f_\delta(\rho_{e,0})$. Denote the maximum of the flux by $f_\delta(\sigma)$. We denote the sets of valid resulting fluxes γ_e by Ω_e. For the incoming edges $e = 1, 2$ this is

$$\begin{aligned} \rho_{e,0} \leq \sigma &\Rightarrow \Omega_e = [0, \gamma_{e,0}], \\ \rho_{e,0} \geq \sigma &\Rightarrow \Omega_e = [0, f_\delta(\sigma)]. \end{aligned} \tag{2.8}$$

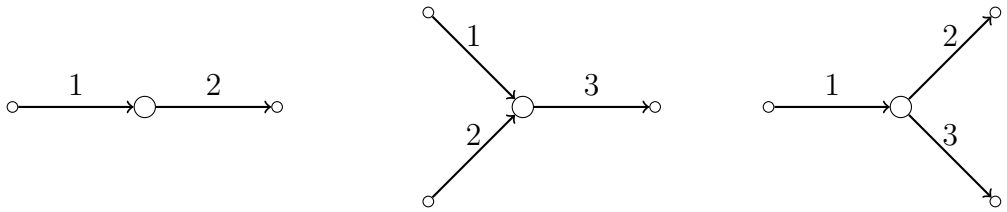


Figure 2.4: A junction with two connected edges (left), a junction with two incoming and one outgoing edge (middle) and a junction with one incoming and two outgoing edges (right).

For the outgoing edge $e = 3$,

$$\begin{aligned} \rho_{e,0} \leq \sigma \quad &\Rightarrow \quad \Omega_e = [0, f_\delta(\sigma)], \\ \rho_{e,0} \geq \sigma \quad &\Rightarrow \quad \Omega_e = [0, \gamma_{e,0}]. \end{aligned} \tag{2.9}$$

Moreover we can define c_e such that

$$\Omega_e = [0, c_e]. \tag{2.10}$$

The fluxes at the junction are found in the following way, see e.g. [54]:

$$\begin{aligned} &\max \quad w\gamma_1 + \gamma_2 \quad \text{w.r.t.} \\ &0 \leq \gamma_1 \leq c_1, 0 \leq \gamma_2 \leq c_2, \gamma_1 + \gamma_2 \leq c_3, \end{aligned} \tag{2.11}$$

where $w > 1$ is a weight for the maximization problem. The unique solution is $\gamma_1 = \min\{c_1, c_3\}$, $\gamma_2 = \min\{c_2, c_3 - \gamma_1\}$, $\gamma_3 = \gamma_1 + \gamma_2$.

Remark 2.1.2. *In the work of Göttlich et al. [48], an alternative maximization problem is used in this situation, i.e.:*

(1) $c_1 + c_2 \leq c_3$*: Then we have to look for* γ_1, γ_2 *such that*

$$\begin{aligned} &\max \gamma_1 + \gamma_2 \quad \textit{w.r.t.} \\ &0 \leq \gamma_1 \leq c_1, 0 \leq \gamma_2 \leq c_2, \gamma_1 + \gamma_2 \leq c_3. \end{aligned}$$

(2) $c_1 + c_2 > c_3$ *: Then we have to look for* γ_1, γ_2 *such that*

$$\begin{aligned} &\max \gamma_1 + \gamma_2 \quad \textit{w.r.t.} \\ &\gamma_1 = \gamma_2 \\ &0 \leq \gamma_1 \leq c_1, 0 \leq \gamma_2 \leq c_2, \gamma_1 + \gamma_2 \leq c_3. \end{aligned}$$

For reducing the complexity of the presented network problem and the optimization problem in Section 2.3, we select the simpler maximization problem (2.11).

Coupling conditions for one incoming edge and two outgoing edges

We consider a junction with one incoming edge $n = 1$ and two outgoing edges $m = 2$. We use the same notation as before, i.e., we define $\gamma_{e,0}$ and the sets Ω_e depending on whether incoming or outgoing edges are considered. Using distribution rates $d_{2,1}, d_{3,1} \in (0,1)$ with $d_{2,1} + d_{3,1} = 1$ the CGP-conditions are

(1) $\gamma_1 \in \Omega_1, d_{e,1}\gamma_1 \in \Omega_e$ for $e = 2, 3$.

(2) Maximize γ_1 w.r.t. (1).

(3) $\gamma_j = d_{e,1}\gamma_1,\ e = 2, 3$.

Using $\Omega_e = [0, c_e], e = 1, 2, 3$, we obtain

$$\gamma_1 = \min\{c_1, c_2/d_{2,1}, c_3/d_{3,1}\}. \tag{2.12}$$

This is exactly, what is known as the FIFO (first in, first out) rule of a dispersing junction.

Remark 2.1.3. *The situation of one incoming and one outgoing edge only (linear network) can be directly deduced from the dispersing case. In fact, in the limit for one distribution parameter, e.g. $d_{3,1} \to 0$, the solution of the* max$-$*problem reduces to*

$$\gamma_1 = \min\{c_1, c_2\}.$$

In the following, we determine the coupling conditions for the discontinuous conservation law (2.1) with linear flux (2.2) using the above approach for the regularized problem (2.3) and considering the limit $\delta \to 0$.

2.1.2 Network Coupling for the discontinuous Flux Function

At first, according to (2.6), we define admissible solutions at junctions for the regularized problem (2.3) for two types of junctions, cf. Figure 2.4. This is a straightforward transfer from the CGP-approach explained in Subsection 2.1.1. Second, we consider the limit $\delta \to 0$ and describe the resulting Riemann solutions at junctions.

Two incoming edges and one outgoing edge

We consider a junction with two incoming edges $n = 2$ and one outgoing edge $m = 1$. The initial densities on edges $e = 1, 2, 3$ are given by $\rho_{1,0}, \rho_{2,0}, \rho_{3,0}$. We note that if $0 \le \rho_{e,0} < 1$ for $e = 1, 2, 3$, then there exists a small $\delta > 0$ such that $\rho_{e,0} \le \sigma, e = 1, 2, 3$. Thus, in case $\rho_{e,0} < 1$ one obtains for δ small enough:

$$\begin{aligned} \bar{\rho}_e \in \{\rho_{e,0}\} \cup ((1 - \delta\rho_{e,0}), 1], \qquad e = 1, 2, \\ \bar{\rho}_3 \in [0, \sigma] \end{aligned} \tag{2.13}$$

and $c_1 = f_\delta(\rho_{1,0}), c_2 = f_\delta(\rho_{2,0}), c_3 = f_\delta(\sigma)$. In the limit $\delta \to 0$ this yields for $\rho_{e,0} < 1$:

$$\begin{aligned} \bar{\rho}_e \in \{\rho_{e,0}\} \cup \{1\}, \quad & \text{if } \rho_{e,0} > 0, \quad e = 1,2, \\ \bar{\rho}_e \in \{\rho_{e,0}\}, \quad & \text{if } \rho_{e,0} = 0, \quad e = 1,2, \\ & \bar{\rho}_3 \in [0,1] \end{aligned} \tag{2.14}$$

and $c_1 = \rho_{1,0}, c_2 = \rho_{2,0}, c_3 = 1$. Both situations are depicted in Figure 2.5 and 2.6.

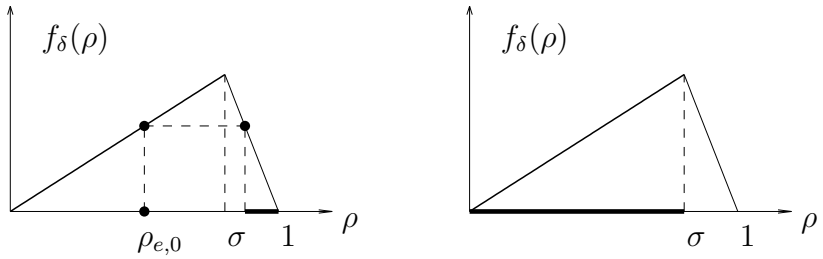


Figure 2.5: Feasible coupling densities for incoming edges $e = 1, 2$ (left) and outgoing edge $e = 3$ (right) in case of the regularized flux function (1.14) and $\rho_{e,0} < 1$.

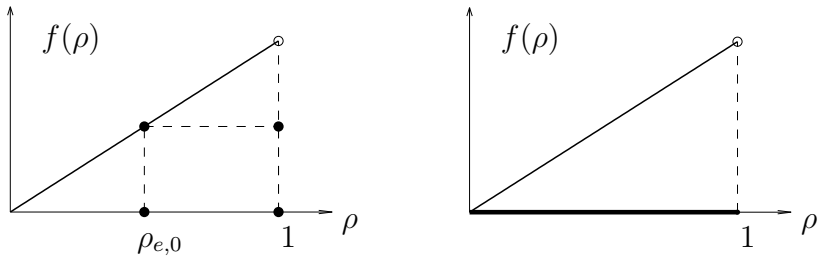


Figure 2.6: Feasible coupling densities for incoming edges $e = 1, 2$ (left) and outgoing edge $e = 3$ (right) in case of the discontinuous flux function (1.13) and $\rho_{e,0} < 1$.

Correspondingly, in the special case $\rho_{e,0} = 1$, one obtains

$$\begin{aligned} \bar{\rho}_e \in [\sigma, 1], \quad e = 1,2, \\ \bar{\rho}_3 \in \{\rho_{3,0}\} = \{1\} \end{aligned} \tag{2.15}$$

and $c_1 = f_\delta(\sigma), c_2 = f_\delta(\sigma), c_3 = f_\delta(\rho_{3,0})$. In the limit $\delta \to 0$, we end up with

$$\bar{\rho}_e \in \{1\}, \quad e = 1,2,3 \tag{2.16}$$

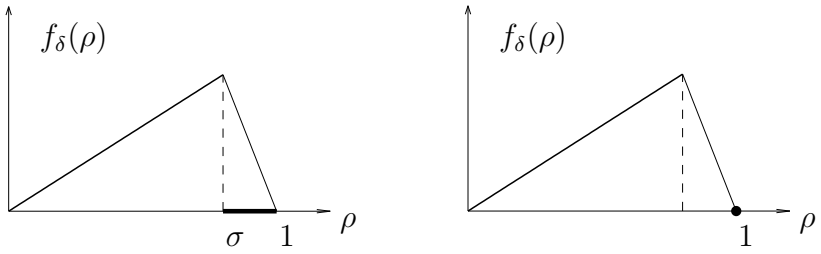


Figure 2.7: Feasible coupling densities for incoming edges $e = 1, 2$ (left) and outgoing edge $e = 3$ (right) in case of the regularized flux function (1.14) and $\rho_{e,0} = 1$.

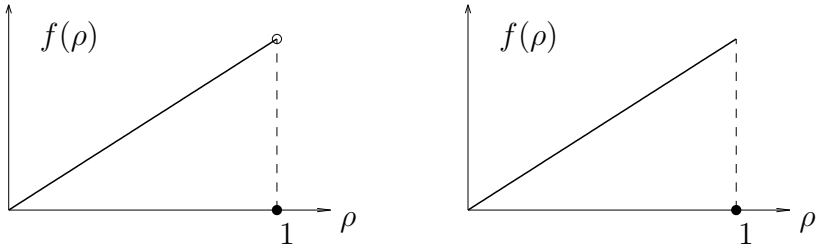


Figure 2.8: Feasible coupling densities for incoming edges $e = 1, 2$ (left) and outgoing edge $e = 3$ (right) in case of the discontinuous flux function (1.13) and $\rho_{e,0} = 1$.

and $c_1 = 1, c_2 = 1, c_3 = 0$, as illustrated in Figure 2.7 and 2.8.

Based on these results, we discuss three different cases that could occur when solving the flux maximization problem at nodes (cf. Subsection 1.2.4 and the model):

• **Case A1:** $0 \leq \rho_{e,0} < 1$

The limit (2.14) yields $c_1 = \rho_{1,0}, c_2 = \rho_{2,0}, c_3 = 1$. We have to distinguish the two cases $\rho_{1,0} + \rho_{2,0} \leq 1$ and $\rho_{1,0} + \rho_{2,0} > 1$. In case $\rho_{1,0} + \rho_{2,0} \leq 1$ we get $\gamma_1 = \min\{\rho_{1,0}, 1\} = \rho_{1,0}$, $\gamma_2 = \min\{\rho_{2,0}, 1 - \rho_{1,0}\} = \rho_{2,0}$ and $\gamma_3 = \rho_{1,0} + \rho_{2,0}$. Moreover, the densities are $\bar{\rho}_1 = \rho_{1,0}$, $\bar{\rho}_2 = \rho_{2,0}$ and $\bar{\rho}_3 = \rho_{1,0} + \rho_{2,0}$. The resulting solutions are constants on edge 1 and 2 and a shock wave with speed $s_3 = 1$ on edge 3.
In case $\rho_{1,0} + \rho_{2,0} > 1$, the limit yields $\gamma_1 = \min\{\rho_{1,0}, 1\} = \rho_{1,0}$, $\gamma_2 = \min\{\rho_{2,0}, 1 - \rho_{1,0}\} = 1 - \rho_{1,0}$ and $\gamma_3 = \gamma_1 + \gamma_2$. The resulting densities are $\bar{\rho}_1 = \rho_{1,0}$, $\bar{\rho}_2 = \bar{\rho}_3 = 1$. The resulting solutions is a constant on edge 1 and shock wave solutions on edge 2 and 3 with speed $s_2 = \frac{\rho_{2,0} - 1 + \rho_{1,0}}{\rho_{2,0} - 1}$ and $s_3 = 1$.

• **Case A2:** $\rho_{1,0} = 1,\ 0 \le \rho_{2,0} \le 1,\ 0 \le \rho_{3,0} < 1$

The limit yields in this case $c_1 = 1, c_2 = \rho_{2,0}, c_3 = 1$. We obtain $\gamma_1 = \min\{1, 1\} = 1$, $\gamma_2 = \min\{\rho_{2,0}, 1 - \gamma_1\} = 0$ and $\gamma_3 = \gamma_1 + \gamma_2 = 1$. The resulting densities are $\bar\rho_1 = 1, \bar\rho_2 = 1, \bar\rho_3 = 1$. For $\rho_{2,0} < 1$, the solutions are shock waves with speed $s_1 = -\infty$, $s_2 = \frac{\rho_{2,0}}{\rho_{2,0}-1}$ and $s_3 = 1$. In case of $\rho_{2,0} = 1$, the shock wave velocity on edge 2 changes to $s_2 = -\infty$.

• **Case A3:** $\rho_{1,0} < 1,\ \rho_{2,0} = 1,\ 0 \le \rho_{3,0} < 1$

The limit yields in this case $c_1 = \rho_{1,0}, c_2 = 1, c_3 = 1$. Here we obtain $\gamma_1 = \rho_{1,0}$, $\gamma_2 = 1 - \rho_{1,0}$ and $\gamma_3 = 1$. The resulting densities are $\bar\rho_1 = \rho_{1,0}$, $\bar\rho_2 = \bar\rho_3 = 1$. The solution of edge 1 is a constant. Solutions on edge 2 and 3 are shock solutions with velocity $s_2 = -\infty$ and $s_3 = 1$.

Remark 2.1.4. *All other cases have an outgoing edge with $\rho_{3,0} = 1$. They lead to zero fluxes γ_e and if $\rho_{e,0} > 0, e = 1, 2$*

$$\bar\rho_1 = 1,\ \bar\rho_2 = 1,\ \bar\rho_3 = 1.$$

One incoming edge and two outgoing edges

Now we consider a junction with one ingoing edge $e = 1$ and two outgoing edges $e = 2, 3$. We consider again different cases for the initial data of the Riemann problems at the junctions. In case $0 \le \rho_{e,0} < 1$ one obtains for δ small enough:

$$\begin{aligned} \bar\rho_1 \in \{\rho_{1,0}\} \cup ((1 - \delta\rho_{1,0}), 1], \\ \bar\rho_e \in [0, \sigma] \qquad e = 2, 3, \end{aligned} \tag{2.17}$$

and $c_1 = f_\delta(\rho_{1,0}), c_2 = f_\delta(\sigma), c_3 = f_\delta(\sigma)$. In the limit $\delta \to 0$ this gives for $\rho_{e,0} < 1$:

$$\begin{aligned} \bar\rho_1 \in \{\rho_{1,0}\} \cup \{1\}, \quad \text{if } \rho_{1,0} > 0, \\ \bar\rho_1 \in \{\rho_{1,0}\}, \quad \text{if } \rho_{1,0} = 0 \\ \bar\rho_e \in [0, 1] \qquad e = 2, 3, \end{aligned} \tag{2.18}$$

and $c_1 = \rho_{1,0}, c_2 = 1, c_3 = 1$. Correspondingly, in the special case $\rho_{e,0} = 1$ for all $e = 1, 2, 3$, one obtains

$$\begin{aligned} \bar\rho_1 \in [\sigma, 1], \\ \bar\rho_e \in \{\rho_{e,0}\} = \{1\}, \quad e = 2, 3, \end{aligned} \tag{2.19}$$

and $c_1 = f_\delta(\sigma), c_2 = f_\delta(\rho_{2,0}), c_3 = f_\delta(\rho_{3,0})$. In the limit $\delta \to 0$, we end up with

$$\bar\rho_e \in \{1\}, \quad e = 1, 2, 3 \tag{2.20}$$

and $c_1 = 1, c_2 = 0, c_3 = 0$. All situations described above are similar to the case *two incoming edges and one outgoing edge*. To get an impression how the feasible densities look like we refer to Figures 2.5, 2.6, 2.7 and 2.8.

For a detailed discussion of the maximization problem at nodes, we consider two different cases:

- **Case B1:** $0 \leq \rho_{e,0} < 1$

We use the limit equation (2.18). Then $\gamma_1 = \min\{\rho_{1,0}, \frac{1}{d_{2,1}}, \frac{1}{d_{3,1}}\} = \rho_{1,0}$ and $\gamma_2 = d_{2,1}\rho_{1,0}$ and $\gamma_3 = d_{3,1}\rho_{1,0}$. The resulting densities are $\bar{\rho}_1 = \rho_{1,0}$, $\bar{\rho}_2 = d_{2,1}\rho_{1,0}$ and $\bar{\rho}_3 = d_{3,1}\rho_{1,0}$. The solution is constant on edge 1 and shock waves with speed 1 on edge 2 and 3.

- **Case B2:** $\rho_{1,0} = 1, 0 \leq \rho_{2,0} < 1, 0 \leq \rho_{3,0} < 1$

Then $\gamma_1 = \min\{1, \frac{1}{d_{2,1}}, \frac{1}{d_{3,1}}\} = 1$ and $\gamma_2 = d_{2,1}$ and $\gamma_3 = d_{3,1}$. The resulting densities are $\bar{\rho}_1 = 1$, $\bar{\rho}_2 = d_{2,1}$ and $\bar{\rho}_3 = d_{3,1}$. The solution is a shock wave with infinite negative speed on edge 1 and speed 1 on edge 2 and 3.

Remark 2.1.5. *All other cases have an outgoing edge with $\rho_{2,0} = 1$ or $\rho_{3,0} = 1$. These cases lead to zero fluxes γ_e and if $\rho_{1,0} > 0$*

$$\bar{\rho}_1 = 1,\ \bar{\rho}_2 = 1,\ \bar{\rho}_3 = 1.$$

Note that for the limiting case, the solution of edges at a junction may depend immediately on other edges/junctions due to the infinite speed of propagation.

2.2 Solution Algorithm

Having a complete network formulation for the problem (2.1), (2.2) at hand, we are now concerned with suitable numerical solution procedures. A conventional way for solving this problem is the regularization of the discontinuous flux and the use of classical schemes for hyperbolic conservation laws, see [4]. In the following, we introduce a network extension of the discontinuous flux Godunov method, which is introduced in Subsection 1.3.3.
The following network notation corresponds to Definition 2.1.1. We note that the algorithm is based on a finite volume method, i.e., the domain of an edge is divided equidistantly into N cells. Each cell is labeled by i, possesses a width of $\Delta x = L^e/N$ and contains the interval $[x_{i-1}, x_i]$ with $x_i = i\Delta x$. For a given time horizon T we introduce an equidistant time-grid with Δt as the time step-size, N_T the total number of time steps and discrete time points $t_n = n\Delta t$. Then, the

discretized density $\rho_i^{n,e}$ complies the averaged numerical solution in the cell i on edge e at time t_n. The CFL condition reduces to $\Delta t \leq a\Delta x$ and the computation of $\rho_i^{n+1,e}$ at the next time-level $n+1$ obeys the recursive formula:

$$\text{(PDE):} \qquad \rho_i^{n+1,e} = \rho_i^{n,e} - \lambda[F_i^{n,e} - F_{i-1}^{n,e}], \tag{2.21}$$

where $\lambda = \frac{\Delta t}{\Delta x}$ defines the grid constant. The numerical flux is given by

$$\text{(FLUX):} \qquad F_{i-1}^{n,e} = \min\{a\rho_{i-1}^{n,e}, \frac{\rho_{max} - \rho_i^{n,e}}{\lambda} + F_i^{n,e}\}. \tag{2.22}$$

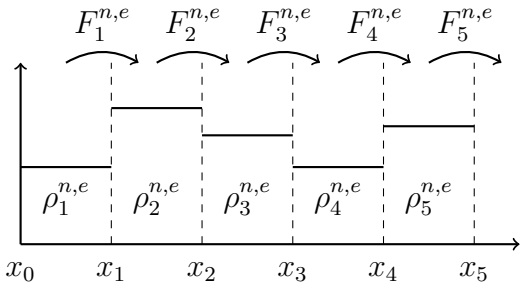


Figure 2.9: Illustration of the finite volume method.

This means, each cell $\rho_i^{n,e}$ has an inflow $F_{i-1}^{n,e}$ and an outflow $F_i^{n,e}$. As stated in (2.22), the flux $F_{i-1}^{n,e}$ is determined by the minimum of the straightforward Upwind flux $a\rho_i^{n,e}$ and the maximal possible flow. Hence, $F_{i-1}^{n,e}$ is based on a bounded Upwind flux such that $\rho_i^{n+1,e}$ never exceeds ρ_{max}.

Remark 2.2.1. *Since the maximal density ρ_{max} on each edge implies an upper bound for $\rho_i^{n,e}$, i.e., $0 \leq \rho_i^{n,e} \leq \rho_{max}$, we also get an upper bound for the numerical flux $F_i^{n,e}$:*

$$0 \leq F_i^{n,e} \leq a\rho_{max}. \tag{2.23}$$

Furthermore, we assume that no quantities are lost or generated at junctions. Thus, in the sense of mass conservation, the sum of all ingoing fluxes is equal to the outgoing ones.

$$\text{(CPL):} \qquad \sum_{e \in \delta_v^{in}} F_N^{n,e} = \sum_{e \in \delta_v^{out}} F_0^{n,e} \tag{2.24}$$

Considering different types of junctions and following the discussion in Section 2.1, we show how to set the numerical flux $F_N^{n,e}$ explicitly, cf. Figure 2.9. This is an important issue since the correct solution for the maximization problem at junctions must be ensured. According to Figure 2.4, we review all potential

scenarios and add a further situation (Junction Type IV) to tackle network sinks.

Junction Type I. We consider a junction with one incoming and one outgoing edge. We denote the incoming edge by $e = 1$ and the outgoing edge by $e = 2$, respectively. Additionally, we define

$$\begin{aligned} c_1^n &= a\rho_N^{n,1}, \\ c_2^n &= \frac{\rho_{max} - \rho_1^{n,2}}{\lambda} + F_1^{n,2}. \end{aligned} \tag{2.25}$$

Generally, the values c_e^n denote the maximal possible flow at intersections for $e = 1, 2$. Thus, the actual outflow of edge 1 is the minimum of c_1^n and c_2^n, i.e., $F_N^{n,1} = \gamma_1^n = \min\{c_1^n, c_2^n\}$ or more precisely:

$$\text{(CPL A):} \qquad F_N^{n,1} = \min\{a\rho_N^{n,1}, \frac{\rho_{max} - \rho_1^{n,2}}{\lambda} + F_1^{n,2}\}. \tag{2.26}$$

Junction Type II. We consider a junction with two incoming edges and one outgoing edge. We denote the incoming edges by $e = 1, 2$ and the outgoing edge by $e = 3$. Additionally, we define

$$\begin{aligned} c_e^n &= a\rho_N^{n,e}, \qquad e = 1, 2 \\ c_3^n &= \frac{\rho_{max} - \rho_1^{n,3}}{\lambda} + F_1^{n,3}. \end{aligned} \tag{2.27}$$

By definition of the coupling conditions, $F_N^{n,e}, e = 1, 2$ can be computed in following way:

$$\text{(CPL B):} \qquad F_N^{n,e} = \begin{cases} \min\{c_1^n, c_3^n\}, & e = 1, \\ \min\{c_2^n, c_3^n - F_N^{n,1}\}, & e = 2. \end{cases} \tag{2.28}$$

Junction Type III. Now we consider a junction with one incoming edge $e = 1$ and two outgoing edges $e = 2, 3$. Again we define

$$\begin{aligned} c_1^n &= a\rho_N^{n,1}, \\ c_e^n &= \frac{\rho_{max} - \rho_1^{n,e}}{\lambda} + F_1^{n,e} \qquad e = 2, 3. \end{aligned} \tag{2.29}$$

The values c_e^n denote the maximal possible flow at the intersection for all edges $e = 1, 2, 3$.

$$\text{(CPL C):} \qquad F_N^{n,1} = \min\{c_1^n, \frac{c_2^n}{d_{2,1}}, \frac{c_3^n}{1 - d_{2,1}}\}. \tag{2.30}$$

Here, the parameter $d_{2,1} \in (0,1)$ denotes the distribution rate of the incoming flux among the outgoing edges:

$$\begin{aligned} F_0^{n,2} &= d_{2,1} F_N^{n,1}, \\ F_0^{n,3} &= (1-d_{2,1}) F_N^{n,1}. \end{aligned} \tag{2.31}$$

Junction Type IV. The last junction type considers only one incoming edge and no outgoing edges, i.e., it is a sink. The set of all sinks is denoted by E^{out}. It is necessary to prescribe outflow boundary conditions for all sinks. We introduce a variable f_{out}^e that limits the outflow of a sink $e \in E^{out}$ in following way.

$$\text{(CPL D):} \qquad F_N^{n,e} = \min\{a\rho_N^{n,e}, f_{out}^e\} \qquad \forall e \in E^{out}. \tag{2.32}$$

Remark 2.2.2. *From a computational point of view, we assume a cycle-free network and a topological ordering of the edges. Then, the numerical flux $F_i^{n,e}$ for all e, i can be computed efficiently: The process starts with the computation of $F_i^{n,e}$ for all edges e linked with a sink, i.e, $e \in E^{out}$. Further, due to the coupling condition (CPL D), the outflow of edge e is known and $F_i^{n,e}$ for all i and $e \in E^{out}$ can be solved. This yields the possible inflow for all edges $e \in E^{out}$. Applying all coupling conditions (CPL A)-(CPL C), the flux $F_N^{n,e}$ for all predecessors $e \in E \setminus E^{out}$ can be calculated. Remove all edges with a sink from the network and repeat this procedure until the set of edges is empty.*

For simulation purposes, the discretized model is summarized as an algorithm called **Discontinuous Flux Godunov Method (DFG)**:

`forwardsolutionPDE()`

---

(1) **For** $n = 1$ **to** $N_T - 1$
(2) `updateFLUX`(n)
(3) `updatePDE`(n)
(4) **End**

`updateFLUX`(n)

---

(1) **For** $j = 1$ **to** $|J|$
(2) **Solve** $F_N^{n,e} \quad \forall e \in \delta_v^{in}$ via coupling conditions CPL
(3) **Solve** $F_0^{n,e} \quad \forall e \in \delta_v^{out}$ via coupling conditions CPL
(4) **For all** $e \in \delta_j^{in}$
(5) **For** $i = N$ **to** 2 **Step** -1
(6) $F_{i-1}^{n,e} = \min\{a\rho_{i-1}^{n,e}, \frac{\rho_{max} - \rho_i^{n,e}}{\lambda} + F_i^{n,e}\}$
(7) **End**
(8) **End**
(9) **End**

`updatePDE`(n)

(1) **For all** $e \in E$
(2) **For** $i = 1$ **to** N
(3) $\rho_i^{n+1,e} = \rho_i^{n,e} - \lambda[F_i^{n,e} - F_{i-1}^{n,e}]$
(4) **End**
(5) **End**

Remark 2.2.3. *In the linear case discussed here the solution of the advection problem with $\lambda = 1$ is exact.*

Buffer Allocation

At this point, we try to find a connection between our model including discontinuous flux and a buffer allocation model that is introduced in the work of Stolletz and Weiss in [94, 101].
In consideration of the buffer allocation model, we consider a queuing network with individual production units m, consisting of a processor and a buffer with size C_m, see Figure 2.10. This model characterize a part (good) n individually by its arrival time $\tau_{m,n}$. The processing time of a part n in machine m is defined by $T_{m,n}$.

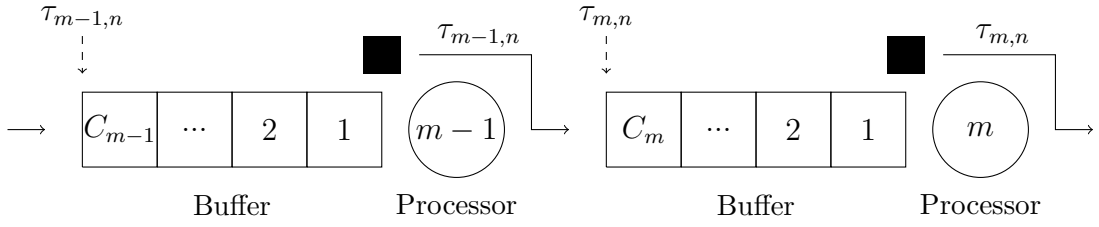


Figure 2.10: A serial production line.

The arrival times $\tau_{m,n}$ are defined by a recursion, so-called (τ-recursion), i.e.,

$$\tau_{m,n} = \max\{\max\{\tau_{m,n-1}, \tau_{m-1,n}\} + T_{m-1,n}, \tau_{m+1,n-(C_m+1)}\}.$$

In the work of Göttlich and Kolb in [50], an ODE system is derived from the discrete τ- recursion model that has the following structure:
The inflow of the buffer m is characterized by

$$\gamma_m(t, \Delta t) = \min\Big\{\mu_{m-1,\max}(t, \Delta t), \frac{w_{m-1}(t)}{\Delta t} + \gamma_{m-1}(t, \Delta t), \frac{C_m + 1 - w_m(t)}{\Delta t} + \gamma_{m+1}(t, \Delta t)\Big\}, \tag{2.33}$$

where $\mu_{m-1,\max}(t, \Delta t)$ is the maximum processing rate at the preceding processor. The buffer amount is given by the formula:

$$w_m(t + \Delta t) = w_m(t) + \Delta t(\gamma_m(t, \Delta t) - \gamma_{m+1}(t, \Delta t)). \tag{2.34}$$

The following approach motivates a PDE model based on a discontinuous flux that considering a large number of finite buffers. We assume that the parts cannot pass the buffers with infinite velocity. Thus, the equation (2.33) reduces to

$$\begin{aligned}\gamma_m(t, \Delta t) = \min\Big\{&\mu_{m-1,\max}(t, \Delta t), \\ &\frac{w_{m-1}(t)}{\Delta t}, \frac{C_m + 1 - w_m(t)}{\Delta t} + \gamma_{m+1}(t, \Delta t)\Big\}.\end{aligned} \tag{2.35}$$

We define a spatial component, in which the distance between two buffers is introduced as Δx. Additionally, we consider the parts as an averaged quantity (density), i.e., we define $\rho_m(t) := w_m(t)/\Delta x$, $\rho_{max} := (C_m + 1)/\Delta x$, $\lambda = \frac{\Delta t}{\Delta x}$, and $a := \frac{1}{\lambda}$. Hence, the inflow (2.35) of the buffer m is representable as

$$\gamma_m(t, \Delta t) = \min\Big\{\underbrace{\min\{a\rho_{m-1}(t), \mu_{m-1,\max}(t, \Delta t)\}}_{=:\tilde{f}(\rho_{m-1}(t))}, \frac{\rho_{max} - \rho_m(t)}{\lambda} + \gamma_{m+1}(t, \Delta t)\Big\}. \tag{2.36}$$

Equation (2.34) yields the evolution of the density

$$\rho_m(t + \Delta t) = \rho_m(t) + \lambda(\gamma_m(t, \Delta t) - \gamma_{m+1}(t, \Delta t)). \tag{2.37}$$

Obviously, (2.36) and (2.37) are similar to the equations (2.21) and (2.22). Thus, this leads to the supposition that (2.36) and (2.37) is the DFG method solving the following partial differential equation:

$$\begin{aligned}\partial_t\rho + \partial_x\tilde{f}(\rho) \cdot H(\rho_{max} - \rho) = 0, \\ \tilde{f}(\rho) = \min\{a\rho, \mu\}.\end{aligned} \tag{2.38}$$

The constant a prescribe the velocity of a good, which it needs in order to move to the next buffer. The processing rate is given by the function $\mu(x)$. Note that the flux function $\tilde{f}$ is similar to the model of Armbruster et al. in [3]. The buffer sizes could be prescribed by a space depended function $\rho_{max}(x)$.

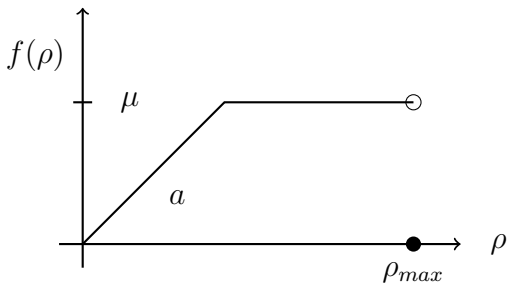


Figure 2.11: Discontinuous flux function of (2.38)

2.3 Mixed Integer Programming Model

For many applications, mixed integer programming models (MIP) play an important role. Even in application of PDE-constraint optimization problems, MIP models establish new possibilities to solve some of this problems in an accurate way, for instance, [25, 37, 51].
The following approach is based on discretization of a PDE model which can be optimized subsequently as a finite dimensional problem. We extend the previous MIP model that is already introduced in Chapter 1 to network topologies.
We use the DFG method for networks to create linear constraints with utilization of floating and binary variables. At first, we transform the discretized PDE (2.21) on a single edge into the MIP system.

$$\text{(PDE):} \qquad \rho_i^{n+1} = \rho_i^{n,e} - \lambda[F_i^{n,e} - F_{i-1}^{n,e}]. \tag{2.39}$$

The numerical flux are defined as $F_{i-1}^{n,e} = \min\{a\rho_{i-1}^{n,e}, \frac{\rho_{max}-\rho_i}{\lambda} + F_i^{n,e}\}$. By introducing binary variables $\xi_i^{n,e} \in \{0,1\}$ the numerical flux can be written as linear inequalities:

$$\begin{aligned}
&\text{(FLUX1):} && a\rho_{i-1}^{n,e} - \xi_{i-1}^{n,e}\mathcal{M} \le F_{i-1}^{n,e},\\
&\text{(FLUX2):} && F_{i-1}^{n,e} \le a\rho_{i-1}^{n,e},\\
&\text{(FLUX3):} && \frac{\rho_{max} - \rho_i^{n,e}}{\lambda} + F_i^{n,e} - (1-\xi_{i-1}^{n,e})\mathcal{M} \le F_{i-1}^{n,e},\\
&\text{(FLUX4):} && F_{i-1}^{n,e} \le \frac{\rho_{max} - \rho_i^{n,e}}{\lambda} + F_i^{n,e}.
\end{aligned} \tag{2.40}$$

where $i = 2, ..., N$, $n = 1, ..., N_T$, $e \in E$. Additionally $\mathcal{M}$ is a large constant, i.e., $\mathcal{M} > a\rho_{max}$

Junction Type I. We consider a junction with one incoming edges and one outgoing edge. Without loss of generality we denote the both incoming edges by integer numbers $e = 1$ and the outgoing one by $e = 2$. Generally, it is necessary

to reformulate the statement (CPL A) into linear inequalities.

$$\begin{aligned}
&\text{(CPL A1):} \quad a\rho_N^{n,1} - \mathcal{M}\xi_N^{n,1} \leq F_N^{n,1},\\
&\text{(CPL A2):} \quad F_N^{n,1} \leq a\rho_N^{n,1},\\
&\text{(CPL A3):} \quad \frac{\rho_{max} - \rho_1^{n,2}}{\lambda} + F_1^{n,2} - \mathcal{M}(1 - \xi_N^{n,1}) \leq F_N^{n,1},\\
&\text{(CPL A4):} \quad F_N^{n,1} \leq \frac{\rho_{max} - \rho_1^{n,2}}{\lambda} + F_1^{n,2}.
\end{aligned} \tag{2.41}$$

Junction Type II. We consider a junction with two incoming edges and one outgoing edge. Without loss of generality we denote the both incoming edges by integer numbers $i = 1, 2$ and the outgoing one by $e = 3$. Additionally, we define

$$\begin{aligned}
c_e^n &= a\rho_N^{n,e}, \qquad e = 1, 2,\\
c_3^n &= \frac{\rho_{max} - \rho_1^{n,3}}{\lambda} + F_1^{n,3}.
\end{aligned} \tag{2.42}$$

We reformulate the statement $F_N^{n,1} := \min\{c_1^n, c_3^n\}$ into a bundle of linear inequalities with binary variables $\xi_N^{n,e}$, $e = 1, 2$.

$$\begin{aligned}
&\text{(CPL B1):} \quad a\rho_N^{n,1} - \mathcal{M}\xi_N^{n,1} \leq F_N^{n,1},\\
&\text{(CPL B2):} \quad F_N^{n,1} \leq a\rho_N^{n,1},\\
&\text{(CPL B3):} \quad \frac{\rho_{max} - \rho_1^{n,3}}{\lambda} + F_1^{n,3} - \mathcal{M}(1 - \xi_N^{n,1}) \leq F_N^{n,1},\\
&\text{(CPL B4):} \quad F_N^{n,1} \leq \frac{\rho_{max} - \rho_1^{n,3}}{\lambda} + F_1^{n,3}.
\end{aligned} \tag{2.43}$$

The outflow $F_N^{n,2} := \min\{c_2^n, c_3^n - F_N^{n,1}\}$ can be transform into the following inequalities.

$$\begin{aligned}
&\text{(CPL B5):} \quad a\rho_N^{n,2} - \mathcal{M}\xi_N^{n,2} \leq F_N^{n,2},\\
&\text{(CPL B6):} \quad F_N^{n,2} \leq a\rho_N^{n,2},\\
&\text{(CPL B7):} \quad \frac{\rho_{max} - \rho_1^{n,3}}{\lambda} + F_1^{n,3} - F_N^{n,1} - \mathcal{M}(1 - \xi_N^{n,1}) \leq F_N^{n,2},\\
&\text{(CPL B8):} \quad F_N^{n,1} \leq \frac{\rho_{max} - \rho_1^{n,3}}{\lambda} + F_1^{n,3} - F_N^{n,1}.
\end{aligned} \tag{2.44}$$

Junction Type III. Now we consider a junction with one incoming edge and two outgoing edges. In due to the previous cases, we sign the ingoing edges by $e = 1$ and the both outgoing edges by $e = 2, 3$.

$$\begin{aligned}
c_1^n &= a\rho_N^{n,1},\\
c_e^n &= \frac{\rho_{max} - \rho_1^{n,e}}{\lambda} + F_1^{n,e}, \qquad e = 2, 3.
\end{aligned} \tag{2.45}$$

Generally the values c_e^n denotes the maximal possible flow at the junction point for all edges $e = 1, 2, 3$. Thus, the actual outflow of the edge 1 is minimum of c_1^n and $c_2^n + c_3^n$, i.e., $\gamma_1^n = \min\{c_1^n, c_2^n + c_3^n\}$. This expression can be formulated as linear inequalities with binary variables.

$$\begin{aligned}
&\text{(CPL C1):} \quad && a\rho_N^{n,1} - \mathcal{M}\xi_N^{n,1} \leq F_N^{n,1},\\
&\text{(CPL C2):} \quad && F_N^{n,1} \leq a\rho_N^{n,1},\\
&\text{(CPL C3):} \quad && \frac{\rho_{max} - \rho_1^{n,2}}{\lambda} + F_1^{n,2} + \frac{\rho_{max} - \rho_1^{n,3}}{\lambda} + F_1^{n,3} - \mathcal{M}(1 - \xi_N^{n,1}) \leq F_N^{n,1},\\
&\text{(CPL C4):} \quad && F_N^{n,1} \leq \frac{\rho_{max} - \rho_1^{n,2}}{\lambda} + F_1^{n,2} + \frac{\rho_{max} - \rho_1^{n,3}}{\lambda} + F_1^{n,3}.
\end{aligned} \tag{2.46}$$

The actual inflow $F_0^{n,e}$ is limited by the maximal possible inflow c_e^n for all outgoing edges $e = 2, 3$.

$$\begin{aligned}
&\text{(CPL C5):} \quad && 0 \leq F_0^{n,2} \leq \frac{\rho_{max} - \rho_1^{n,2}}{\lambda} + F_1^{n,2},\\
&\text{(CPL C6):} \quad && 0 \leq F_0^{n,3} \leq \frac{\rho_{max} - \rho_1^{n,3}}{\lambda} + F_1^{n,3}.
\end{aligned} \tag{2.47}$$

Summarizing, the mixed integer model derived by the DFG method for networks is given by

$$\begin{aligned}
&\min \sum_{e\in E} \sum_{n=1}^{N_T-1} \sum_{i=1}^{N} C_i^{n,e}\rho_i^{n,e} + D_i^{n,e}F_i^{n,e}\\
&\text{subject to}\\
&\text{(PDE), (CPL)}\\
&\text{(FLUX1) - (FLUX4)}\\
&\text{(CPL A1) - (CPL A4)}\\
&\text{(CPL B1) - (CPL B8)}\\
&\text{(CPL C1) - (CPL C6) ,}
\end{aligned} \tag{2.48}$$

where $C_i^{n,e}$, $D_i^{n,e}$ are real weights for the linear objective function.

2.4 Presolve Techniques for the MIP Model

A disadvantage of the mixed integer model is the huge computational effort for large problems. As a consequence, the mixed integer problem cannot be solved efficiently without the use of additional structural information. Therefore, we are interested in efficient presolve techniques that can be applied to the MIP model

to reduce the size of the mixed integer problem and its computational solution time.
In this section we introduce a description of bounds strengthening for the presented MIP model. The task is to find and remove redundant parts of the problem formulation, and thus output a somehow strengthened version of the model.
Bound strengthening techniques for general LP and MIP models are well-known investigated, e.g., [2, 92]. In particular, bound strengthening techniques for MIP formulations of PDE models are considered in [33].
At first, we present the basic concepts of bound strengthening for general MIP models. Afterwards, we apply the presolve level 1 approach of [33] to our MIP network model. Finally, we introduce an improvement of the previous presolving strategy, namely the presolve level 2 that is especially based on the knowledge of the PDE structure.
The notations, the general bound strengthening techniques, and concepts of the presolve level 1 technique in this chapter orientate to [33].

2.4.1 Boundstrengthening in general

Note that each variable of the constraint system has lower and upper bounds. The aim is to improve the bounds for the variables or if possible solve some variables directly for reducing the size of the constraint system. In principle, best possible bounds can be obtained by taking each variable as objective function and solve a minimization (for the lower bound) and a maximization problem (for the upper bound). However, such a procedure would be too time consuming in practice. Bound strengthening is a simpler technique to obtain such improvements by using informations only from the constraint system and the given bounds.
An arbitrary MIP Model is given by

$$\begin{aligned} &\min c^T x \\ &\text{subject to} \\ &Ax \geq b \\ &x \in \mathbb{R}_+^{n-p} \times \{0,1\}^p, \end{aligned}$$

where $c \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, and $A \in \mathbb{R}^{m \times n}$. We consider the i-th constraint inequality of the system $Ax \geq b$, which has the following form

$$\text{(IEQ):} \qquad \sum_{j=1}^{n} a_{ij} x_j \geq b_j. \tag{2.49}$$

Each variable x_j has a lower and upper bound l_j, u_j with $l_j \leq x_j \leq u_j$ with $l_j \in \mathbb{R}_+ \cup \{-\infty\}$,$u_j \in \mathbb{R}_+ \cup \{+\infty\}$. If a better bound for an x_j exists, i.e., $l_j \leq l_j^*$

or $u_j \geq u_j^*$, then the set of feasible solutions does not change, i.e.,

$$\begin{aligned} X &= \{x \in \mathbb{R}_+^{n-p} \times \{0,1\}^p : Ax \geq b, l_j \leq x_j \leq u_j\} \\ &= \{x \in \mathbb{R}_+^{n-p} \times \{0,1\}^p : Ax \geq b, l_j^* \leq x_j \leq u_j^*\}. \end{aligned}$$

Bound strengthening is a technique to obtain better bounds by using informations soley from the constraint system $Ax \geq b$ and given bounds $l \leq x \leq u$.
In the following, we extract a variable x_j of the i-th constraint inequality (2.49). This is equivalent to

$$a_{ij}x_j \geq b_j - \sum_{k=1,k\neq j}^{n} a_{ik}x_k. \tag{2.50}$$

Now we define the positive and negative parts of the coefficient a_{ij}:

$$a_{ij}^+ := \max\{a_{ij}, 0\}, \quad a_{ij}^- := \min\{a_{ij}, 0\}.$$

We consider two cases. The first case holds if $a_{ij} > 0$. Thus, we can expand the inequality (2.50) to

$$x_j \geq \frac{1}{a_{ij}}\left(b_j - \sum_{k=1,k\neq j}^{n} a_{ik}^+ u_k - \sum_{k=1,k\neq j}^{n} a_{ik}^- l_k\right) =: l_j^*. \tag{2.51}$$

Let x_j be a real variable. if $l_j^* > l_j$ is valid, we have found an improved lower bound. Then we set $l_j := \max\{l_j^*, l_j\}$. If x_j is an integer variable, the improved lower bound l^* can be rounded up to the next integer, i.e., $l_j := \max\{l_j^*, \lceil l_j \rceil\}$. We denote by `lowerboundStrengthening`($x_j$,(IEQ)) the process that tries to update the lower bound on variable x_j using the MIP constraint (IEQ).
The other case $a_{i,j} < 0$ works analogously and results the following upper bound equation

$$x_j \leq \frac{1}{a_{ij}}\left(b_j - \sum_{k=1,k\neq j}^{n} a_{ik}^+ u_k - \sum_{k=1,k\neq j}^{n} a_{ik}^- l_k\right) =: u_j^*. \tag{2.52}$$

Thus, if x_j is a real variable, the improved upper bound is computed by $u_j := \min\{u_j^*, u_j\}$. If x_j is an integer, the improved upper bound can be rounded to the next integer, i.e., $u_j := \max\{u_j^*, \lfloor u_j \rfloor\}$. Analogously to the lower bound procedure, we define the process by `upperboundStrengthening`(x_j,(IEQ)).
In case of an equation constraint, i.e.,

$$\text{(EQ):} \qquad \sum_{j=1}^{n} a_{ij}x_j = b_j, \tag{2.53}$$

we can split (2.53) into two inequalities

$$\sum_{j=1}^{n} a_{ij}x_j \geq b_j, \quad -\sum_{j=1}^{n} a_{ij}x_j \geq -b_j. \tag{2.54}$$

Both inequalities of (2.54) yield an upper and a lower bound for the variable x_j by the previous process. We denote by `boundStrengthening`(x_j,(EQ)) the process that tries to update the lower and upper bound on variable x_j using the MIP equality constraint (EQ).
In general MIP preprocessing, the previous steps are performed for all constraints $i = 1, ..., m$ and all variables $j = 1, ..., n$. This procedure is carried out until either no better bounds are found anymore or an infeasible problem is detected. In the latter case, the constraints contradicts each other if the preprocessing procedure finds bounds such that $l_j > u_j$. Hence, the preprocessing classifies the MIP as an infeasible problem. In the case of $l_j = u_j$, the variable x_j is solved and can be removed out of the MIP for reducing the complexity of the model.

2.4.2 Presolve Level 1

The previous subsection describes the basics of a bound strengthening method for general MIP problems. At this point, it is therefore unclear what sequence of bound strengthening routines is performed to obtain an efficient presolving technique. Under certain circumstances, the presolving procedure could take a long computation time, or the desired results are not fulfilled.
In this subsection, we introduce the presolve level 1 routine that is based on the presented bounds strengthening procedures and a careful reordering of the constraints.

At first, a notation of the bounds is defined in consideration of the presolve level 1 routine.

Definition 2.4.1. *As abbreviations we write* $\underline{\rho}_i^{n,e}, \overline{\rho}_i^{n,e}, \underline{F}_i^{n,e}, \overline{F}_i^{n,e}, \underline{\xi}_i^{n,e}, \overline{\xi}_i^{n,e}$ *for the lower and upper bounds of the variables* $\rho_i^{n,e}$, $F_i^{n,e}$, $\xi_i^{n,e}$ *respectively.*

Remark 2.4.2. *A variable is solved by preprocessing or known if and only if the lower and upper bound coincides. Thus, solved variables can be reduces from the constrained system.*

Next, we determine explicitly the bound strengthening routines for each constraints of the MIP model.
In consideration of the procedure in Subsection 2.4.1, the bounds of the density are computed by the routine

`boundStrengthening`($\rho_i^{n,e}$,(PDE)), i.e.,

$$\underline{\rho}_i^{n,e} = \underline{\rho}_i^{n-1,e} + \lambda \underline{F}_i^{n-1,e} - \lambda \overline{F}_{i-1}^{n-1,e},$$
$$\overline{\rho}_i^{n,e} = \overline{\rho}_i^{n-1,e} + \lambda \overline{F}_i^{n-1,e} - \lambda \underline{F}_{i-1}^{n-1,e}.$$

For the bound strengthening of the flux variables, we do not need to use the linear formulations (FLUX1) - (FLUX4). We can directly apply the nonlinear constraint (FLUX) instead. For the flux constraint, we introduce the following procedure `nonlinearboundStrengthening`($F_{i-1}^{n,e}$,(FLUX)), which yields the corresponding bounds

$$\underline{F}_{i-1}^{n,e} = \min\{a\underline{\rho}_{i-1}^{n,e}, \frac{\rho_{max} - \overline{\rho}_i^{n,e}}{\lambda} + \underline{F}_i^{n,e}\},$$
$$\overline{F}_{i-1}^{n,e} = \min\{a\overline{\rho}_{i-1}^{n,e}, \frac{\rho_{max} - \underline{\rho}_i^{n,e}}{\lambda} + \overline{F}_i^{n,e}\}.$$

A simple calculation shows that $\underline{F}_{i-1}^{n,e}$, ($\overline{F}_{i-1}^{n,e}$) is a upper (lower) bound of $F_{i-1}^{n,e}$. The lower bound of the binary variable $\xi_i^{n,e}$ is computed as follows. The constraint (FLUX1) obtains the inequality

$$\frac{a\rho_i^{n,e} - F_i^{n,e}}{\mathcal{M}} \leq \xi_i^{n,e}. \tag{2.55}$$

Obviously, the left hand side of the term (2.55) is a lower bound of $\xi_i^{n,e}$. However, $\xi_i^{n,e}$ is a binary variable, and thus the lower bound is rounded up to the next integer. This reveals the routine `lowerboundStrengthening`($\xi_i^{n,e}$,(FLUX1)) with the following computation

$$\underline{\xi}_i^{n,e} = \max\left\{\underline{\xi}_i^{n,e}, \left\lceil \frac{a\underline{\rho}_i^{n,e} - \overline{F}_i^{n,e}}{\mathcal{M}} \right\rceil\right\}.$$

We find an upper bound of $\xi_i^{n,e}$ by using the constraint (FLUX3). Consequently, we obtain for $\xi_i^{n,e}$ an upper bound, i.e.,

$$\begin{aligned}\xi_i^{n,e} &\leq 1 - \frac{1}{\mathcal{M}}\left(F_{i+1}^{n,e} - F_i^{n,e} + \frac{\rho_{max} - \rho_{i+1}^{n,e}}{\lambda}\right)\\ &\leq 1 - \frac{1}{\mathcal{M}}\left(\overline{F}_{i+1}^{n,e} - \underline{F}_i^{n,e} + \frac{\rho_{max} - \overline{\rho}_{i+1}^{n,e}}{\lambda}\right).\end{aligned}$$

Hence, this yields the routine `upperboundStrengthening`($\xi_i^{n,e}$,(FLUX3)) with the following computation

$$\overline{\xi}_i^{n,e} = \min\left\{\overline{\xi}_i^{n,e}, \left\lfloor 1 - \frac{1}{\mathcal{M}}(\overline{F}_i^{n,e} - \underline{F}_{i+1}^{n,e} + \frac{\rho_{max} - \overline{\rho}_{i+1}^{n,e}}{\lambda}) \right\rfloor\right\}.$$

Finally, we calculate the lower and upper bounds for the flux at the junctions, i.e., $F_0^{n,e}$, $F_N^{n,e}$. However, we use each constraints of the coupling conditions on each junctions.
At first, we consider a junction with one incoming edge $e = 1$ and one outgoing edge $e = 2$. The bound strengthening routine of the flux $F_N^{n,1}$ in due of the junction type I is defined as follows.
`nonlinearboundStrengthening`($F_{N,1}^{n,e}$,(CPL A))

$$\underline{F}_N^{n,1} = \min\left\{ a\underline{\rho}_N^{n,1}, \frac{\rho_{max} - \overline{\rho}_1^{n,2}}{\lambda} + \underline{F_1}^{n,2} \right\},$$
$$\overline{F}_N^{n,1} = \min\left\{ a\overline{\rho}_N^{n,1}, \frac{\rho_{max} - \underline{\rho}_1^{n,2}}{\lambda} + \overline{F_1}^{n,2} \right\}.$$

Now we consider the boundstrengthening routine for the junction type II, i.e., `nonlinearboundStrengthening`($F_{N,e}^{n,e}$,(CPL B)):

$$\underline{c}_1 := a\underline{\rho}_N^{n,1}, \quad \underline{c}_2 := a\underline{\rho}_N^{n,1}, \quad \underline{c}_3 := \frac{\rho_{max} - \overline{\rho}_1^{n,3}}{\lambda} + \underline{F}_1^{n,3},$$
$$\overline{c}_1 := a\overline{\rho}_N^{n,1}, \quad \overline{c}_2 := a\overline{\rho}_N^{n,1}, \quad \overline{c}_3 := \frac{\rho_{max} - \underline{\rho}_1^{n,3}}{\lambda} + \overline{F}_1^{n,3}.$$

$$\underline{F}_N^{n,1} = \min\{\underline{c}_1, \underline{c}_3\}, \quad \overline{F}_N^{n,1} = \min\{\overline{c}_1, \overline{c}_3\},$$
$$\underline{F}_N^{n,2} = \min\{\underline{c}_2, \underline{c}_3 \ - \overline{F}_N^{n,1}\}, \quad \overline{F}_N^{n,2} = \min\{\overline{c}_2, \overline{c}_3 \ - \underline{F}_N^{n,1}\}.$$

$$\underline{F}_1^{n,3} = \underline{F}_N^{n,1} + \underline{F}_N^{n,2}, \quad \overline{F}_1^{n,3} = \overline{F}_N^{n,1} + \overline{F}_N^{n,2}.$$

Finally, we use the junction type III for bound strengthening of the flux at the junction, i.e., `nonlinearboundStrengthening`($F_{N,e}^{n,e}$,(CPL C)):

$$\underline{F}_N^{n,1} = \min\left\{ a\underline{\rho}_N^{n,1}, \frac{\rho_{max} - \overline{\rho}_1^{n,2}}{\lambda} + \underline{F_1}^{n,2} + \frac{\rho_{max} - \overline{\rho}_1^{n,3}}{\lambda} + \underline{F_1}^{j,3} \right\}, \tag{2.56}$$
$$\overline{F}_N^{n,1} = \min\left\{ a\overline{\rho}_N^{n,1}, \frac{\rho_{max} - \underline{\rho}_1^{n,2}}{\lambda} + \overline{F_1}^{n,2} + \frac{\rho_{max} - \underline{\rho}_1^{n,3}}{\lambda} + \overline{F_1}^{n,3} \right\}. \tag{2.57}$$

After defining the bound strengthening routines for each model constraint, we sort the constraints and apply bound strengthening in order of the network solution algorithm `forwardsolutionPDE()`. Hence, the presolve level 1 routine is summarized to the following algorithm:

```
presolveLevel1()
```

(1) **For** $n = 1$ **to** $N_T - 1$
(2) **For** $v = 1$ **to** $|V|$
(3) nonlinearboundStrengthening($F_N^{n,e}$,(CPL)) $\forall e \in \delta_v^{in}$
(4) nonlinearboundStrengthening($F_0^{n,e}$,(CPL)) $\forall e \in \delta_v^{out}$
(5) lowerboundStrengthening($\xi_N^{n,e}$,(CPL)) $\forall e \in \delta_v^{in}$
(6) upperboundStrengthening($\xi_N^{n,e}$,(CPL)) $\forall e \in \delta_v^{in}$
(7) **For all** $e \in \delta_v^{in}$
(8) **For** $i = N$ **to** 2 **Step** -1
(9) nonlinearboundStrengthening($F_{i-1}^{n,e}$,(FLUX))
(10) **End**
(11) **End**
(12) **End**
(13) **For all** $e \in E$
(14) **For** $i = 1$ **to** N
(15) boundStrengthening($\rho_i^{n,e}$,(PDE))
(16) upperboundStrengthening($\xi_i^{n,e}$,(FLUX1))
(17) lowerboundStrengthening($\xi_i^{n,e}$,(FLUX3))
(18) **End**
(19) **End**
(20) **End**

Remark 2.4.3. *We expect that the presolve algorithm solves completely the network problem, if the network problem has only one unique feasible solution. Such network problems exists if only the junction types I, II, and IV are used, and also the inflow and the initial values are known. In that case, each bound strengthening step is equivalent to the computation steps of the forward solver. Thus, the presolve algorithm reveals the unique feasible solution of the network problem.*

2.4.3 Presolve Level 2

In practice, the presolving level 1 strategy does not calculate the best bounds for the mixed integer problem of Section 2.3. Therefore, we introduce another method for an efficient presolving. For our motivation, we consider a conservation law with two different initial data $\rho(x,0)$ and $\overline{\rho}(x,0)$ that fulfills $\rho(x,0) \leq \overline{\rho}(x,0)$. The monotonicity statement of the Kruskow theorem for conservation laws leads to the following result. The density $\overline{\rho}(x,t)$ is larger than the density $\rho(x,t)$ for all $x \in \mathbb{R}$ and $t > 0$, i.e., $\rho(x,t) \leq \overline{\rho}(x,t)$. Indeed, the DFG method yields the same results for discretized version of our model, cf. Lemma 1.3.2. Hence, an

additional solving of the conservation law yields an upper bound. The procedure is the same to obtain lower bounds. However, this is valid for the PDE in one dimension. The question raises whether this procedure is applicable to our network problem?
Nevertheless, it is possible to obtain lower and upper bounds for our problem if we modify the forward network simulation of Section 2.2. In detail, the presented presolving level 2 routine is based directly on the forward network simulation with modified coupling conditions that are specified in this subsection.

We assume that the inflow and the initial data at starting time $t = 0$ is known. Additionally, the following procedure is restricted to the optimal routing problem, i.e., we are interested in finding optimal distribution rates in junctions with respect to an objective function.
Next, we introduce the modified coupling conditions for the presented presolving level 2 routine.

Definition 2.4.4 (Modified coupling conditions)**.** *The set (CPL UP) consists of the coupling conditions CPL A,B,D introduced in Section 2.2 and CPL* C^{UP}*. Analogously, the set (CPL LOW) contains of the coupling conditions CPL A,B,D and CPL* C^{LOW}*.*
The coupling condition CPL C^{UP} *is defined as follows:*
We consider a junction with one incoming edge $e = 1$ *and two outgoing edges* $e = 2, 3$*. The fluxes at the junction is given by*

$$\text{(CPL } C^{UP}\text{):} \qquad \begin{aligned} F_N^{n,1} &= \min\{c_1^n, c_2^n + c_3^n\}, \\ F_0^{n,2} &= \min\{c_1^n, c_2^n\}, \\ F_0^{n,3} &= \min\{c_1^n, c_3^n\}, \end{aligned}$$

where

$$\begin{aligned} c_1^n &= a\rho_N^{n,1}, \\ c_e^n &= \frac{\rho_{max} - \rho_1^{n,e}}{\lambda} + F_1^{n,e}, \qquad e = 2, 3. \end{aligned}$$

Note that the values c_e^n *denote the maximal possible flow at the intersection for all edges* $e = 1, 2, 3$*. Respectively, the coupling condition CPL* C^{LOW} *is defined in a similar way*

$$\text{(CPL } C^{LOW}\text{):} \qquad \begin{aligned} F_N^{n,1} &= \min\{c_1^n, c_2^n + c_3^n\}, \\ F_0^{n,2} &= 0, \\ F_0^{n,3} &= 0. \end{aligned}$$

Remark 2.4.5. *The coupling conditions* C^{UP} *and* C^{LOW} *of Definition 2.4.4 do not fulfill any mass conservation. In general, it does not hold* $F_N^{n,1} = F_0^{n,2} + F_0^{n,3}$*.*

However, the modified coupling conditions can be plugged in the DFG method for networks. The result is a novel bound strengthening routine:

`densityBoundStrengthening`()

---

(1) **Initial:** $\overline{\rho}_i^{1,e} := \underline{\rho}_i^{1,e} := \rho_i^{1,e}, \quad \forall i = 1, ..., N$
(2) **Initial:** $\widetilde{F}_0^{n,e} := \underset{\sim}{F}_0^{n,e} := F_0^{n,e}, \quad \forall n = 1, ..., N_T - 1, \quad \forall e \in E^{in}$
(3) **For** $n = 1$ **to** $N_T - 1$
(4) **For** $v = 1$ **to** $|V|$
(5) **Solve** $\widetilde{F}_N^{n,e}, (\underset{\sim}{F}_N^{n,e}) \quad \forall e \in \delta_v^{in}$ via CPL UP, (CPL LOW)
(6) **Solve** $\widetilde{F}_0^{n,e}, (\underset{\sim}{F}_0^{n,e}) \quad \forall e \in \delta_v^{out}$ via CPL UP, (CPL LOW)
(7) **For all** $e \in \delta_v^{in}$
(8) **For** $i = N$ **to** 2 **Step** -1
(9) $\widetilde{F}_{i-1}^{n,e} = \min\{a\overline{\rho}_{i-1}^{n,e}, \frac{\rho_{max} - \overline{\rho}_i^{n,e}}{\lambda} + \widetilde{F}_i^{n,e}\}$
(10) $\underset{\sim}{F}_{i-1}^{n,e} = \min\{a\underline{\rho}_{i-1}^{n,e}, \frac{\rho_{max} - \underline{\rho}_i^{n,e}}{\lambda} + \underset{\sim}{F}_i^{n,e}\}$
(11) **End**
(12) **End**
(13) **End**
(14) **For all** $e \in E$
(15) **For** $i = 1$ **to** N
(16) $\overline{\rho}_i^{n+1,e} = \overline{\rho}_i^{n,e} - \lambda[\widetilde{F}_i^{n,e} - \widetilde{F}_{i-1}^{n,e}]$
(17) $\underline{\rho}_i^{n+1,e} = \underline{\rho}_i^{n,e} - \lambda[\underset{\sim}{F}_i^{n,e} - \underset{\sim}{F}_{i-1}^{n,e}]$
(18) **End**
(19) **End**
(20) **End**

Nevertheless, the routine `densityBoundStrengthening`() is equivalent to the forward network solver including the modified coupling conditions.
The computational steps (5) and (6) of the presented routine evaluate the numerical inflow and outflow with respect to the coupling conditions (CPL UP) and (CPL LOW). The resulting quantities $\overline{\rho}_i^{n,e}$ and $\underline{\rho}_i^{n,e}$ are the upper and lower bounds of $\rho_i^{n,e}$. In due of the algorithm `densityBoundStrengthening`(), $\widetilde{F}_i^{n,e}$ and $\underset{\sim}{F}_i^{n,e}$ denotes the numerical fluxes of $\overline{\rho}_i^{n,e}$ and $\underline{\rho}_i^{n,e}$ respectively.

Finally, all upper and lower bounds of the network problem can be solved by the presolving level 2 algorithm that is structured as follows.
At first, the routine `densityBoundStrengthening`() is called to compute the lower und upper bounds of all density variables $\rho_i^{n,e}$.
Afterwards, `presolveLevel1()` is performed to evaluate upper and lower bounds of the remaining variables $F_i^{n,e}$, $\xi_i^{n,e}$.

`presolveLevel2()`

(1) `densityBoundStrengthening()`
(2) `presolveLevel1()`

Finally, we prove that the presented presolve routine works correctly:

The routine `densityBoundStrengthening()` yields always upper and lower bounds of the density values $\rho_i^{n,e}$.

Here is a short outline of the proof. In consideration of a single edge of the network, the routines `densityBoundStrengthening()` and `forwardsolutionPDE()` are equivalent, i.e, both routines are the DFG method. Firstly, we derive a monotonicity criteria for the DFG method on a single edge in due of inflow and outflow conditions, i.e.,

$$\rho_i^{n,e} \leq \overline{\rho}_i^{n,e} \Rightarrow \rho_i^{n+1,e} \leq \overline{\rho}_i^{n+1,e}. \tag{2.58}$$

Afterwards, we show that the routines `densityBoundStrengthening()` and `forwardsolutionPDE()` fulfills the assumptions of the monotonicity criteria for all edges of the network. Finally, we obtain (2.58) for all edges $e \in E$ and the statement is proven.

Especially for the proof, we reduce the notation of the inflow and outflow conditions. According to the discrete network model, the numerical inflow (outflow) of an edge e is given by $F_0^{n,e}$ ($F_N^{n,e}$). We observe that these terms are always representable by a minimum function, i.e., $F_0^{n,e} = F_N^{n,e} = \min\{\cdot,\cdot\}$. In the following, we generalize the notation for the numerical inflow and outflow as

$$F_0^{n,e} = \min\{c_{pre}^e, c_{in}^e\}, \quad F_N^{n,e} = \min\{a\rho_N^{n,e}, c_{out}^e\},$$

where e is an arbitraty edge of the network. Also, we assume that e is connected to preceeding edges $\tilde{e} \in \delta^{in}(\{e\})$, see e.g. Figure 2.12. The maximal inflow of edge e is prescribed by c_{pre}^e. Also, c_{pre}^e is representable as the sum of the maximal possible outflow of the incoming edges $\tilde{e} \in \delta^{in}(\{e\})$, i.e.,

$$c_{pre}^e := \sum_{\tilde{e} \in \delta^{in}(\{e\})} d_{e,\tilde{e}} \cdot a\rho_N^{n,\tilde{e}},$$

where $0 \leq d_{e,\tilde{e}} \leq 1$ is the actual distribution rate of the coupling. Note that the inflow $F_0^{n,e}$ is also bounded by the maximal allowed inflow

$$c_{in}^e := \frac{\rho_{max} - \rho_1^{n,e}}{\lambda} + F_1^{n,e}. \tag{2.59}$$

Additionally, the outflow of an edge e is representable as $F_N^{n,e} = \min\{a\rho_N^{n,e}, c_{out}^e\}$, where c_{out}^e is the maximal possible outflow and it depends on the actual coupling condition.

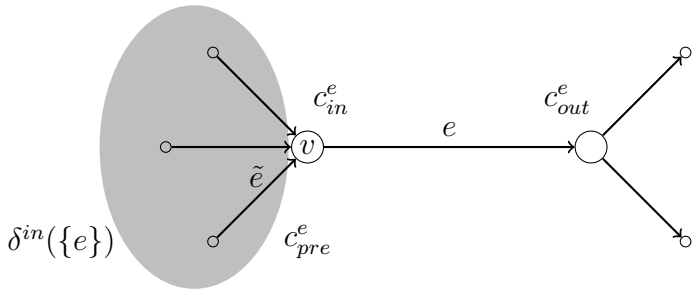


Figure 2.12: An edge e with predecessor edges $\tilde{e} \in \delta^{in}(\{e\})$. The inflow is bounded by c_{pre}^e, c_{in}^e and the outflow is bounded by c_{out}^e.

Thus, c_{pre}^e, c_{in}^e, c_{out}^e, and the density $\rho_i^{n,e}$ characterizes the inflow and outflow of the edge e.

Remark 2.4.6. *In the following, $\widetilde{F}_i^{n,e}$ denotes the corresponding flux for the density $\overline{\rho}_i^{n,e}$. Additionally, the fluxes $\widetilde{F}_0^{n,e}$, $\widetilde{F}_N^{n,e}$ are characterized by $\overline{c}_{pre}^e$, $\overline{c}_{in}^e$, $\overline{c}_{out}^e$.*

The following lemma is a monotonicity criteria for a single edge depending on the inflow and outflow coefficients c_{pre}^e, c_{in}^e, c_{out}^e.

Lemma 2.4.7 (Single edge monotonicity)**.** *The condition $\rho_i^{n,e} \leq \overline{\rho}_i^{n,e}$ holds for an arbitrary n, for all $i = 1, ..., N$, and for a fixed edge e. Additionally, the statements $c_{out}^e \geq \overline{c}_{out}^e$ and $c_{pre}^e \leq \overline{c}_{pre}^e$ are valid.*
Then the DFG method that is introduced in Section (2.2) yields $\rho_i^{n+1,e} \leq \overline{\rho}_i^{n+1,e}$ for all $i = 1, ..., N$. Additionally, the statement $c_{in}^e \geq \overline{c}_{in}^e$ holds.

Proof. The numerical flux of the DFG method is defined as

$$F_i^{n,e} := \min\{a\rho_i^{n,e}, \frac{\rho_{max} - \rho_{i+1}^{n,e}}{\lambda} + F_{i+1}^{n,e}\},$$
$$\widetilde{F}_i^{n,e} := \min\{a\overline{\rho}_i^{n,e}, \frac{\rho_{max} - \overline{\rho}_{i+1}^{n,e}}{\lambda} + \widetilde{F}_{i+1}^{n,e}\}.$$

Let $F_i^{n,e} < a\rho_i^{n,e}$ be the numerical flux in the blocking state. Thus, the flux is given by $F_i^{n,e} = \frac{\rho_{max} - \rho_{i+1}^{n,e}}{\lambda} + F_{i+1}^{n,e}$ and we obtain the recursion

$$F_i^{n,e} = \sum_{k=i}^{N} \frac{\rho_{max} - \rho_k^{n,e}}{\lambda} + c_{out}^e.$$

By the assumptions $c_{out}^e \geq \overline{c}_{out}$ and $\rho_i^{n,e} \leq \overline{\rho}_i^{n,e}$, the following estimation holds

$$\sum_{k=i}^{N} \frac{\rho_{max} - \overline{\rho}_k^{n,e}}{\lambda} + \overline{c}_{out}^e \leq \sum_{k=i}^{N} \frac{\rho_{max} - \rho_k^{n,e}}{\lambda} + c_{out}^e = F_i^{n,e} < a\rho_i^{n,e} \leq a\overline{\rho}_i^{n,e}.$$

As a consequence, the numerical flux

$$\widetilde{F}_i^{n,e} = \frac{\rho_{max} - \overline{\rho}_{i+1}^{n,e}}{\lambda} + \widetilde{F}_{i+1}^{n,e} = \sum_{k=i}^{N} \frac{\rho_{max} - \overline{\rho}_k^{n,e}}{\lambda} + \overline{c}_{out}^e < a\overline{\rho}_i^{n,e}$$

is also in the blocking state and we obtain

$$F_i^{n,e} = \frac{\rho_{max} - \rho_{i+1}^{n,e}}{\lambda} + F_{i+1}^{n,e} \quad \Rightarrow \quad \widetilde{F}_i^{n,e} = \frac{\rho_{max} - \overline{\rho}_{i+1}^{n,e}}{\lambda} + \widetilde{F}_{i+1}^{n,e} \quad \text{with } \widetilde{F}_i^{n,e} \leq F_i^{n,e}. \tag{2.60}$$

The negation of (2.60) implies

$$\widetilde{F}_i^{n,e} = a\overline{\rho}_i^{n,e} \Rightarrow F_i^{n,e} = a\rho_i^{n,e} \quad \text{with } F_i^{n,e} \leq \widetilde{F}_i^{n,e}. \tag{2.61}$$

The next step is to prove that $\rho_i^{n+1,e} \leq \overline{\rho}_i^{n+1,e}$. Therefore, we distinguish three different cases.

- **Case 1**: Let $\widetilde{F}_{i-1}^{n,e} = \frac{\rho_{max} - \overline{\rho}_i^{n,e}}{\lambda} + \widetilde{F}_i^{n,e}$. The (PDE) constraint yields

$$\overline{\rho}_i^{n+1,e} \overset{\text{(PDE)}}{=} \overline{\rho}_i^{n,e} - \lambda\widetilde{F}_i^{n,e} + \lambda\widetilde{F}_{i-1}^{n,e} = \rho_{max} \geq \rho_i^{n+1,e}.$$

- **Case 2**: Let $\widetilde{F}_{i-1}^{n,e} = a\overline{\rho}_{i-1}^{n,e}$ and $F_i^{n,e} = \frac{\rho_{max} - \rho_{i+1}^{n,e}}{\lambda} + F_{i+1}^{n,e}$.
  Then (2.60) and (2.61) result $\widetilde{F}_i^{n,e} \underset{(2.60)}{\leq} F_i^{n,e}$ and $F_{i-1}^{n,e} \underset{(2.61)}{=} a\rho_{i-1}^{n,e}$. Hence, this yields

$$\overline{\rho}_i^{n+1,e} \overset{\text{(PDE)}}{=} \overline{\rho}_i^{n,e} - \lambda\widetilde{F}_i^{n,e} + a\lambda\overline{\rho}_{i-1}^{n,e} \geq \rho_i^{n,e} - \lambda F_i^{n,e} + a\lambda\rho_{i-1}^{n,e} \overset{\text{(PDE)}}{=} \rho_i^{n+1,e},$$

  where $i > 1$. At the left boundary cell $i = 1$, we receive the estimation

$$\overline{\rho}_1^{n+1,e} = \overline{\rho}_1^{n,e} - \lambda\widetilde{F}_i^{n,e} + \lambda\overline{c}_{pre}^e \geq \rho_1^{n,e} - \lambda F_i^{n,e} + \lambda c_{pre}^e = \rho_1^{n+1,e}.$$

- **Case 3**: Let $\widetilde{F}_{i-1}^{n,e} = a\overline{\rho}_{i-1}^{n,e}$ and $F_i^{n,e} = a\rho_i^{n,e}$. Note that the CFL condition yields $(1 - a\lambda) \geq 0$. Thus, we get

$$\overline{\rho}_i^{n+1,e} \overset{\text{(PDE)}}{=} (1 - a\lambda)\overline{\rho}_i^{n,e} + a\lambda\overline{\rho}_{i-1}^{n,e} \geq (1 - a\lambda)\rho_i^{n,e} + a\lambda\rho_{i-1}^{n,e} \overset{\text{(PDE)}}{=} \rho_i^{n+1,e},$$

  where $i > 1$. At the left boundary cell $i = 1$, we obtain

$$\overline{\rho}_1^{n+1,e} = (1 - a\lambda)\overline{\rho}_1^{n,e} + \lambda\overline{c}_{pre}^e \geq (1 - a\lambda)\rho_1^{n,e} + \lambda c_{pre}^e = \rho_1^{n+1,e}.$$

Finally, we obtain $\rho_i^{n+1,e} \leq \overline{\rho}_i^{n+1,e}$.
Now we prove $\overline{c}_{in}^e \leq c_{in}^e$. Therefore, we consider two different cases:

- **Case 1**: Let $F_1^{n,e} = \frac{\rho_{max} - \rho_2^{n,e}}{\lambda} + F_2^{n,e}$. Then (2.60) yields

$$\overline{c}_{in}^e \overset{(2.59)}{=} \frac{\rho_{max} - \overline{\rho}_1^{n,e}}{\lambda} + \widetilde{F}_1^{n,e} \leq \frac{\rho_{max} - \rho_1^{n,e}}{\lambda} + F_1^{n,e} \overset{(2.59)}{=} c_{in}^e$$

- **Case 2**: Now we consider the case $F_1^{n,e} = a\rho_1^{n,e}$. This results

$$\begin{aligned}
\overline{c}_{in}^e &= \frac{\rho_{max} - \overline{\rho}_1^{n,e}}{\lambda} + \widetilde{F}_1^{n,e} \leq \frac{\rho_{max} - \overline{\rho}_1^{n,e}}{\lambda} + a\overline{\rho}_1^{n,e} \\
&= \frac{1}{\lambda}(\rho_{max} - \underbrace{(1 - a\lambda)}_{\geq 0}\overline{\rho}_1^{n,e}) \leq \frac{1}{\lambda}(\rho_{max} - (1 - a\lambda)\rho_1^{n,e}) \\
&= \frac{\rho_{max} - \rho_1^{n,e}}{\lambda} + a\rho_1^{n,e} = \frac{\rho_{max} - \rho_1^{n,e}}{\lambda} + F_1^{n,e} = c_{in}^e.
\end{aligned}$$

□

Theorem 2.4.8. *The routine* `densityBoundStrengthening()` *computes upper and lower bounds of the density values* $\rho_i^{n,e}$.

Proof. We restrict the proof only for upper bounds $\overline{\rho}_i^{n,e}$. However, the proof for the lower bounds works analogously.

The density $\rho_i^{n,e}$ is computed by the algorithm `forwardsolutionPDE()`. The upper bounds $\overline{\rho}_i^{n,e}$ are computed by the `densityBoundStrengthening()`. However, the routine `densityBoundStrengthening()` is equivalent to the `forwardsolutionPDE()` algorithm that differs only from the coupling condition in junction type III.

We assume that $\rho_i^{n,e} \leq \overline{\rho}_i^{n,e}$ holds for a discrete time n. Then, the main goal of the proof is to show that

$$\rho_i^{n+1,e} \leq \overline{\rho}_i^{n+1,e} \quad \text{for all } i = 1, ..., N, \quad e \in E. \tag{2.62}$$

First, we show that the assumptions of Lemma 2.4.7 are fulfilled for all edges of the entire network, i.e., $c_{out}^e \geq \overline{c}_{out}^e$, $c_{pre}^e \leq \overline{c}_{pre}^e$ for all $e \in E$. Consequently, the statement (2.62) follows directly from the Lemma 2.4.7.

The outflow of an edge e depends obviously on the density values of the outgoing edges $\tilde{e} \in \delta^{out}(\{e\})$. That means that c_{out}^e depends also on $c_{pre}^{\tilde{e}}$ and $c_{in}^{\tilde{e}}$. Therefore, it is only possible to find informations about c_{out}^e if we find informations about $c_{pre}^{\tilde{e}}$ and $c_{in}^{\tilde{e}}$. This leads to an induction with respect to a topological ordering of

all edges $e \in E$.

The set of all edges E can be divided in disjoint subsets E^k if the network is free of cycles, i.e.,

$$E = \dot{\bigcup_{k\geq 0}} E^k, \quad E^k := \delta^{in}(E^{k-1}), \quad E^0 := E^{out}.$$

Note that the sequence $E^0, E^1, ..., E^k, ...$ corresponds to a topological ordering of the edges. Next, we proof that $c^e_{out} \geq \overline{c}^e_{out}$, $c^e_{in} \geq \overline{c}^e_{in}$, $c^e_{pre} \leq \overline{c}^e_{pre}$ for all $e \in E^k$ by induction with respect to k.
Induction Start: $(k = 0)$; We choose all outgoing edges $e \in E^0 := E^{out}$. The coupling (CPL D) yields

$$F^{n,e}_N = \min\{a\rho^{n,e}_N, f^n_{out}\}, \quad \widetilde{F}^{n,e}_N = \min\{a\overline{\rho}^{n,e}_N, f^n_{out}\},$$

Obviously, the outflow is limited by f^e_{out}, i.e.,

$$c^e_{out} = \overline{c}^e_{out} = f^e_{out}.$$

Additionally, Lemma 2.4.7 yields $\overline{c}^e_{in} \leq c^e_{in}$.

Induction hypothesis (IH): For all edges $e \in E^k$, it holds the following condition: $\overline{c}^e_{in} \leq c^e_{in}$.

Induction step: $(k \to k+1)$; $E^{k+1} := \delta^{in}(E^k)$. Let $e \in E^{k+1}$.
The aim is to show that $c^e_{out} \geq \overline{c}^e_{out}$ for $e \in E^{k+1}$ and $c^{\tilde{e}}_{pre} \leq \overline{c}^{\tilde{e}}_{pre}$ for $\tilde{e} \in E^k$.

Junction I: We consider a junction with one incoming edge $e = 1$ and one outgoing edge $e = 2$. The fluxes at the junction reveals

$$F^{n,2}_0 = F^{n,1}_N = \min\{\overbrace{a\rho^{n,1}_N}^{c^2_{pre}}, \overbrace{\frac{\rho_{max} - \rho^{n,2}_1}{\lambda} + F^{n,2}_2}^{c^2_{in}, c^1_{out}}\},$$
$$\widetilde{F}^{n,2}_0 = \widetilde{F}^{n,1}_N = \min\{\underbrace{a\overline{\rho}^{n,1}_N}_{\overline{c}^2_{pre}}, \underbrace{\frac{\overline{\rho}_{max} - \overline{\rho}^{n,2}_1}{\lambda} + \widetilde{F}^{n,2}_2}_{\overline{c}^2_{in}, \overline{c}^1_{out}}\}.$$

After identifying and comparison of the coefficients c^1_{out}, c^1_{in}, c^1_{pre}, ..., we obtain the following estimations

$$c^2_{pre} = a\rho^{n,1}_N \leq a\overline{\rho}^{n,1}_N = \overline{c}^2_{pre},$$
$$c^1_{out} = c^2_{in} \overset{IH}{\geq} \overline{c}^2_{in} = \overline{c}^1_{out}.$$

Lemma 2.4.7 yields $\overline{c}^1_{in} \leq c^1_{in}$.
Junction II: We consider a junction with two incoming edges $e = 1, 2$ and one outgoing edge $e = 3$.

$$F_N^{n,1} = \min\{a\rho_N^{n,1}, \overbrace{\frac{\rho_{max} - \rho_1^{n,3}}{\lambda} + F_2^{n,3}}^{c^1_{out}}\},$$
$$F_N^{n,2} = \min\{a\rho_N^{n,2}, \overbrace{\frac{\rho_{max} - \rho_1^{n,3}}{\lambda} + F_2^{n,3} - F_N^{n,1}}^{c^2_{out}}\},$$
$$F_0^{n,3} = \min\{\overbrace{a\rho_N^{n,1} + a\rho_N^{n,2}}^{c^3_{pre}}, \overbrace{\frac{\rho_{max} - \rho_1^{n,3}}{\lambda} + F_2^{n,3}}^{c^3_{in}}\}.$$

The fluxes $\widetilde{F}_N^{n,1}$,$\widetilde{F}_N^{n,2}$,$\widetilde{F}_0^{n,3}$ are computed analogously. A simple comparison of the coefficients yields

$$c^1_{out} = c^3_{in} \overset{IH}{\geq} \overline{c}^3_{in} = \overline{c}^1_{out}.$$

By assumption $\rho_N^{n,e} \leq \overline{\rho}_N^{n,e}$ for $e = 1, 2$ and by the induction hypothesis (IH), we obtain

$$c^2_{out} = c^3_{in} - F_N^{n,1} = c^3_{in} - \min\{a\rho_N^{n,1}, \overbrace{c^1_{out}}^{c^3_{in}}\} = \max\{c^3_{in} - a\rho_N^{n,1}, 0\}$$
$$\overset{IH}{\geq} \max\{\overline{c}^3_{in} - a\overline{\rho}_N^{n,1}, 0\} = \overline{c}^3_{in} - \min\{a\overline{\rho}_N^{n,1}, \overbrace{\overline{c}^1_{out}}^{\overline{c}^3_{in}}\} = \overline{c}^2_{out},$$

$$c^3_{pre} = a\rho_N^{n,1} + a\rho_N^{n,2} \leq a\overline{\rho}_N^{n,1} + a\overline{\rho}_N^{n,2} = \overline{c}^3_{pre}.$$

Junction III: We consider a junction with one incoming edges $e = 1$ and two outgoing edges $e = 2, 3$. The coupling conditions of the MIP model with respect to $\rho_i^{n,e}$ has the following form

$$\begin{aligned} F_N^{n,1} &= \min\{c_1^n, c_2^n + c_3^n\}, \\ F_0^{n,2} &\leq c_2^n, \\ F_0^{n,3} &\leq c_3^n, \end{aligned} \tag{2.63}$$

where c_1^n,c_2^n,c_3^n is defined as

$$c_1^n = a\rho_N^{n,1}, \quad c_2^n = \frac{\rho_{max} - \rho_1^{n,2}}{\lambda} + F_1^{n,2}, \quad c_3^n = \frac{\rho_{max} - \rho_1^{n,3}}{\lambda} + F_1^{n,3}.$$

By introduction of an unknown $d \in [0,1]$, we can generalize the conditions of (2.63) into

$$\begin{aligned} F_N^{n,1} &= \min\{c_1^n, \overbrace{c_2^n + c_3^n}^{c_{out}^1}\}, \\ F_0^{n,2} &= \min\{\overbrace{dc_1^n}^{c_{pre}^2}, \overbrace{c_2^n}^{c_{in}^2}\}, \\ F_0^{n,3} &= \min\{\overbrace{(1-d)c_1^n}^{c_{pre}^3}, \overbrace{c_3^n}^{c_{in}^3}\}. \end{aligned} \tag{2.64}$$

Obviously, (2.64) fulfills (2.63) for any $d \in [0,1]$. The coupling condition (CPL C^{UP}) is given by

$$\widetilde{F}_N^{n,1} = \min\{\overline{c}_1^n, \overbrace{\overline{c}_2^n + \overline{c}_3^n}^{\overline{c}_{out}^1}\}, \quad \widetilde{F}_0^{n,2} = \min\{\overbrace{\overline{c}_1^n}^{\overline{c}_{pre}^2}, \overbrace{\overline{c}_2^n}^{\overline{c}_{in}^2}\}, \quad \widetilde{F}_0^{n,3} = \min\{\overbrace{\overline{c}_1^n}^{\overline{c}_{pre}^3}, \overbrace{\overline{c}_3^n}^{\overline{c}_{in}^3}\},$$

where $\overline{c}_1^n$,$\overline{c}_2^n$,$\overline{c}_3^n$ are defined analogously to c_e^n for $e = 1,2,3$ by using $\overline{\rho}_i^{n,e}$ instead of $\rho_i^{n,e}$. We compare all maximal possible fluxes again and we obtain the following estimations:

$$\begin{aligned} c_{out}^1 = c_{in}^2 + c_{in}^3 &\overset{IH}{\geq} \overline{c}_{in}^2 + \overline{c}_{in}^3 = \overline{c}_{out}^1, \\ c_{pre}^2 = dc_1^n &\leq \overline{c}_1^n = \overline{c}_{pre}^2, \\ c_{pre}^3 = (1-d)c_1^n &\leq \overline{c}_1^n = \overline{c}_{pre}^3. \end{aligned}$$

Hence, the statements $c_{out}^e \geq \overline{c}_{out}^e$, $c_{in}^e \geq \overline{c}_{in}^e$, $c_{pre}^e \leq \overline{c}_{pre}^e$ are fulfilled for all $e \in E^{k+1}$. Consequently, the induction step is proven and thus the assumputions of Lemma 2.4.7 are fulfilled for all edges $e \in E$ and $i = 1, ..., N$. Finally, we obtain $\rho_i^{n+1,e} \leq \overline{\rho}_i^{n+1,e}$ for all $e \in E$ and $i = 1, ..., N$.
The proof for the lower bounds works analogously. □

Remark 2.4.9. *The presolve level 2 routine does not work for each network coupling condition. For instance, the routine fails for the network model that is introduced in [48]. In detail, the monotonicity property is not fulfilled for the merging junction case introduced in Remark 2.1.2. Consider the following counter-example:*
Let $\rho_1(x,0) = 0.3$, $\overline{\rho}_1(x,0) = 0.4$, $\rho_2(x,0) = \overline{\rho}_2(x,0) = 0.8$, *and* $\rho_3(x,0) = \overline{\rho}_3(x,0) = 0$ *be the initial data of the edges* $e = 1,2,3$ *. Additionally, the maximal density is* $\rho_{max} = 1$. *The coupling condition of Remark 2.1.2 leads to the following outflow:* $\gamma_1 = \gamma_2 = 0.3$, $\overline{\gamma}_1 = \overline{\gamma}_2 = 0.4$.
Thus, the solution on edge $e = 2$ *is a back traveling shock wave. However, the shock wave of solution* $\overline{\rho}_2(x,t)$ *moves slower than the shock wave of solution* $\rho_2(x,t)$. *Finally, we reveal* $\rho_2(x,t) \geq \overline{\rho}_2(x,t)$. *This is a contradiction to the monotonicity property* $\rho_2(x,t) \leq \overline{\rho}_2(x,t)$.

2.5 Numerical Results

In Subsection 2.5.1 we present the results of the forward simulation for the network problem. In particular, we investigate the properties of single junctions as well as complete networks. Additionally, we give a validation of the DFG method by comparison with the Godunov Method for the regularized problem (RFG). In Subsection 2.5.2 we compare the MIP model to a black box MATLAB optimization. Also, in Subsection 2.5.3 we highlight the computational efficiency of the presolving routines for the MIP model. All computations are performed on the same platform, namely a 3.0 GHz Dualcore computer with 8 GB RAM. The forward PDE solver and all presolve algorithms are implemented in MATLAB [83]. The MIP model is solved using the commercial solver ILOG CPLEX [71].

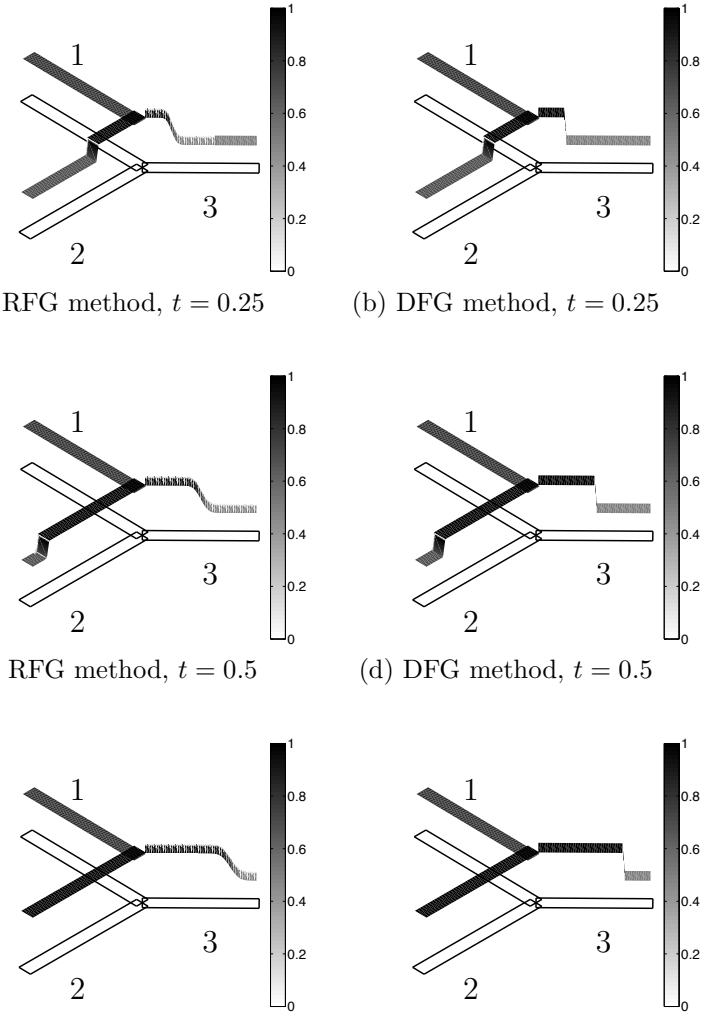
2.5.1 Network simulations

In the following, we present numerical results to validate and compare the DFG method to a classical Godunov method (RFG) for the regularized problem on networks. As already discussed in Chapter 1, the RFG scheme need very small time step size in the limit $\delta \to 0$ to produce qualitatively good solutions. We start with simulations for a network consisting of two types of junctions only: a merging type and a dispersing type. Depending on the numerical method, we refer to the corresponding coupling conditions. The regularization parameter δ is chosen arbitrary small for a valid approximation of the discontinuous case. In both studies, we are concerned with the quality of solutions and their interpretation. For the simulation we use the maximal density $\rho_{max} = 1$, the default velocity $a = 1$ and $\Delta x = \Delta t = 5 \cdot 10^{-3}$ for the DFG method and $\delta = 10^{-2}$ as well as $\Delta t = 5 \cdot 10^{-5}$ (cf. CFL condition) for the RFG scheme.

Merging Junction

At first, we consider a test case with two incoming and one outgoing edge. The initial values of the two incoming edges are chosen as $\rho_{1,0} = 0.8, \rho_{2,0} = 0.7$ and the outgoing edge is set to $\rho_{3,0} = 0.5$. Compared to the analytical investigations in Subsection 2.1.2 (cf. **Case A1**), we expect the resulting boundary densities $\bar{\rho_1} = 0.8$, $\bar{\rho_e} = 1$ for $e = 2, 3$ and the maximal outflow $\gamma_2 = \min\{0.7, 1-0.8\} = 0.2$ for the ingoing edge $e = 2$. We observe a backward traveling shock wave solution at edge $e = 2$ with speed $s_2 = -\frac{5}{3}$ due to formula $s_2 = \frac{\rho_{2,0}-\frac{1}{5}}{\rho_{2,0}-1}$. However, the inflow of the outgoing edge $e = 3$ is $\gamma_3 = 1$ where the left boundary value is $\bar{\rho_3} = 1$. The solution is a forward traveling shock wave with velocity $s_3 = 1$.
We compare the numerical solutions of the regularized network model computed by the RFG method and the DFG method for networks. The Riemann solutions discussed above are reproduced by both methods. The main difference is that the solution computed by the RFG method are smeared while the DFG method

gives exact results. There is also a great discrepancy considering the CPU times. The RFG method consumes 228.5 seconds in contrast to the DFG method that needs 0.2 seconds. The results are shown in Figure 2.13.



(a) RFG method, $t = 0.25$ (b) DFG method, $t = 0.25$

(c) RFG method, $t = 0.5$ (d) DFG method, $t = 0.5$

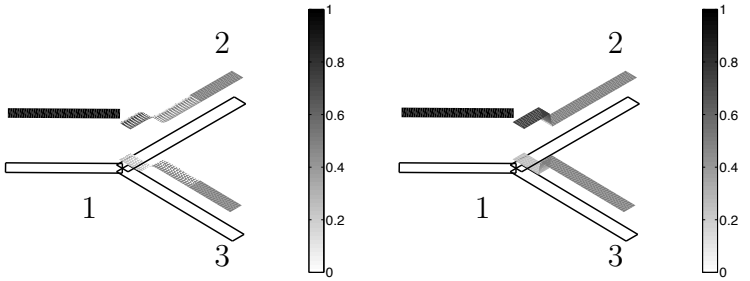
(e) RFG method, $t = 0.75$ (f) DFG method, $t = 0.75$

Figure 2.13: Results of the RFG and the DFG method.

Dispersing Junction

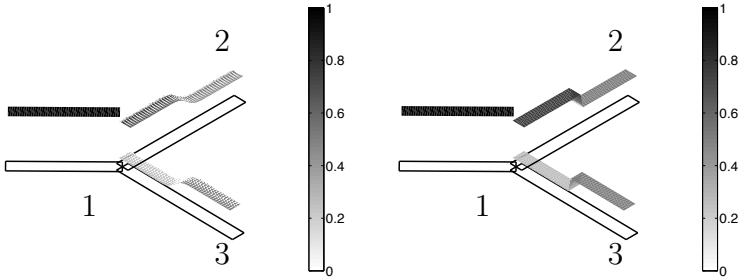
Next, we consider another junction configuration, i.e., one incoming edge $e = 1$ and two outgoing edges $e = 2, 3$ where we set the distribution rates $d_{2,1} = 0.75$, $d_{3,1} = 0.25$. We assume the following initial densities: $\rho_{1,0} = 1$ for the ingoing edge and and $\rho_{2,0} = \rho_{3,0} = 0.5$ for the outgoing ones. Corresponding to Subsection 2.1.2 (cf. **Case B2**), we get $\gamma_1 = \{1, \frac{4}{3}, 4\} = 1$, $\gamma_2 = \frac{3}{4}$ and $\gamma_3 = \frac{1}{4}$

with boundary densities $\bar{\rho_1} = 1$, $\bar{\rho_2} = \frac{3}{4}$ and $\bar{\rho_3} = \frac{1}{4}$. Then, the solution of edge 1 is a backward traveling shock wave with $s_1 = -\infty$ whereas the solution of edges $e = 2, 3$ are forward traveling shock waves with speed $s_e = 1$. The properties of the numerical solutions and the computing times behave like in the previous example and the simulation results are presented in Figure 2.14.



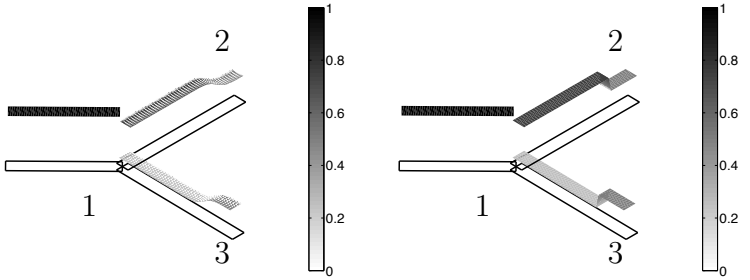
(a) RFG method, $t = 0.25$

(b) DFG method, $t = 0.25$



(c) RFG method, $t = 0.5$

(d) DFG method, $t = 0.5$



(e) RFG method, $t = 0.75$

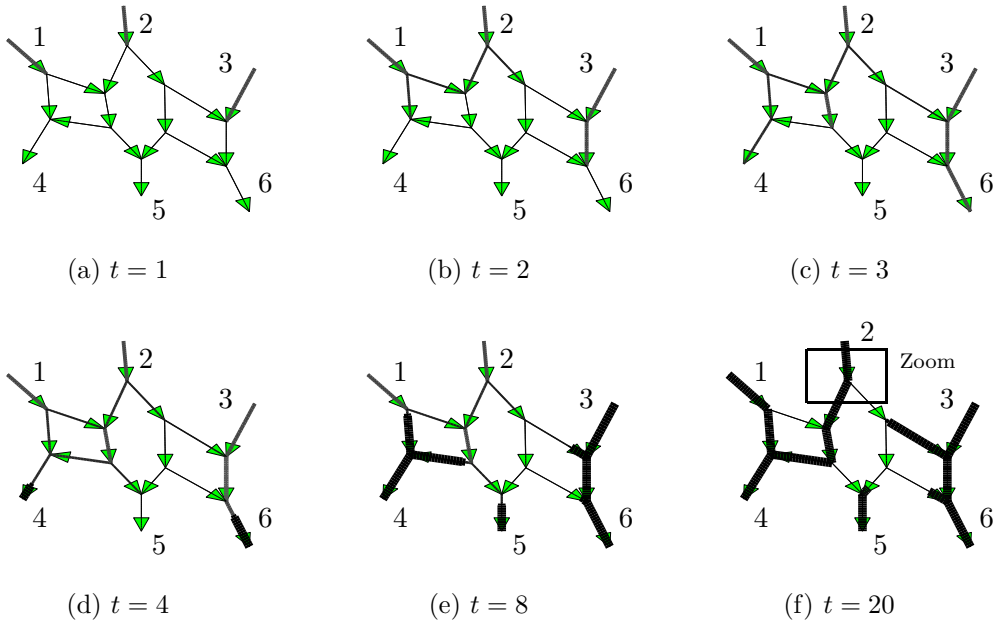
(f) DFG method, $t = 0.75$

Figure 2.14: Results of the RFG and the DFG method.

Network Sample

Finally, we combine our knowledge on merging and dispersing junctions and discuss a sample network. As we have seen, the CPU time of the DFG method is much faster and more precise than the RFG scheme. Therefore, we restrict to

the DFG method and the application to the discontinuous flux function. The following example shows simulation results of a connected network with 18 edges. The network structure is given in Figure 2.15. For simplicity, all relevant parameters are fixed to 1, i.e., $a_e = L_e = \rho_{max} = 1$ for all edges e. We choose a time horizon of $T = 20$ with the same discretization as before. The overall CPU time measures 3.5 seconds. The ingoing edges are defined by $e = 1, 2, 3$ with a constant inflow of $f(\rho_e(0,t)) = 0.4$, $\forall t$. We assume that the whole network is empty at time $t = 0$, i.e, $\rho_e(x, 0) = 0$ for all $x \in [0, 1]$ where each edge is mapped to the unit interval. Additionally, the outgoing edges of the network are assigned by $e = 4, 5, 6$. We assume that no goods leave the network, i.e., the outflow of edges $e = 4, 5, 6$ is equal to 0 and thus blocked. Wherever we have the freedom to distribute goods, we fix a constant distribution rate of 0.6 for edges going to the left and 0.4, respectively, for edges going to the right.
For the numerics, we realize the inflow by setting the numerical flux to $F_0^{n,e} = f(\rho_e(0,t)) = 0.4$ for $e = 1, 2, 3$ and analogously the outflow to $F_N^{n,e} = 0$ for $e = 4, 5, 6$. The simulation results are shown in Figure 2.15 for different times t. For time $t < 3$, we see that the flow of goods is spread over the complete network. Due to the blocking of the outflow edges, we recognize tailbacks (thick lines) in the network for $t > 3$.



(a) $t = 1$ (b) $t = 2$ (c) $t = 3$

(d) $t = 4$ (e) $t = 8$ (f) $t = 20$

Figure 2.15: Simulation of a supply-chain network for different times.

Apparently, not every edge is filled with the maximal density at time $t = 20$. This results from the coupling conditions derived in Section 2.1.2. Considering the zoom in Figure 2.15 at time $t = 20$, we know that for at least one outgoing

edge with $\rho_0 = 1$ we end up with zero fluxes γ_e and boundary densities $\bar{\rho}_e$ (cf. Remark 2.1.5). Similar situations occur at many nodes inside the network. In other words, whenever a tailback reaches an already blocked junction it is not possible to distribute parts any more. Hence, in contrast to the original intension that the network would be fully filled up by blocking edges $4, 5, 6$, we observe a steady state where some edges have maximal density, some are partly filled and some remain empty.

2.5.2 MATLAB Optimization vs. MIP Optimization

An optimization process is responsible in finding parameters of the system with respect to the objective function. There are many different approaches, but perhaps one of the easiest approach is black box optimization. It does not requires any significant knowledge about the system and works for each simulation process. The black box approach needs only an optimization routine and an objective function which includes the simulation process. Furthermore, the PDE simulation is repeated successively, until the optimization algorithm recognizes the solution as optimal.

Next, we apply the MATLAB routine `fminsearch()` to our black box optimization. Note that the routine `fminsearch()` finds the minimum of a scalar function by using the Nelder-Mead algorithm. Moreover, we are interested in finding the optimal distribution rates of the following network problem. To reduce additional computation times, the MATLAB optimization approach is restricted to constant distribution rates $d_{4,1}$, d_{5_1}. Finally, we compare the solutions to the results of the MIP model.

We use the network in Figure 2.16 with following initial datas. All edges of the network are empty at time $t = 0$, i.e., $\rho^e(x, 0) = 0$ for all $x \in [0, 1], e \in E$. The

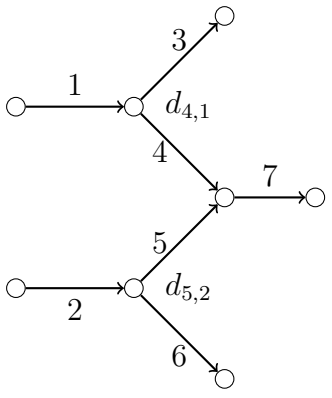


Figure 2.16: Network sample

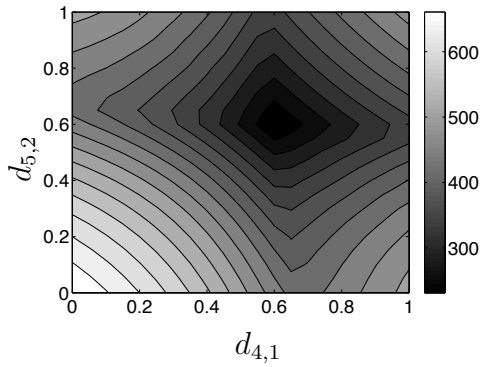


Figure 2.17: Plot of the cost functional

incoming edges $e = 1$ and $e = 2$ have an inflow of value $f^e_{in} = 0.25$. Additionally,

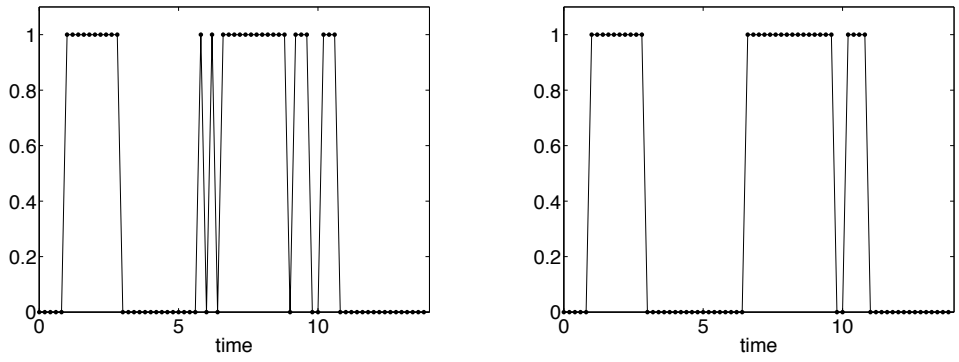
all outgoing edges $e = 3$, $e = 6$, and $e = 7$ have an outflow of zero, i.e,

$$f(\rho^e(0,t)) = 0.25, \qquad e \in \{1,2\},$$
$$f(\rho^e(1,t)) = 0, \qquad e \in \{3,6,7\}.$$

The time horizon is set to $T = 14$ and we use the step sizes $\Delta x = \Delta t = 0.2$. The objective function is formulated as

$$\sum_{e=1}^{2} \sum_{i=1}^{N} \sum_{n=1}^{N_T} \rho_i^{n,e} \to \min .$$

The boundary condition ensures that no quantity leaves the network. Furthermore, the objective function stays minimal if no back traveling shock wave (tailback) reaches the edges $e = 1$ and $e = 2$; i.e., the solution becomes optimal if the quantity is distributed completely to edges $e = 3, ..., 7$ and the quantity in edges $e = 1, 2$ is reduced to a minimum. Thus, this is possible if 60 % of the quantity is distributed to the edges $e = 4$ and $e = 5$.
The objective functional concerning the two distribution parameters is shown in Figure 2.17. Clearly, this problem has only one minimum. In due of the MATLAB approach, we choose $d_1^4 = d_1^5 = 0.5$ as an initial values for the optimization. After only 0.054 seconds of computation time, the MATLAB algorithm terminates with a default tolerance of 10^{-4}. The optimization approach yields the optimal distribution rates $d_1^4 = d_2^5 = 0.6$. If we change the initial values of the optimization to $d_1^4 = 0.2$, $d_1^5 = 0.8$, the MATLAB routine requires about 0.132 seconds of computation time.



(a) Distributionrate $d^{4,1}$ (b) Distributionrate $d^{5,1}$

Figure 2.18: Optimal Control by MIP

The MIP model reveals time-dependent distribution rates, cf. Figure 2.18. The computation time of the MIP model is about 11.43 seconds and is slower than

the computation time of the MATLAB optimization approach. However, the objective value of both solutions is equal, i.e., $J^* = 230$. As the MIP model leads to global optimal solutions, this MATLAB approach yields also a global optimal solution.

2.5.3 MIP Optimization with Presolving

In this experiment, we show and compare the efficiency of the presolving algorithms of Section 2.4. Hence, the main criteria is the degree of simplification of the MIP model for a faster computation. All results are computed with CPLEX [71] using default settings. Accordingly, the absolute MIP gap tolerance is set to 10^{-6}. Furthermore, all presented results (computed by CPLEX) are solved with a gap of 0.00%.

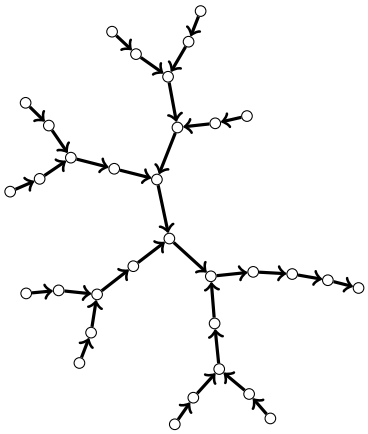


Figure 2.19: Network sample with serial and merging junctions

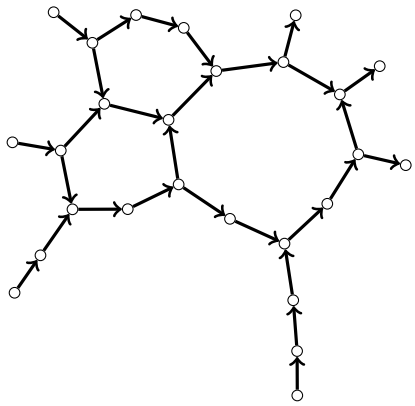


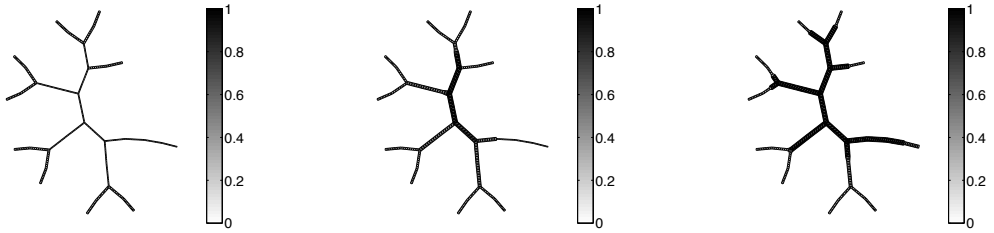
Figure 2.20: Network Sample

Merging Network

In this example, the evaluating network consists only of serial and merging junction types. This way, dispersing junctions are neglected in that case. Also, the inflow of the network is known and prescribed. The solution for the serial and merging junction type is uniquely defined. Hence, the underlying network problem has no degrees of freedom for any control and has a unique solution.
The network structure is given in Figure 2.19. The velocity and the maximal density is set to $a = \rho_{max} = 1$. Also, the network consists of 32 edges and all incoming edges have an inflow of $f^e_{in} = 0.3$. All outgoing edges have an outflow boundary, i.e., $f^e_{out} = 1$. The length of each edge is set to $L_e = 1$. The system is empty at time $t = 0$, i.e., $\rho^e(x, 0) = 0$ for all edges e and $x \in [0, 1]$. The time horizon is specified to $T = 8$. We test this scenario with step sizes $\Delta x = 0.1$ and

$\Delta x = 0.05$. Thereby, the time step size is always set to $\Delta t = \Delta x$. The objective function is formulated as

$$\sum_{e \in E} \sum_{i=1}^{N} \sum_{n=1}^{N_T} \rho_i^{n,e} \to \min .$$



(a) $t = 2$ (b) $t = 5$ (c) $t = 8$

Figure 2.21: Simulation of a supply-chain network for different times.

	Presolving time		CPLEX solving time	
$\Delta x = \Delta t$	PRE LVL 1	PRE LVL 2	NO PRE	PRE LVL 1 or LVL 2
0.1	0.22s	0.27s	2.21s	0.45s
0.05	0.46s	0.53s	128.50s	2.04s

Table 2.1: Computation times in seconds for MIP models with presolving.

The solution of the MIP model is illustrated in Figure 2.21. We observe that the quantity flows through the incoming edges into the center of the network. Then the density increases to the maximal value in the inner network. In Figure 2.21 (b), we recognize a tailback (thick lines) which spread through the complete network, cf. Figure 2.21 (c).
Presolve level 1 as well as Presolve level 2 solves all variables of the MIP model indicated that the Presolve algorithms find for all lower bounds and upper bounds which coincide. Hence, the presolving level 1 and level 2 yields the same result. Thereby, we solve the network problem with the presolve routine level 1 or 2 (PRE LVL 1 or 2) and without our presolve routines (NO PRE). The computation times are compared in Table 2.1. We notice that the preprocessed MIP needs less time than the MIPs without any preprocessing.

Minimizing Buffers

One obtains the following scenario. The network consists of 28 edges and is illustrated in Figure 2.20. All incoming edges have an inflow of $f_{in} = 0.6$. The

transport velocity and the maximal density is set to one, i.e., $a = \rho_{max} = 1$. The outgoing edges have an outflow without any congestions, i.e., $f_{out} = a\rho_{max} = 1$. We consider this scenario for four different time horizons, i.e., $T = 6, 8, 10, 12$. The objective function is formulated as

$$\sum_{e \in E} \sum_{i=1}^{N} \sum_{n=1}^{N_T} \rho_i^{n,e} \to \min .$$

Both presolve techniques compute a certain number of upper and lower bounds for variables which coincides. The amount of solved variables is shown in Table 2.3. The total number of variables of the MIP grows with increasing the time horizon T. However, the presolve level 1 solves only an identical number of variables, for example, about 10000 binary variables for all different time horizons. This is related to the fact that the presolving level 1 technique computes good bounds for the first time steps, but not for the later ones. Hence, there are more flow decisions on a network for increasing time, which cannot be solved by this presolving routine. In contrast to presolve level 1, the presolve level 2 routine computes more variables, in particular, the number of binaries, solved by the presolve level 2 routine, is much higher. The presolving level 2 calculates principally better bounds for the density variables. Bounds are used to determine feasible positions of possible existing tailbacks. This information is coupled to the binary variables $\xi_i^{n,e}$ of the MIP model.

	Presolving time		CPLEX solving time		
	PRE LVL 1	PRE LVL 2	NO PRE	PRE LVL 1	PRE LVL 2
$T = 6$	0.21s	0.28s	2.26s	2.17s	0.61s
$T = 8$	0.26s	0.33s	573.32s	424.91s	31.95s
$T = 10$	0.33s	0.41s	infeasible	47445.85s	331.32s
$T = 12$	0.48s	0.51s	infeasible	infeasible	2013.77s

Table 2.2: Computation times in seconds for MIP models with presolving.

We compute the MIP model with the commercial CPLEX solver. The computation times of the scenarios are shown in Table 2.2. Obviously, the presolving level 2 leads to faster computation times, as well as the presolving level 2 finds a higher number of binaries. However, the computation time of the MIP model rises enormously for any presolve technique with respect to the increasing time horizon.

T		PRE LVL 1			PRE LVL 2		
	Variables:	$\rho_i^{n,e}$	$F_i^{n,e}$	$\xi_i^{n,e}$	$\rho_i^{n,e}$	$F_i^{n,e}$	$\xi_i^{n,e}$
$T = 6$	Solved:	11048	12073	9862	12270	13144	14121
	Total:	17080	18480	15120	17080	18480	15120
	Average:	64.68%	65.33%	65.22%	71.84%	71.13%	93.39%
$T = 8$	Solved:	11380	12517	10178	13185	14086	17469
	Total:	22680	24640	20160	22680	24640	20160
	Average:	50.18%	50.80%	50.49%	58.13%	57.17%	86.65%
$T = 10$	Solved:	11380	12597	10178	13585	14486	20529
	Total:	28280	30800	25200	28280	30800	25200
	Average:	40.24%	40.90%	40.39%	48.04%	47.03%	81.46%
$T = 12$	Solved:	11280	12567	10124	13985	14886	23589
	Total:	33880	36960	30240	33880	36960	30240
	Average:	33.29%	34.00%	33.48%	41.28%	40.28%	78.00%

Table 2.3: Number of solved variables by preprocessing. A variable is solved if upper and lower bound coincide.

Chapter 3

Material Flow on Conveyor Belts

Achieving products in a manufacturing process with the same quality, optimum material utilization, and long-term profitability, the entire material flow through a manufacturing unit needs to be planned and controlled in detail. The required productivity and product flexibility in the process is thereby achieved by means of highly automated machining centers and production lines. The individual functionalities of the machine tools and processing units as well as the material flow must be considered over the complete production process.

In this chapter, we present three mathematical models for material flows on conveyor belts. The first one is a basic microscopic model, which tracks each part in the material flow system and uses Newton's law together with a detailed description of the acting forces to simulate the evolution of material distribution and density. This modeling approach is well known from molecular simulations, see e.g. [72] for a recent review. In the engineering community, models based on this or similar principles are state of the art for material flow simulation, see e.g. [90, 91, 103] as well as for other applications such as granular flow [24, 77], computer graphics [36, 89] or traffic flow [57]. In this work, we introduce only the basic concepts of microscopic modeling. For more details, we refer to [47, 67].

The other two mathematical models are based on macroscopic approaches. This implies that these approaches use an average quantity as density (parts per area), and the dynamic is prescribed by a material flux (parts per time). The first macroscopic model is a two-dimensional extension of the model that is introduced in Chapter 1. A phenomenological study of the microscopic model yields the second macroscopic model, based on a two-dimensional nonlocal hyperbolic partial differential equation (see [34] and the references therein for an overview). This model is especially suitable to provide first estimates on material flow and throughput rate of the production line. Similar ideas are used to rigorously derive macroscopic models from microscopic ones via kinetic models, see [49, 104, 105].

The chapter is structured as follows: A concept and short introduction of the microscopic model is mentioned in Section 3.1. In Section 3.2 we subsequently present two macroscopic models for material flow on conveyor belts. At first, we introduce the flow model that is a two-dimensional extension of the model al-

ready introduced in Chapter 1. Afterwards, we present the extended flow model that is an improvement of the previous flow model. In Section 3.3 we discuss two numerical solution approaches for the macroscopic models. In detail, we introduce a finite volume method with dimensional splitting in Subsection 3.3.1 and a discontinuous Galerkin approach in Subsection 3.3.2. Due to control problems for manufacturing systems, we investigate an optimization approach based on the extended flow model, see Section 3.4. Finally, numerical results are shown in Section 3.5. In particular, we compare the different presented numerical methods and validate the macroscopic models against real world data. In conclusion, some test cases are investigated.

3.1 Microscopic Modeling

In the microscopic material flow model the physical movement of each single particle or cargo on material flow elements is studied in a general setting, i.e., a 3-dimensional space. Each cargo is described as an unbounded rigid body with the corresponding mass and moment of inertia. The interactions between the cargo among themselves or cargo and conveyor belt are presented through the physical laws of contact mechanics [88]. This approach is mainly used in material sciences (see e.g. [40, 74]) or granular flow (see e.g. [78]). In the following, we review the well-established microscopic model in its standard formulation as it applies to the transport of cylindrical cargo on a conveyor belt (see Figure 3.1), where the cargo is separated by a rigid singularizer.

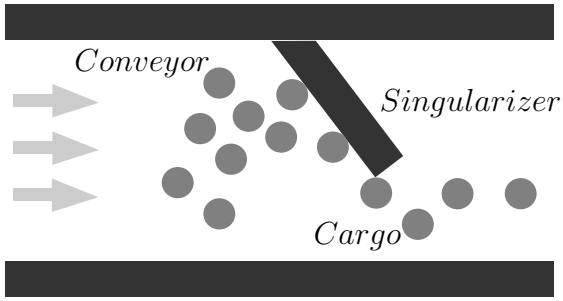


Figure 3.1: Cargo is separated on the conveyor belt by a rigid singularizer.

The material flow process is described as the sum of the unbounded movable cargo and the contact between other cargo and the material flow elements. The equation of motion for the movement of the cargo i is derived by means of Newton's law

of motion:

$$\frac{d\boldsymbol{x}_i(t)}{dt} = \boldsymbol{v}_i(t), \tag{3.1a}$$

$$m_i \frac{d\boldsymbol{v}_i(t)}{dt} = \sum_{n=1}^{N_{\boldsymbol{f}}} \boldsymbol{f}_{i,n}(t), \quad i = 1, \ldots, N_n, \tag{3.1b}$$

$$\boldsymbol{x}_i(0) = \boldsymbol{x}_{i,0}, \quad \boldsymbol{v}_i(0) = \boldsymbol{v}_{i,0}, \quad i = 1, \ldots, N_n, \tag{3.1c}$$

where $\boldsymbol{x}_i \in \mathbb{R}^3$ is the cargo position vector, $\boldsymbol{v}_i \in \mathbb{R}^3$ is the cargo velocity vector, $m_i \in \mathbb{R}^+$ is the cargo mass, N_n is the total number of cargo, $\boldsymbol{f}_{i,n} \in \mathbb{R}^3$ is the sum of $N_{\boldsymbol{f}}$ forces affecting the conveyed material. As example, contact forces, occurring friction forces and the gravitation can be used to specify the microscopic model. Note that there is no need to prescribe boundary conditions, as the effect of the boundaries is handled by the contact force that occurs when a cargo collides with the conveyor boundary.

Example: Contact Force

In the following, we introduce an example for a simple contact force for two colliding cargo objects. By observation, the cargo always lies on the conveyor belt if the conveyor belt velocity is quite slow. As a consequence, we can exclude an overlaying effect of cargo. This leads to the suitable assumption to locate the cargo only on a two dimensional plane. Also, we assume that the cargo objects are circle shaped with radius R. Additionally, we define the penetration depth $\delta_{i,j}$ of two interacting cargo objects i and j, i.e.,

$$\delta_{i,j} = 2R - \|\boldsymbol{x}_i - \boldsymbol{x}_j\|.$$

The contact force of two interacting cargo objects i and j is prescribed by

$$\boldsymbol{f}_{i,j}^{contact} = -\kappa \delta_{i,j} H(\delta_{i,j}) \frac{x_i - x_j}{\|x_i - x_j\|},$$

where H denotes the Heaviside function and κ is a constant that depends on the material property. An illustration of two colliding objects is given in Figure 3.2. The resulting contact force $\boldsymbol{f}_i^{contact}$ for a cargo object i is computed by the sum over all contact forces $\boldsymbol{f}_{i,j}^{contact}$, i.e.,

$$\boldsymbol{f}_i^{contact} := \sum_{i \neq j} \boldsymbol{f}_{i,j}^{contact}. \tag{3.2}$$

If the cargo objects i and j have no interaction, the corresponding penetration depth $\delta_{i,j}$ becomes negative and the contact force $\boldsymbol{f}_{i,j}^{contact}$ is zero. However, if the cargo objects interacts with each other, a repulsing force occurs that is linear proportional to the penetration depth of the cargo objects i and j.

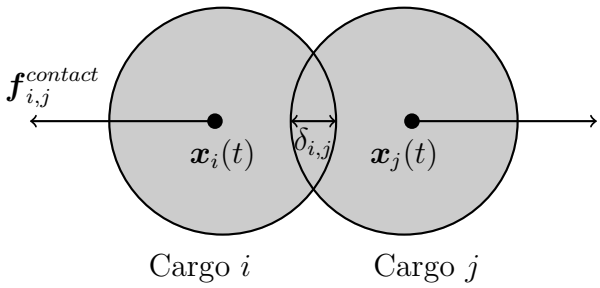


Figure 3.2: Interacting cargo objects

3.2 Macroscopic Modeling

Continuous models relying on conservation laws are used in different engineering areas, e.g. traffic flow [39], manufacturing systems [3], crowd and evacuation dynamics [20, 21].
We consider the setting illustrated in Figure 3.1 where we mainly assume that the number of cargo inside the system should be large. A singularizer is installed to redirect and sort the cargo to another position on the moving conveyor belt, i.e., phenomena such as queuing and changes of transport directions will occur. It is well known that microscopic models capture the most accurate dynamics but get computational extremely costly and produce inefficient simulation times. Clearly, the macroscopic approaches shall represent the right dynamical behavior of the material flow and provide suitable simulation times as well. This can be achieved using a macroscopic model avoiding the individual tracking of parts through the system using averaged quantities as part density (parts per area) and flux (parts per time). As an approximation, we propose two dimensional hyperbolic partial differential equations (PDEs), or conservation laws, which determine the motion of the part density.
To derive appropriate macroscopic models for the conveyor belt, the main ingredients and assumptions are:

(I) Mass should be conserved, i.e, we do not gain or lose cargo.

(II) The model must allow the formation of congestions at obstacles.

(III) Similar to traffic models, a maximal density is needed to deal with overcrowded situations.

3.2.1 The Flow Model

We set up an equation for the evolution of the part density at position and time. For simplicity the velocity field is given by a fixed and smooth vector field $\boldsymbol{v}^{stat}(\boldsymbol{x})$

describing the moving conveyor belt. Then, mathematically, the flow of material depends obviously on the density. Therefore, we introduce the part density as a two dimensional space and time depending function $\rho : \Omega \times \mathbb{R}^+ \to \mathbb{R}^+$, with $\Omega \subset \mathbb{R}^2$ that governs dynamics of the following setting:

$$\partial_t \rho + \nabla \cdot (f(\rho)\boldsymbol{v}^{stat}(\boldsymbol{x}))) = 0, \tag{3.3a}$$
$$f(\rho) = \rho \cdot H(\rho_{max} - \rho), \tag{3.3b}$$
$$\rho(\boldsymbol{x}, 0) = \rho_0(\boldsymbol{x}), \quad \boldsymbol{x} \in \mathbb{R}^2, \tag{3.3c}$$

where ρ_{max} is given as a user-defined constant (maximum possible number of parts), $\rho_0(\boldsymbol{x})$ is the initial distribution of parts and H denotes the Heaviside-function which is either 1 or 0. That means, in the first case, if $\rho(\boldsymbol{x}, t) < \rho_{max}$ parts do not collide and are transported with velocity $\boldsymbol{v}^{stat}$. Otherwise, if $\rho(\boldsymbol{x}, t) \geq \rho_{max}$, the parts are immediately redirected so that the density does not become higher than ρ_{max}, see (II) and (III). According to (3.3a), the transportation is modeled by a conservation law, i.e., no mass of parts are loss or gained in the system, and (I) is fulfilled.

Remark 3.2.1. *The boundary conditions of* (3.3a) *at* $\partial\Omega$ *are imposed by the geometry of the conveyor belt. We divide the boundary into two areas:*

$$\partial\Omega = \partial\Omega_{wall} \cup \partial\Omega_{inflow},$$

where $\partial\Omega_{wall}$ *describes solid boundaries and* $\partial\Omega_{inflow}$ *denotes the inflow region. At* $\partial\Omega_{inflow}$*, we set homogeneous Dirichlet conditions. Otherwise, at* $\partial\Omega_{wall}$*, we apply free slip conditions.*

$$\rho(\boldsymbol{x}, t) = 0, \qquad \boldsymbol{x} \in \partial\Omega_{inflow}, \tag{3.4a}$$
$$\langle \boldsymbol{v}^{stat}(\boldsymbol{x}), n \rangle = 0, \qquad \boldsymbol{x} \in \partial\Omega_{wall}, \tag{3.4b}$$

with n *being the normal vector to* $\partial\Omega$*.*
Note that in our experimental setting we do not need an inflow profile since all experiments are initialized with an initial distribution given by equation (3.3c)*.*

Static Velocity Field

The static field generates a direction field in $\mathbb{R}^2$ which models all trajectories of moving objects without self-interactions. As one can imagine, the field is motivated by the experiment introduced in Section 3.1. Ingredients such as the conveyor belt itself, the singularizer and the boundaries have to be represented in a correct way. Therefore, the static vector field is subdivided in different domains $A - C$, see Figure 3.3, left.
Each domain is assigned to a dominating vector. Domain A prescribes the movement of objects transported with the velocity of conveyor belt v_T. Thus, within

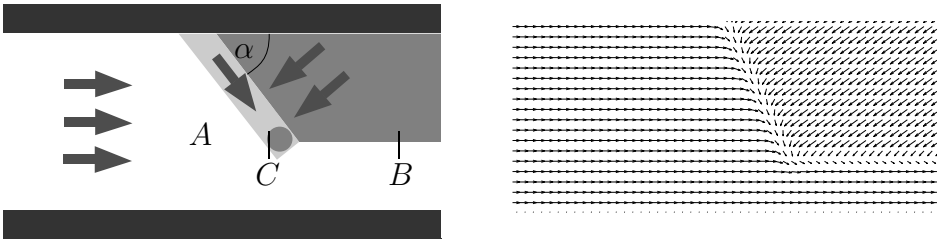


Figure 3.3: Static velocity field of the conveyor belt. Left picture: schematic view. Right picture: smoothed version for numerical simulations.

this area, the static field is defined as

$$\boldsymbol{v}^{stat}(\boldsymbol{x}) = v_T \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad \boldsymbol{x} \in A.$$

Domain B characterizes the shape of the singularizer. Since in reality it is not possible that objects get through the obstacle, the static field should prohibit trajectories intersecting the obstacle domain. This is done using an outgoing vector field, i.e., trajectories move out of the domain B. For that reason the dominating vector is directed to the normal of the obstacle surface.

$$\boldsymbol{v}^{stat}(\boldsymbol{x}) = v_T \begin{pmatrix} -\sin(\alpha) \\ \cos(\alpha) \end{pmatrix}, \quad \boldsymbol{x} \in B.$$

Generally, cargo move along the singularizer. For that reason, we introduce an additional velocity domain C to describe the slide effect at obstacles. The dimensions of the domain C are chosen such that its length corresponds to the length of the singularizer, while its width corresponds to the diameter of one of the objects that are transported on the conveyor belt. In this domain, the dominating vector is therefore given by:

$$\boldsymbol{v}^{stat}(\boldsymbol{x}) = v_T \begin{pmatrix} \cos(\alpha)\cos(\alpha) \\ \sin(\alpha)\cos(\alpha) \end{pmatrix}, \quad \boldsymbol{x} \in C.$$

Remark 3.2.2. *Note that walls can also be integrated in the static velocity field $\boldsymbol{v}^{stat}(\boldsymbol{x})$. For instance, consider the construction of domain B and use the normal vector n of the walls as the dominating vector of the domain.*

To avoid problems of well-posedness as well as stability issues in the numerical simulations, we use a smoothed version of the above described velocity field in all numerical experiments. The concrete static velocity field used in the simulations is displayed in Figure 3.3 (right picture).
The static velocity field prescribes the movement of non-colliding parts. According to the flow model, the transportation of the quantity moves always along the direction of the static velocity field $\boldsymbol{v}^{stat}$, although the parts interact with

each other. In the latter case, this behavior is quite unrealistic in some cases, and we expect an redirection of the velocity of interacting objects. Therefore, we introduce an improvement of the previous flow model that contains an additional velocity field for interacting cargo objects.

3.2.2 The extended Flow Model

The main idea of the model extension is a conservation law with a mass-dependent velocity field, cf. [20, 21]. The corresponding PDE which is in fact a conservation law can be stated as

$$\partial_t \rho + \nabla \cdot (\rho(\boldsymbol{v}^{dyn}(\rho) + \boldsymbol{v}^{stat}(\boldsymbol{x}))) = 0, \tag{3.5a}$$

$$\boldsymbol{v}^{dyn}(\rho) = H(\rho - \rho_{max}) \cdot \boldsymbol{I}(\rho), \tag{3.5b}$$

$$\boldsymbol{I}(\rho) = -\epsilon \frac{\nabla(\eta * \rho)}{\sqrt{1 + \|\nabla(\eta * \rho)\|_2^2}}, \tag{3.5c}$$

$$\rho(\boldsymbol{x}, 0) = \rho_0(\boldsymbol{x}), \quad \boldsymbol{x} \in \mathbb{R}^2, \tag{3.5d}$$

where $\rho = \rho(\boldsymbol{x}, t)$, H denotes the common Heaviside function assigning zero to negative arguments and ρ_{max} the fixed maximal density.
Corresponding to (I), equation (3.5a) determines the evolution of the initial part density (3.5d) depending on the velocity field consisting of two parts: the time-independent velocity field $\boldsymbol{v}^{stat}(\boldsymbol{x})$ and a dynamic velocity field $\boldsymbol{v}^{dyn}(\rho)$. The field $\boldsymbol{v}^{stat}(\boldsymbol{x})$ prescribes the transport velocity induced by the conveyor belt. Thus $\boldsymbol{v}^{stat}(\boldsymbol{x})$ defines the velocity field of single cargo without any interaction between each other. Note that $\boldsymbol{v}^{stat}(\boldsymbol{x})$ is already introduced in Subsection 3.2.1. However, the dynamic component $\boldsymbol{v}^{dyn}(\rho)$ in equation (3.5c) reflects the movement of colliding objects, similar to [20, 21]. We assume that the objects never move out of the x_1, x_2-plane, i.e., objects cannot overlay in the third dimension. By observation (II), the parts accumulate at the singularizer. But in reality colliding objects do not penetrate each other. This implies that the density could not be larger than the density of a close-packing of parts ρ_{max}, see (III). That means, we have to prevent situations that yield densities $\rho > \rho_{max}$ for $\rho_0(x_1, x_2) < \rho_{max}$ in a certain time $t > 0$ and space $\boldsymbol{x} \in \mathbb{R}^2$. This scenario is relevant if the divergence of the velocity field $\nabla \cdot \boldsymbol{v}^{stat}(\boldsymbol{x})$ is negative and $\rho > 0$. To ensure that the density ρ does not become much larger than ρ_{max} the density dependent velocity $\boldsymbol{v}^{dyn}(\rho)$ is introduced to reduce this effect. The velocity field $\boldsymbol{v}^{dyn}(\rho)$ disperses clouds with $\rho > \rho_{max}$. Thus, further compressions are prevented and the density does not exceed ρ_{max} anymore. The term (3.5c) is obviously active if $\rho > \rho_{max}$, i.e., $H(\rho - \rho_{max}) = 1$, and 0 (inactive) vice versa.
We introduce the non-local operator $\boldsymbol{I}(\rho)$ that is controllable with the constant parameter $\epsilon > 0$. The negative gradient field yields the steepest descent of the convolution $\eta * \rho$, where η is a sufficiently smooth function with the properties

$\int_{\mathbb{R}^2} \eta(\boldsymbol{x})\, d\boldsymbol{x} = 1$ and $\lim_{\varepsilon \to 0} \frac{1}{\varepsilon^2} \eta\left(\frac{\boldsymbol{x}}{\varepsilon}\right) = \delta(\boldsymbol{x})$, where $\delta(\boldsymbol{x})$ is the Dirac delta distribution. Such a function is also called a mollifier or smoothing function. The denominator of $\boldsymbol{I}(\rho)$ ensures that the vector norm is bounded, i.e., $\|\boldsymbol{I}(\rho)\|_2 \leq \epsilon$. Consequently, the parts feel a force pushing them in direction to a lower density. Moreover, inside a fully compressed cloud, the density is constant in space and therefore the term $\nabla(\eta * \rho)$ does not give any contribution to the force field. This is in accordance with the physical behavior where the forces inside the congested region sum up to zero. Thus, the density dependent force term $\boldsymbol{I}(\rho)$ will only act in a small neighborhood of the boundary of a congested region.

Let us summarize: The friction force between the parts and the conveyor belt implies a strong damping effect. Thus, in reality, the velocity of non-colliding parts converge to the transport velocity of the conveyor belt quite fast. In the macroscopic model, due to the Heaviside function, the non-interacting (or free flow) velocity is immediately $\boldsymbol{v}^{stat}(\boldsymbol{x})$. This is possible because the macroscopic model does not consider any inertia. On the other hand, if parts interact in the microscopic model, a contact force (3.2) will appear which repulses interacting parts. In the macroscopic model, a dispersing velocity field $\boldsymbol{v}^{dyn}(\rho)$ is activated which has a repulsive effect by the term $\boldsymbol{I}(\rho)$.

3.3 Numerical Methods

Now we present suitable numerical methods for the partial differential equations (3.3) and (3.5). The first approach is based on a one dimensional finite volume method which is extended into a two dimensional problem solver by dimensional splitting. The other approach is a discontinuous Galerkin method which is useful to compute accurate solutions on complex geometries.

3.3.1 Finite Volume Approach with Dimensional Splitting

The following procedures are based on the finite volume methods with dimensional splitting, see [79]. The computation works with a discrete data set of the density and velocity in space and time. The two dimensional spatial domain is discretized equidistantly in rectangular cells. Each cell is identified by the indices i, j. The center of a cell i, j is located at $\boldsymbol{x}_{i,j} = (x_{1,i}, x_{2,j})^T$. The lengths of the cells are given by the spatial step sizes Δx_1, Δx_2. Additionally the time t is discretized by step size Δt. We use the following space and time grid:

$$x_{1,i} = i\Delta x_1,\ i = 1, .., N_{x_1},\ x_{2,j} = j\Delta x_2,\ j = 1, .., N_{x_2},\ t_k = k\Delta t,\ k = 1, .., N_t.$$

The cells are presented as $Q_{i,j} = [x_{1,i-\frac{1}{2}}, x_{1,i+\frac{1}{2}}] \times [x_{2,j-\frac{1}{2}}, x_{2,j+\frac{1}{2}}]$. Note that for numerical simulations the spatial domain is bounded and has a rectangular shape.

Furthermore $\lambda_d = \frac{\Delta t}{\Delta x_d}$ for $d = 1, 2$ are the grid constants. The density ρ is now defined as a step function

$$\rho(\boldsymbol{x}, t_k) = \rho_{i,j}^k \in \mathbb{R} \text{ for } \boldsymbol{x} \in Q_{i,j}.$$

A common way to solve two dimensional problems is the application of a dimensional splitting, i.e., a fractional-step approach in which one-dimensional problems are solved sequentially along each coordinate direction. In that way the multidimensional problem is split into a sequence of one dimensional problems.

Flow Model

Note that the flux function $f(\rho)$ contains the discontinuous heaviside function H. By analogy to the RFG method in Subsection 1.3.1, we regularize the flux f for the presented numerical method, i.e.,

$$f_\delta(\rho) = \min\{\rho, \frac{1}{\delta}(\rho_{max} - \rho)\} \quad \text{for } \delta > 0.$$

The multidimensional problem 3.3a is split into a sequence of one dimensional problems. More concretely this means: Compute the problem

$$\partial_t \rho + \partial_{x_1}(f_\delta v_1^{stat}) = 0$$

by a finite volume method (e.g. Godunov) in the x_1-direction for one time step Δt. Subsequently, compute the problem in the x_2-direction for the time step Δt, i.e.,

$$\partial_t \rho + \partial_{x_2}(f_\delta v_2^{stat}) = 0.$$

This procedure leads to the following scheme:

`macro_solver`()

(1.1) **For** $k = 0$ **to** $N_t - 1$
(1.2) **For** $j = 1$ **to** N_{x_2}
(1.3) **For** $i = 1$ **to** N_{x_1}
(1.4) $F_1^+ := F^G(\rho_{i,j}^k, \rho_{i+1,j}^k, v_1^{stat}(\boldsymbol{x}_{i+\frac{1}{2}}, j))$
(1.5) $F_1^- := F^G(\rho_{i-1,j}^k, \rho_{i,j}^k, v_1^{stat}(\boldsymbol{x}_{i-\frac{1}{2}}, j))$
(1.6) $\tilde{\rho}_{i,j}^k = \rho_{i,j}^k - \lambda_1[F_1^+ - F_1^-]$
(1.7) **End**
(1.8) **End**
(1.9) **For** $i = 1$ **to** N_{x_1}
(1.10) **For** $j = 1$ **to** N_{x_2}

(1.11) $F_2^+ := F^G(\tilde{\rho}_{i,j}^k, \tilde{\rho}_{i,j+1}^k, v_2^{stat}(\boldsymbol{x}_{i,j+\frac{1}{2}}))$
(1.12) $F_2^- := F^G(\tilde{\rho}_{i,j-1}^k, \tilde{\rho}_{i,j}^k, v_2^{stat}(\boldsymbol{x}_{i,j-\frac{1}{2}}))$
(1.13) $\rho_{i,j}^{k+1} = \tilde{\rho}_{i,j}^k - \lambda_2[F_2^+ - F_2^-]$
(1.14) **End**
(1.15) **End**
(1.16) **End**

The numerical flux F^G is known as the Godunov flux, i.e.,

$$F_G(\rho_{i,j}^k, \rho_{i+1,j}^k, v_1^{stat}(\boldsymbol{x}_{i+\frac{1}{2}}, j) := \begin{cases} \min_{w\in[\rho_{i,j}^k, \rho_{i+1,j}^k)} v_1^{stat}(\boldsymbol{x}_{i+\frac{1}{2}}, j) f_\delta(w), & \rho_{i,j}^k \leq \rho_{i+1,j}^k, \\ \max_{w\in[\rho_{i+1,j}^k, \rho_{i,j}^k)} v_1^{stat}(\boldsymbol{x}_{i+\frac{1}{2}}, j) f_\delta(w), & \rho_{i,j}^k \geq \rho_{i+1,j}^k. \end{cases}$$

More details of the previous one-dimensional scheme with an spatial dependent velocity field are found in [96].

Remark 3.3.1. *As we have seen in Chapter 1, the discontinuous flux Godunov method (DFG) yields more efficient results for one-dimensional problems than the regularized flux Godunov method (RFG). Thus, the step size restriction does not depend on the regularization parameter δ, and the shock waves are drawn in an accurate way. A naive approach for a numerical scheme of the two-dimensional flow model could be a splitting method combined with the DFG method. However, such a splitting method does not work in practice.*
The problem starts if $\rho_{i,j}^k = \rho_{max}$. Flux information in direction x_1 of a cell with density ρ_{max} do not depend on the flux information in direction x_2 and vice versa. Note that the DFG flux for a cell with density ρ_{max} depends only on the choice of the succeeding cells in one direction and the boundary outflow.

Extended Flow Model

By analogy to the previous scheme, the multidimensional problem (3.5) is split into a sequence of one dimensional problems. Therefore, the fluxes $\rho(\boldsymbol{v}^{dyn}(\rho) + \boldsymbol{v}^{stat}(\boldsymbol{x}))$ used in the numerics are split in each dimension. The gradient and the convolution are parts of the dispersive term $\boldsymbol{I}(\rho)$. It is necessary to discuss the gradient and the convolution for the numerical solution method. In detail, the gradient of the convolution term $\eta * \rho$ is a two dimensional vector where the gradient operator can be directly applied to the mollifier η. This eliminates the differential operator ∇ if the function $\nabla\eta$ is well-known.

$$\nabla(\eta * \rho) = (\partial_{x_1}\eta * \rho, \partial_{x_2}\eta * \rho)^T. \tag{3.6}$$

For clarification, we consider only the first component of the vector (3.6). For the numerical method it is necessary to evaluate the flux between the cells. For

that reason we compute the convolution in the spatial point $\boldsymbol{x} = (x_{1,i+\frac{1}{2}}, x_{2,j})^T$ at a fixed time t_k.

$$(\partial_{x_1}\eta * \rho)(\boldsymbol{x}) = \int_{\mathbb{R}^2} \partial_{x_1}\eta(\boldsymbol{x}-\boldsymbol{\tau})\rho(\boldsymbol{\tau})d\boldsymbol{\tau} \tag{3.7a}$$

$$= \sum_{p,q} \rho^k_{p,q} \int_{Q_{p,q}} \partial_{x_1}\eta(\boldsymbol{x}-\boldsymbol{\tau})d\boldsymbol{\tau} \tag{3.7b}$$

$$= \sum_{p,q} \rho^k_{p,q} \cdot c^1_{i-p,j-q}, \tag{3.7c}$$

where the weights $c^d_{p,q}$ are defined as

$$c^1_{p,q} := \int_{Q_{p+\frac{1}{2},q}} \partial_{x_1}\eta(\boldsymbol{\tau})d\boldsymbol{\tau}, \quad c^2_{p,q} := \int_{Q_{p,q+\frac{1}{2}}} \partial_{x_2}\eta(\boldsymbol{\tau})d\boldsymbol{\tau}. \tag{3.8}$$

As an analogy to the first component of the vector (3.6), the computation of $(\partial_{x_1}\eta * \rho)$ yields the weights $c^2_{p,q}$.

Remark 3.3.2. *The expression* (3.7c) *is formulated as an infinite sum. For the numerical implementations, the sum is considered in a finite way with $S_1 \cdot S_2$ summands.*

The numerical flux in one dimension, i.e., $d = 1$ at points $\boldsymbol{x}_{i+\frac{1}{2},j}$ and t_k is a modified Roe flux combined with the non local term $\boldsymbol{I}(\rho)$:

$$F_1(\rho, \rho^k_{i,j}, \rho^k_{i+1,j}, \boldsymbol{x}_{i+\frac{1}{2},j}) = \begin{cases} \rho^k_{i,j} H(\rho^k_{i,j} - \rho_{max}) I_1(\rho)(\boldsymbol{x}_{i+\frac{1}{2},j}), & I_1(\rho)(\boldsymbol{x}_{i+\frac{1}{2},j}) \geq 0 \\ \rho^k_{i+1,j} H(\rho^k_{i+1,j} - \rho_{max}) I_1(\rho)(\boldsymbol{x}_{i+\frac{1}{2},j}), & I_1(\rho)(\boldsymbol{x}_{i+\frac{1}{2},j}) \leq 0. \end{cases}$$

$I_1(\rho)$ respectively $I_2(\rho)$ are the first and second components of the vector $\boldsymbol{I}(\rho)$. Furthermore, the static flux is chosen as the classical Roe flux

$$G_1(\rho^k_{i,j}, \rho^k_{i+1,j}, \boldsymbol{v}^{stat}_{i+\frac{1}{2},j}) = \begin{cases} \rho^k_{i,j} v^{stat}_{1,i+\frac{1}{2},j}, & v^{stat}_{1,i+\frac{1}{2},j} \geq 0 \\ \rho^k_{i+1,j} v^{stat}_{1,i+\frac{1}{2},j}, & v^{stat}_{1,i+\frac{1}{2},j} \leq 0, \end{cases}$$

where the discretized static velocity field is given by

$$\boldsymbol{v}^{stat}_{i+\frac{1}{2},j} := (v^{stat}_{1,i+\frac{1}{2},j}, v^{stat}_{2,i+\frac{1}{2},j})^T := \boldsymbol{v}^{stat}(\boldsymbol{x}_{i+\frac{1}{2},j}).$$

The fluxes in x_2-direction $F_2(\rho, \rho^k_{ij}, \rho^k_{i,j+1}, \boldsymbol{x}_{i,j+\frac{1}{2}})$ and $G_2(\rho^k_{i,j}, \rho^k_{i,j+1}, \boldsymbol{v}^{stat}_{i,j+\frac{1}{2}})$ are defined analogously.
The routine `extended_macro_solver()` describes a numerical solver for the extended flow model. The dynamic velocity field is solved explicitly for time t_k in the routine `compute_velocityfield(...)`. The static velocity field is time

invariant and an update routine with respect to time is redundant. In lines 1.3 - 1.16 in `extended_macro_solver()`, the continuity equation for the velocity field $(\boldsymbol{v}^{dyn}(\rho) + \boldsymbol{v}^{stat}(x))$ is solved for the next time step t_{k+1} by dimension splitting.

`extended_macro_solver()`

(1.1) **For** $k = 0$ **to** $N_t - 1$
(1.2) `compute_velocityfield()`
(1.3) **For** $j = 1$ **to** N_{x_2}
(1.4) **For** $i = 1$ **to** N_{x_1}
(1.5) $F_1^+ := F_1(\rho, \rho_{i,j}^k, \rho_{i+1,j}^k, \boldsymbol{x}_{i+\frac{1}{2},j}) + G_1(\rho_{i,j}^k, \rho_{i+1,j}^k, \boldsymbol{v}_{i+\frac{1}{2},j}^{stat})$
(1.6) $F_1^- := F_1(\rho, \rho_{i-1,j}^k, \rho_{i,j}^k, \boldsymbol{x}_{i-\frac{1}{2},j}) + G_1(\rho_{i-1,j}^k, \rho_{i,j}^k, \boldsymbol{v}_{i-\frac{1}{2},j}^{stat})$
(1.7) $\tilde{\rho}_{i,j}^k = \rho_{i,j}^k - \lambda_1[F_1^+ - F_1^-]$
(1.8) **End**
(1.9) **End**
(1.10) **For** $i = 1$ **to** N_{x_1}
(1.11) **For** $j = 1$ **to** N_{x_2}
(1.12) $F_2^+ := F_2(\rho, \tilde{\rho}_{i,j}^k, \tilde{\rho}_{i,j+1}^k, \boldsymbol{x}_{i,j+\frac{1}{2}}) + G_2(\tilde{\rho}_{i,j}^k, \tilde{\rho}_{i,j+1}^k, \boldsymbol{v}_{i,j+\frac{1}{2}}^{stat})$
(1.13) $F_2^- := F_2(\rho, \tilde{\rho}_{i,j-1}^k, \tilde{\rho}_{i,j}^k, \boldsymbol{x}_{i,j-\frac{1}{2}}) + G_2(\tilde{\rho}_{i,j-1}^k, \tilde{\rho}_{i,j}^k, \boldsymbol{v}_{i,j-\frac{1}{2}}^{stat})$
(1.14) $\rho_{i,j}^{k+1} = \tilde{\rho}_{i,j}^k - \lambda_2[F_2^+ - F_2^-]$
(1.15) **End**
(1.16) **End**
(1.17) **End**

`compute_velocityfield()`

(2.1) **For all** i, j
(2.2) $D_{x_1}\rho_{i,j} := \sum_{p,q} \rho_{p,q}^k \cdot c_{i-p,j-q}^1$
(2.3) $D_{x_2}\rho_{i,j} := \sum_{p,q} \rho_{p,q}^k \cdot c_{i-p,j-q}^2$

(2.4) $I_1(\rho)(\boldsymbol{x}_{i+\frac{1}{2},j}) = -\epsilon \frac{D_{x_1}\rho_{i,j}}{\sqrt{1+(D_{x_1}\rho_{i,j})^2+(D_{x_2}\rho_{i,j})^2}}$
(2.5) $I_2(\rho)(\boldsymbol{x}_{i,j+\frac{1}{2}}) = -\epsilon \frac{D_{x_2}\rho_{i,j}}{\sqrt{1+(D_{x_1}\rho_{i,j})^2+(D_{x_2}\rho_{i,j})^2}}$
(2.6) **End**

Remark 3.3.3. *Some properties of the previous numerical scheme.*

1. *If the integrals* (3.8) *are evaluated exactly, the convolution of the discretized density* ρ *is also exact. Thus, the corresponding dynamic velocity field* $\boldsymbol{v}^{dyn}$ *is evaluated exactly for the discretized density* ρ*.*

2. *Using the notation* $\rho(\boldsymbol{v}^{dyn}(\rho) + \boldsymbol{v}^{stat}(\boldsymbol{x})) = (\mathcal{F}_1(\rho, \boldsymbol{x}), \mathcal{F}_2(\rho, \boldsymbol{x}))^T$*, we note that the above discrete flux fulfills*

$$F_1(\bar{\rho}, \bar{\rho}, \bar{\rho}, \boldsymbol{x}_{i+\frac{1}{2},j}) + G_1(\bar{\rho}, \bar{\rho}, \boldsymbol{v}^{stat}_{i+\frac{1}{2},j}) = \mathcal{F}_1(\bar{\rho}, \boldsymbol{x}_{i+\frac{1}{2},j}),$$
$$F_2(\bar{\rho}, \bar{\rho}, \bar{\rho}, \boldsymbol{x}_{i,j+\frac{1}{2}}) + G_2(\bar{\rho}, \bar{\rho}, \boldsymbol{v}^{stat}_{i,j+\frac{1}{2}}) = \mathcal{F}_2(\bar{\rho}, \boldsymbol{x}_{i,j+\frac{1}{2}})$$

   for all $\bar{\rho} \in \mathbb{R}^+$*. This is necessary to get a consistent discretization of the continuous flux* $\rho(\boldsymbol{v}^{dyn}(\rho) + \boldsymbol{v}^{stat}(\boldsymbol{x}))$*.*

3. *The presented method is positive preserving as long as the grid constants fulfill* $\lambda_d < \frac{1}{2(\epsilon + \max\{v_d^{stat}\})}$*. Indeed, let* $\rho_{ij}^k > 0$ *for all* i, j*. Then we have* $|I_d(\rho)| \leq \epsilon$ *and can conclude*

$$F_1^+ :=\leq \rho_{ij}^k(\epsilon + v^{stat}_{1,i+\frac{1}{2},j}),$$
$$F_1^- :=\geq -\rho_{ij}^k(\epsilon + v^{stat}_{1,i-\frac{1}{2},j}).$$

   Therefore,

$$\tilde{\rho}_{ij}^k := \rho_{ij}^k - \lambda_1 \left(F_1^+ - F_1^-\right) > 0.$$

   Analogous arguments applied to $F_2^\pm$ *yield* $\rho_{ij}^{k+1} > 0$*.*

Let us now analyze the complexity of the numerical method for the extended flow model.

Lemma 3.3.4 (Macroscopic model: Runtime performance)**.** *Let the computation times of a single operation be defined by*
c_1*: Floating Point Addition and Subtraction,*
c_2*: Floating Point Multiplication and Division,*
c_3*: Comparison,*
c_4*: Trigonometric, Square root and Pow operations,*
c_5*: Negation,*
c_6*: Jump operation,*
c_7*: Assignment,*
c_8*: Integer Increment/Decrement.*
Then the runtime computation time T_{run}^{macro} *of the algorithm is assessable by the formula*

$$\begin{aligned} T_{run}^{macro} = \quad & N_t \cdot (N_{x_1} N_{x_2} (S_1 S_2 (2c_1 + c_2 + 2c_3 + 4c_6 + 2c_7 + 2c_8) \\ & + 16c_1 + 24c_2 + 13c_3 + 2c_5 + 3c_6 + 5c_7 + 3c_8)). \end{aligned} \tag{3.9}$$

Proof. Each iteration of a **For** loop costs a comparison, jump operation, assignment and an integer increase. A **For** loop with N_t iterations needs the following computation time

$$T_{Loop} = N_t(c_3 + c_6 + c_7 + c_8).$$

We estimate the computation time of the procedure `compute_velocityfield`(). The expressions in line 2.2 and line 2.3 have $2 \cdot S_1S_2$ additions, multiplications and assignments. Note that this calculation is implemented with two convoluted **For** loops. Line 2.4 and line 2.5 have 4 additions, 6 multiplications, 2 square root operations, 2 negations, 2 assignments. The **For** loop in line 2.1 repeats this computation $N_{x_1}N_{x_2}$ times. This yields the computation time for the procedure `compute_velocityfield`():

$$\begin{aligned} T_{Vel,1.2} = N_{x_1}N_{x_2}(2S_1S_2 \cdot [c_1 + c_2 + c_6 + (c_3 + c_6 + c_7 + c_8)] \\ +4c_1 + 6c_2 + 2c_3 + 2c_5 + 2c_6 + (c_3 + c_6 + c_7 + c_8)). \end{aligned}$$

Line 1.5 to line 1.7 in the main routine `extended_macro_solver`() uses 6 additions, 9 multiplications and 1 assignment. Furthermore the call of the function $F_1()$ or $G_1()$ needs a comparison operation. The assignments of F^+, F^- are not necessary and can be neglected. The convoluted **For** loops in line 1.3 and 1.4 repeat the operations in line 1.5-1.7 ($N_{x_1}N_{x_2}$) times. This yields a computation time for line 1.3-1.9:

$$T_{Loop,1.3-1.9} = N_{x_1}N_{x_2}(6c_1 + 9c_2 + 4c_3 + c_7 + (c_3 + c_6 + c_7 + c_8)).$$

The computation time of line 1.10 - 1.16 is equal to line 1.3-1.9. The **For** loop in line 1 repeats the computation for one time-step N_t times. This yields the entire computation time for the routine `extended_macro_solver`()

$$T_{run}^{macro} = \quad N_t \cdot (T_{Vel,1.2} + 2N_{x_1}N_{x_2}(6c_1 + 9c_2 + 5c_3 + 2c_7 + c_8).$$

This completes our proof. □

Remark 3.3.5. *A few remarks are in order.*

1. *The runtime performance is independent of the total number of objects. It just depends on the number of time and space steps, i.e., the complexity is $\mathcal{O}(N_{x_1}N_{x_2}S_1S_2N_t)$. Mollifier with non compact support are reduced to a finite number of grid points. For less computation times, it is recommendable to use mollifiers with small supports.*

2. *To ensure stability of the algorithm* ***`extended_macro_solver()`****, the CFL condition must be satisfied, i.e., $\frac{\Delta t}{\Delta x_d} \max_\rho \|(\frac{\partial}{\partial \rho}[\rho(\boldsymbol{v}^{dyn}(\rho) + \boldsymbol{v}^{stat}(\boldsymbol{x}))]\|_\infty \leq 1$ for $d = 1, 2$. In our case this is valid for a smoothed version of the Heaviside function. A similar expression can also be derived for the use of the (non-smooth) Heaviside function.*

3.3.2 Discontinuous Galerkin Methods

Discontinuous Galerkin methods (DG methods) play an important role in finding approximations of many physical applications based on hyperbolic partial differential equations. For example, popular applications are found in gas dynamics, compressible and incompressible flows, chemical transports, granular flows, and more. We refer to [9, 11, 17] for a short overview. These methods have some interesting benefits, e.g., they preserve the flexibility of finite elements in handling complicated geometries and they yield very accurate approximations. As already seen in Subection 3.3.1, finite volume methods use constant cell averages. In consideration of upwinding, this leads to artificial numerical diffusion which can influence the approximation quality. Indeed, this leads to consider other approximation tools like the following discontinuous Galerkin method.

The main goal is finding solutions of hyperbolic partial differential equations of the form

$$\partial_t \rho + \nabla \cdot (\boldsymbol{F}(\rho)) = 0. \tag{3.10}$$

The macroscopic model equations (3.3) and (3.5) can be written into (3.10) by the right choice of the flux $\boldsymbol{F}$. Thus, the flux $\boldsymbol{F}$ is a two-dimensional function and discontinuous in ρ. Note that the flux of the flow model and the extended flow model contains the discontinuous Heavyside function H. However, due of the discontinuous Galerkin method, the flux $\boldsymbol{F}$ is approximated by polynomials. This will be shown in the later steps. Hence, it is necessary to require continuous flux functions $\boldsymbol{F}$ to ensure numerical stability of the DG method. In particular, we specify the continuous fluxes of the macroscopic models in Remark 3.3.6.

Remark 3.3.6. *The presented discontinuous Galerkin method uses the following fluxes for the macroscopic models.*

- *The flux of the flow model* (3.3) *is*

$$\begin{aligned} \boldsymbol{F}(\rho) &= (f_\delta(\rho) v_1^{stat}, f_\delta(\rho) v_2^{stat})^T, \\ f_\delta(\rho) &= \min\{\rho, \frac{1}{\delta}(\rho_{max} - \rho)\} \quad \text{for } \delta > 0. \end{aligned}$$

  The smoothed flux f_δ is already introduced in Section 3.3.1 and in Chapter 1.

- *The extended flow model* (3.5) *can be approximate by the smooth flux*

$$\begin{aligned} \boldsymbol{F}(\rho) &= (\rho(v_1^{stat} + v_1^{dyn}), \rho(v_2^{stat} + v_2^{dyn}))^T, \\ v_d^{dyn}(\rho) &= \tilde{H}(\rho - \rho_{max}) I_d(\rho), \quad d = 1, 2, \end{aligned}$$

  where $\tilde{H}$ is a smoothed version of the Heavyside function, i.e.,

$$\tilde{H}(u) = \frac{1}{\pi} \arctan(\beta u) + \frac{1}{2}, \quad \beta > 0.$$

Space Integration

In this presentation of the DG method, some materials are drawn from these work [18, 56, 66, 70].
We consider a finite element discretization of the spatial domain $\Omega \simeq \Omega_h = \dot{\bigcup}_{k=1}^{K} D^k$, where Ω_h is a disjoint union of triangle elements D^k. Also, we assume that the position of each vertices of D^k can only coincide to other vertices of neighboring triangle elements. An example of such finite element discretization or triangulation is given in Figure 3.4. Note that h estimates the "size" of all triangle element D^k. In this thesis, h denotes the length of the largest triangle edge of all elements D^k.

Let $V = L^2(\Omega, \mathbb{R}^+)$ be the solution space of (3.10). Now let the approximate

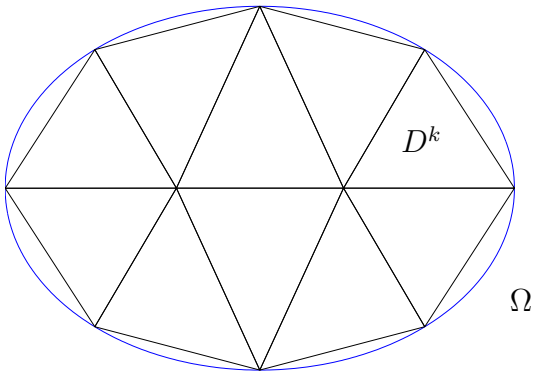


Figure 3.4: A finite element discretization (triangulation) of a domain Ω (ellipse).

space $V_h \subset V$ be defined by

$$V_h := \{v \in V : v|_{D^k} \in P^N, k = 1, ..., K\},$$

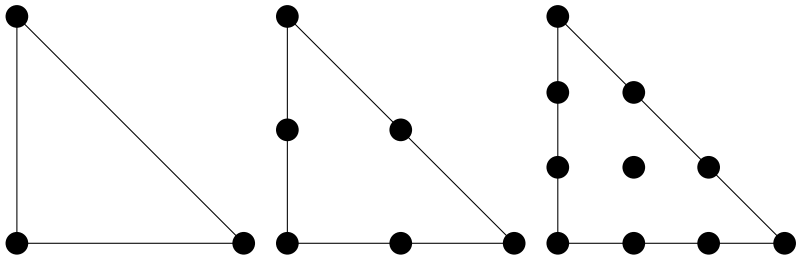
where P^N is the space of the polynomials of degree N. By definition the solutions v are discontinuous at the triangle interfaces. For the scheme we characterize all elements $v \in V_h$ by a nodal basis. In this presentation, a nodal basis is a special case of a polynomial basis. Note that a two dimensional polynomial has

$$N_p := \frac{(N+1)(N+2)}{2}$$

degrees of freedom for choosing the coefficients. All polynomials $v|_{D^k}$, restricted to a triangle shaped domain D^k, are constructible by nodal basis functions $\ell_i^k(\boldsymbol{x}) \in P^N$ with

$$\ell_i^k(\boldsymbol{x}_j^k) = \begin{cases} 1 & i = j, \\ 0 & i \neq j, \end{cases} \quad \text{for all } i, j = 1, ..., N_p,$$

where $\boldsymbol{x}_j^k \in D^k$ are nodal points on the finite element k. The polynomials $\ell_i^k(\boldsymbol{x})$ are called Lagragian basis functions. The nodal points $\boldsymbol{x}_i^k$ for $i = 1, ..., N_p$ are distributed on each triangle element D^k as respective shown in Figure 3.5. A detailed description of finite elements can be found in [12, 66] for an overview.



(a) $N = 1$ (b) $N = 2$ (c) $N = 3$

Figure 3.5: Nodal points of the basis for linear, quadratic, and cubic triangle elements D^k, see [12].

An approximation of the solution (3.10) is given by an element of V_h, i.e.,

$$\rho_h^k(\boldsymbol{x}, t) := \sum_{i=1}^{N_p} \rho_i^k(t)\ell_i^k(\boldsymbol{x}), \quad \boldsymbol{F}_h^k(\boldsymbol{x}, t) := \sum_{i=1}^{N_p} \boldsymbol{F}(\rho_i^k(t))\ell_i^k(\boldsymbol{x}), \quad \forall \boldsymbol{x} \in D^k. \quad (3.11)$$

The functions $\rho_i^k(t)$ are unknowns and characterizes the solution ρ_h^k at time t. We distinguish that the approximations ρ_h^k and the flux $\boldsymbol{F}_h^k$ fulfills (3.10) in an arbitrary way, i.e.,

$$\partial_t \rho_h^k(\boldsymbol{x}, t) + \nabla \cdot \boldsymbol{F}_h^k(\boldsymbol{x}, t) = \mathcal{R}_h^k(\boldsymbol{x}, t), \quad \forall \boldsymbol{x} \in D^k,$$

where $\mathcal{R}_h^k(\boldsymbol{x}, t)$ is the residual. Generally, the approximation ρ_h^k does not fulfill (3.10) exactly and the residual is not zero in all cases. Furthermore, we must decide in which sense the residual should vanish. Therefore, we choose a test function $\phi(\boldsymbol{x}) \in V_h$ that is representable as

$$\phi_h^k(\boldsymbol{x}) := \sum_{i=1}^{N_p} \phi_i^k \ell_i^k(\boldsymbol{x}), \quad \forall \boldsymbol{x} \in D^k.$$

We now require that the residual is orthogonal to all test functions in V_h, i.e.,

$$\int_{D^k} \mathcal{R}_h^k(\boldsymbol{x}, t)\phi_h^k(\boldsymbol{x}) d\boldsymbol{x} = 0.$$

This is true if and only if

$$\int_{D^k} \mathcal{R}_h^k(\boldsymbol{x}, t)\ell_j^k(\boldsymbol{x}) d\boldsymbol{x} = 0, \quad \forall j = 1, ..., N_p$$

holds. Thus, we obtain

$$\int_{D^k} (\partial_t \rho_h^k(\boldsymbol{x},t) + \nabla \cdot \boldsymbol{F}_h^k(\boldsymbol{x},t))\ell_j^k(\boldsymbol{x})d\boldsymbol{x} = 0. \tag{3.12}$$

Integrating (3.12) by parts yields

$$\begin{aligned} &\int_{D^k} \frac{\partial \rho_h^k(\boldsymbol{x},t)}{\partial t}\ell_j^k(\boldsymbol{x}) - \boldsymbol{F}_h^k(\boldsymbol{x},t)\cdot\nabla\ell_j^k(\boldsymbol{x})d\boldsymbol{x} \\ &= -\int_{\partial D^k} \boldsymbol{n}\cdot\boldsymbol{F}_h^k(\boldsymbol{x},t)\ell_j^k(\boldsymbol{x})d\boldsymbol{x} \quad \forall j = 1,...,N_p, \end{aligned} \tag{3.13}$$

where $\boldsymbol{n}$ represents the local outward pointing normal. The solution at the interfaces between triangle elements is multiply defined. At this moment, we have a lack of conditions on the local solution and the test functions. Therefore, we need here a correct combination of solutions to reduce the degree of freedoms. We select a numerical flux $\boldsymbol{F}^*$ for the fluxes at the triangle interfaces. An illustrated example is given in Figure 3.6.
Thus, equation (3.13) leads to the local statement.

$$\begin{aligned} &\int_{D^k} \frac{\partial \rho_h^k(\boldsymbol{x},t)}{\partial t}\ell_j^k(\boldsymbol{x}) - \boldsymbol{F}_h^k(\boldsymbol{x},t)\cdot\nabla\ell_j^k(\boldsymbol{x})d\boldsymbol{x} \\ &= -\int_{\partial D^k} \boldsymbol{n}\cdot\boldsymbol{F}^*\ell_j^k(\boldsymbol{x})d\boldsymbol{x} \quad \forall j = 1,...,N_p, \end{aligned} \tag{3.14}$$

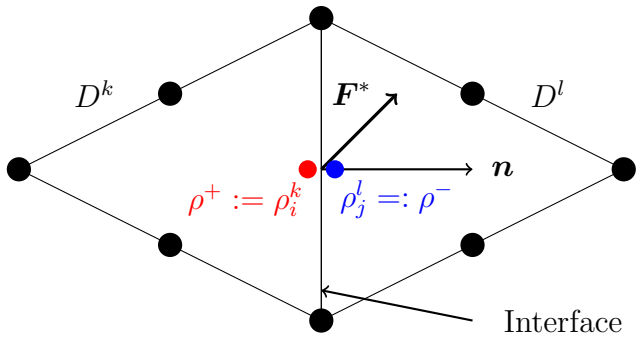


Figure 3.6: Interface of two neighboring triangles D^k and D^l. The position of the nodal points x_i^k (red) and x_j^l (blue) coincides, i.e., $x_i^k = x_j^l$. The interior and exterior densities ρ^+, ρ^- define the numerical flux $\boldsymbol{F}^*$ at the transition point $x_i^k = x_j^l$.

Especially in this work, we choose the local Lax-Friedrichs flux for the presented DG method:

$$\boldsymbol{F}^*(\rho^+,\rho^-) = \frac{\boldsymbol{F}(\rho^+) + \boldsymbol{F}(\rho^-)}{2} + \frac{C}{2}\boldsymbol{n}(\rho^+ - \rho^-),$$

where ρ^+, ρ^- are the interior and exterior solution value. Respectively, C is the local maximum of the directional flux

$$C = \max_{\rho\in[\rho^+,\rho^-]} \left| n_x \frac{\partial F_1}{\partial \rho} + n_y \frac{\partial F_2}{\partial \rho} \right|.$$

The goal is to achieve an ODE system to obtain the quantity $\rho_i^k(t)$. We plug now (3.11) into (3.13) and we get the following statement

$$\sum_{i=1}^{N_p} \left[\frac{\partial \rho_i^k(t)}{\partial t} \int_{D^k} \ell_i^k(\boldsymbol{x}) \ell_j^k(\boldsymbol{x}) d\boldsymbol{x} - \boldsymbol{F}(\rho_i^k(t)) \cdot \int_{D^k} \ell_i^k(\boldsymbol{x}) \nabla \ell_j^k(\boldsymbol{x}) d\boldsymbol{x} \right] =$$
$$- \int_{\partial D^k} \boldsymbol{n} \cdot \sum_{i=1}^{N_p} \boldsymbol{F}^* \ell_i^k(\boldsymbol{x}) \ell_j^k(\boldsymbol{x}) d\boldsymbol{x} = - \sum_{e=1}^{3} \int_{\text{interface } e} \boldsymbol{n}_e \cdot \sum_{i=1}^{N_p} \boldsymbol{F}^* \ell_i^k(\boldsymbol{x}) \ell_j^k(\boldsymbol{x}) d\boldsymbol{x}, \tag{3.15}$$

where $\boldsymbol{n}_e$ denotes the outward pointing normal of the interface e of the triangle D^k. The ODE system (3.15) can be written into a matrix notation, i.e.,

$$\mathcal{M}^k \frac{\partial \rho^k(t)}{\partial t} + \mathcal{S}_1^k F_1(\rho^k(t)) + \mathcal{S}_2^k F_2(\rho^k(t)) = - \sum_{e=1}^{3} \mathcal{M}^{k,e} (\boldsymbol{n}_e \cdot \boldsymbol{F}^*), \tag{3.16}$$

where ρ^k is a vector of dimension N_p containing the cell unknowns ρ_i^k. The local mass matrices $\mathcal{M}^k$, and the stiffness matrices $\mathcal{S}_1^k$, $\mathcal{S}_2^k$ are defined by

$$\mathcal{M}_{i,j}^k = \int_{D^k} \ell_i^k(\boldsymbol{x}) \ell_j^k(\boldsymbol{x}) d\boldsymbol{x},$$
$$\mathcal{S}_{d,i,j}^k = \int_{D^k} \ell_i^k(\boldsymbol{x}) \partial_{x_d} \ell_j^k(\boldsymbol{x}) d\boldsymbol{x}, \quad d = 1, 2, \forall i, j = 1, ..., N_p, \quad k = 1, ..., K,$$
$$\mathcal{M}_{i,j}^{k,e} = \int_{\text{interface } e} \ell_i^k(\boldsymbol{x}) \ell_j^k(\boldsymbol{x}) d\boldsymbol{x}, \quad e = 1, 2, 3.$$

Remark 3.3.7. *The coefficient matrices $\mathcal{M}_{i,j}^k, \mathcal{S}_{d,i,j}^k, \mathcal{M}_{i,j}^{k,e}$ for $d = 1, 2$ and $e = 1, 2, 3$ depend only on the choice of the basis functions and the corresponding triangulation. Therefore, it is useful to compute theses matrices once only for a complete simulation. This can be done by a preprocessing routine.*

Discontinuous and Shock Solutions - Filtering

As is known already, nonlinear equations lead to shocks or discontinuities in solutions. However, the polynomial approximation of solutions of the DG method is not able to prescribe discontinuities so far. If we apply the previous DG method to problems with shock solutions, the following problems will occur:

- The appearance of artificial and persistent oscillations around the point of discontinuity.
- The loss of pointwise convergence at the point of discontinuity.

This phenomenon is already known as the Gibbs phenomenon and its behavior is well understood [55].
Anyway, a high order polynomial basis on the elements gives an high order accuracy of the scheme for smooth solutions. However, the DG method handles discontinuities with persistent oscillations that distort the approximate solution or influence the stability properties. Therefore, we propose the following filter approach in stabilizing the computations and in reducing the oscillations.

The filter approach [15, 65] considers ways to recover some accuracy informations hidden in the oscillatory solutions. One possibility is filtering out high frequent redundant oscillations (high order polynomials) in the solutions. In the following, we consider the canonical basis

$$\psi_m(\boldsymbol{r}) = r_1^i r_2^j, , \quad (i,j) \geq 0; \quad i+j \leq N, \tag{3.17}$$

$$m := j + (N+1)i + 1 - \frac{i}{2}(i-1), \quad (i,j) \geq 0; \quad i+j \leq N. \tag{3.18}$$

which spans the space of N-dimensional polynomials in two variables $\boldsymbol{r} = (r_1, r_2)$. Additionally, the spatial variable $\boldsymbol{r}$ is restricted to a reference triangle I, i.e., $\boldsymbol{r} \in I := \{(r_1, r_2) : r_1, r_2 \geq -1, r_1 + r_2 \leq 0\}$. However, it is a complete polynomial basis and it can be orthonormalized through a Gram-Schmidt process. The resulting basis is denoted by $\tilde{\psi_m}(\boldsymbol{r})$.
The next step is to transform the basis function $\tilde{\psi_m}(\boldsymbol{r})$ back on a triangle element D^k. This is realizable by a linear mapping $\Psi : I \to D^k$. Thus, we obtain the basis function on D^k by $\tilde{\psi}_m^k(\boldsymbol{x}) := \tilde{\psi}(\Psi^{-1}(\boldsymbol{x}))$ with the property

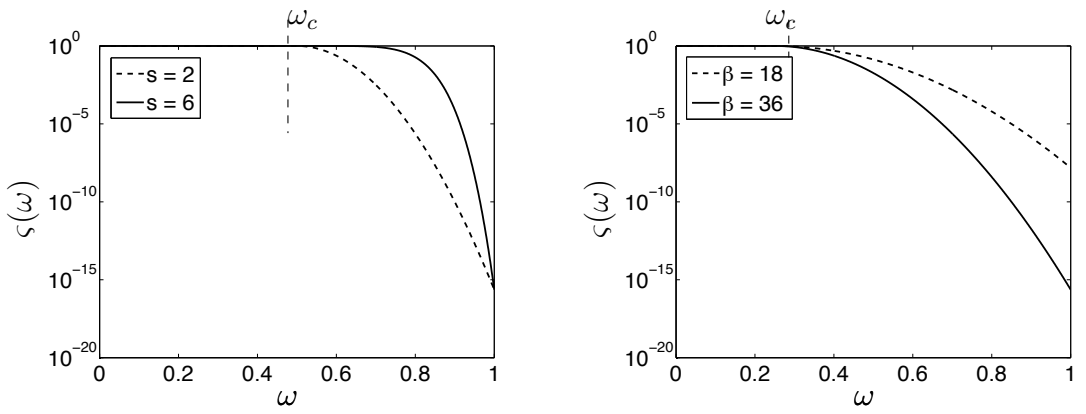
$$\int_{D^k} \tilde{\psi}_m(\boldsymbol{x}) \tilde{\psi}_n(\boldsymbol{x}) dx = \delta_{m,n}.$$

An approximate solution of an element D^k is given by

$$\rho_h^k(\boldsymbol{x}) = \sum_{i=1}^{N_p} \rho_i^k \ell_i^k(\boldsymbol{x}) = \sum_{m=1}^{N_p} \tilde{\rho}_m^k \tilde{\psi}_m^k. \tag{3.19}$$

The solution above is given in a multidimensional Lagrange polynomial basis ℓ_i^k. Now we transform $\rho_h^k(\boldsymbol{x})$ into the basis consisting of $\tilde{\psi}_m^k$. Note that the polynomial $\tilde{\psi}_m^k$ has the degree $\deg(\tilde{\psi}_m^k) = i + j$. The idea of filtering is to reduce the coefficient $\tilde{\rho}_m^k$ of high polynomial basis elements $\tilde{\psi}_m^k$. A popular choice is the exponential filter

$$\varsigma(\omega) = \exp(-\beta \omega^{2s}) \tag{3.20}$$



(a) $\beta = 36$, $\omega_c = 0.5$. (b) $s = 2$, $\omega_c = 0.25$.

Figure 3.7: Examples of how the filter function (3.21) varies from the three parameters; the order s, the cutoff $N_c = N\omega_c$, and the maximum damping parameter β.

to obtain the filtered expansion

$$\rho_h^{k,F}(\boldsymbol{x}) = \sum_{i,j\geq 0}^{i+j\leq N} \varsigma\left(\frac{i+j}{N}\right)\tilde{\rho}_m^k \tilde{\psi}_m^k,$$

where the filter is characterized by the the maximum damping parameter $\beta > 0$ and the order $s > 0$. It is reasonable to use other filter approaches, see [15, 65]. In this work, we use a filter of the form

$$\varsigma(\omega) = \begin{cases} 1, & 0 \leq \omega \leq \omega_c = \frac{N_c}{N} \\ \exp(-\beta((\omega-\omega_c)/(1-\omega_c)^{2s}), & \omega_c \leq \omega \leq 1. \end{cases} \tag{3.21}$$

The filter (3.21) is an extension of the exponential filter (3.20). N_c presents a cutoff, i.e., polynomials $\tilde{\rho}_m^k$ with degree $\deg(\tilde{\rho}_m^k) \leq N_c$ are left untouched. An example of the filter (3.21) with different parameters is shown in Figure 3.7.

Since filtering usage should be used both, as minimal as possible and as much as needed. This is necessary to stabilize the method, reduce oscillatory solutions, and reduce artificial viscosity.

Remark 3.3.8. *For instance, other strategies to avoid redundant oscillations and stabilize DG solutions are slope limiters [70], or subcell shock capturing strategies [86, 87].*

Convolution Integration

In particular, the dispersive term $\boldsymbol{I}(\rho)$ of (3.5) depends on the convolution of the density ρ and the gradient of the mollifier η, i.e.,

$$\nabla(\eta * \rho) = (\partial_{x_1}\eta * \rho, \partial_{x_2}\eta * \rho)^T.$$

Hence, it is necessary to include the convolution process into the discontinuous Galerkin Scheme. Without loss of generality, we consider the convolution of the approximate solution $\rho_h \in V_h$ and $\partial_1\eta$ in the nodal point $\boldsymbol{x}_i^k$ of a triangle k, i.e.,

$$\begin{aligned}(\partial_{x_1}\eta * \rho_h)(\boldsymbol{x}_i^k) &= \int_\Omega \eta(\boldsymbol{x}_i^k - \boldsymbol{\tau})\rho_h(\boldsymbol{\tau})d\boldsymbol{\tau} = \sum_{l=1}^{K}\int_{D^l} \eta(\boldsymbol{x}_i^k - \boldsymbol{\tau})\rho_h^l(\boldsymbol{\tau})d\boldsymbol{\tau} \\ &= \sum_{l=1}^{K}\int_{D^l} \eta(\boldsymbol{x}_i^k - \boldsymbol{\tau})\sum_{j=1}^{N_p}\rho_j^l\ell_j^l(\boldsymbol{\tau})d\boldsymbol{\tau} \\ &= \sum_{l=1}^{K}\sum_{j=1}^{N_p}\rho_j^l \underbrace{\int_{D^l} \eta(\boldsymbol{x}_i^k - \boldsymbol{\tau})\ell_j^l(\boldsymbol{\tau})d\boldsymbol{\tau}}_{:=c_{i,j}^{k,l}} = \sum_{l=1}^{K}\sum_{j=1}^{N_p}\rho_j^l c_{i,j}^{k,l}.\end{aligned}$$

The computation for the convolution of $\rho_h \in V_h$ and $\partial_2\eta$ works analogously.

Remark 3.3.9. *Note that the weights $c_{i,j}^{k,l}$ are time independent. Therefore, the $c_{i,j}^{k,l}$ can be computed once only before the simulation starts. However, the computation can results in high computational efforts for a large number of triangles K and polynomial degree N. Under certain circumstances, it is necessary to determine and store a number of $\mathcal{O}((N_pK)^2)$ weights to evaluate the convolution $(\partial_{x_1}\eta * \rho_h)$ for all nodal points.*

Time Integration

The DG approximation leads to a system of N_p ordinary differential equations over each element D^k. After inverting the local mass matrix $\mathcal{M}_k$, the system (3.16) can be transformed in the following matrix form:

$$\frac{d\rho^k(t)}{dt} = \mathcal{A}(\rho^k),$$

where $\rho^k(t)$ is a vector of dimension N_p containing the cell unknowns ρ_i^k. $\mathcal{A}(\rho^k)$ denotes the components of the right hand side of the ODE system (3.16) multiplied by the inverse mass matrix $\mathcal{M}_{i,j}^k$. The corresponding ODE system can be solved by explicit methods, e.g., forward Euler, explicit Runge, Runge-Kutta, and many more. For more details, we refer to [66, 70].

Example: Forward Euler Method A simple approach is to use the explicit Euler scheme to solve (3.16). As a result, the DG computation procedure is illustrated by the following steps:

1. Computation of $\tilde{\rho}^k$ is given as follows

$$\tilde{\rho}^k = \rho^k(t_n) + \Delta t \mathcal{A}(\rho^k(t_n)), \quad \forall k = 1, ..., K.$$

2. Reconstruction of the updated solution $\tilde{\rho}^k$ by applying

$$\rho^k(t_{n+1}) = \mathcal{F}(\tilde{\rho}^k), \quad \forall k = 1, ..., K,$$

where $\mathcal{F}$ denotes the filter process that is discussed above.

3.4 Optimization Approach

The singularizer has the task to sort and redirect the cargo on the conveyor belt. Afterwards, the cargo is transported to the next machine tools for additional production stages or quality controls. Therefore, it is useful to know how the cargo moves along the singularizer and how the cargo is redirected to the next machine tool. Often, machine tools have a limited capacity and require a certain input (inflow). Hence, it is necessary to control the material flow of the conveyed cargo by a demand f^*_{out}. Indeed, there is a high degree of freedom for controlling. Moreover, we are interested in finding the cargo position in front of the singularizer to fulfill the demand.
The presented task is modeled by a PDE restricted optimization problem. Note that there exist several approaches for different applications which are based on PDE-restricted optimization problems, e.g., [22, 63, 80, 99] for an overview.

$$\min_u \frac{1}{2} \int_0^T \left(\int_{\partial\Omega_{out}} \rho(\boldsymbol{v}^{stat} + \boldsymbol{v}^{dyn}) \cdot \boldsymbol{n} dx - f^*_{out}(t) \right)^2 dt \tag{3.22a}$$

subject to

$$\partial_t \rho(\boldsymbol{x}, t) + \nabla \cdot (\rho(\boldsymbol{x}, t)(\boldsymbol{v}^{stat}(\boldsymbol{x}) + \boldsymbol{v}^{dyn}(\boldsymbol{x}, t))) = 0, \tag{3.22b}$$

$$\rho(0, \boldsymbol{x}) = \begin{cases} u(\boldsymbol{x}) & \forall \boldsymbol{x} \in \Omega_{control} \\ 0 & \text{otherwise,} \end{cases} \tag{3.22c}$$

$$0 \leq u(\boldsymbol{x}) \leq \rho_{max}. \tag{3.22d}$$

The approach (3.22) is formulated as a PDE-restricted optimization problem based on the extended flow model (3.5a). The corresponding objective function (3.22a) computes a "distance" between the demand and several outflows of the extended flow model. That means if the outflow of the extended model $\rho(\boldsymbol{v}^{stat} + \boldsymbol{v}^{dyn})$ is close to f^*_{out}, the objective function becomes small. Additionally, (3.22b)

describes the PDE-constraint with initial data (3.22c). The control function $u : \Omega_{control} \to [0, \rho_{max}]$ characterizes the initial density (cargo position) at time 0 in $\Omega_{control}$. A sketch of the spatial domain and the outflow boundary is given in Figure 3.8.

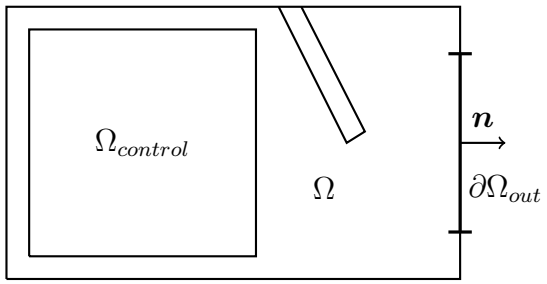


Figure 3.8: Optimal Control Problem

3.4.1 Black Box Optimization

The problem (3.22) searches a function u as the initial density for the extended flow model with respect to an objective function (3.22a). Indeed, there are many approaches to solve PDE-restricted optimization problems [26, 75, 100]. However, one of the simplest approaches is the black box method. Therefore, u is divided into N discrete values to reduce the complexity of the optimization problem, see Figure 3.9. The black box method is based on a nonlinear optimization method. Therefore, the PDE model is repeated until the optimization routine find an optimal solution u. Famous standard optimization approaches are, e.g., gradient descent methods, Nelder-Mead, etc. More optimization approaches can be found in, e.g., [93].

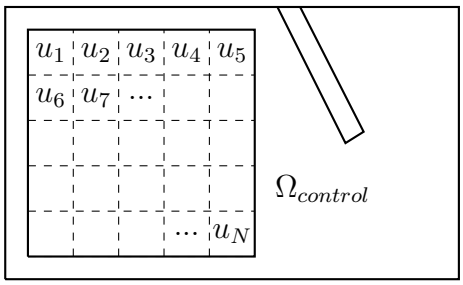


Figure 3.9: Discretization of the control u.

3.5 Numerical Results

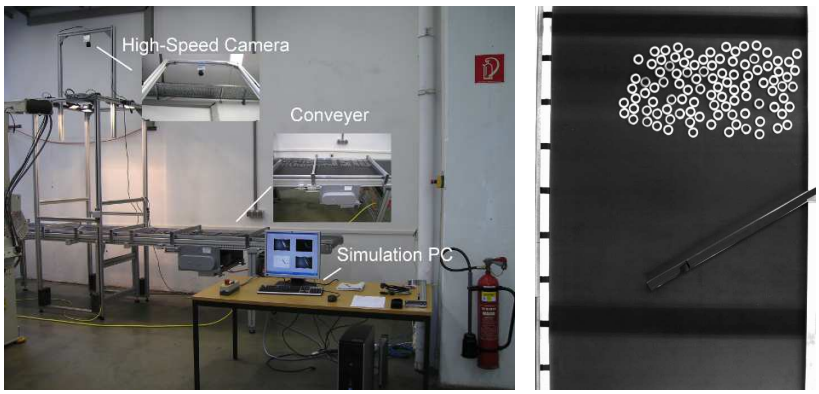
Finally, we present computational results of the flow and extended flow models. In particular, we cover the following aspects:

- In Subsection 3.5.1 we give a validation of the macroscopic models against real world experiments.
- In Subsection 3.5.2 we investigate the numerical efforts for the extended flow model. Therefore, we compare the discontinuous Galerkin method in Subsection 3.3.2 with the finite volume approach of Subsection 3.3.1.
- Additionally, in Subsection 3.5.3 we analyze the lane and pattern artifacts of the extended flow model and validate the results of the finite volume approach against the results of the discontinuous Galerkin method.
- In Subsection 3.5.4 we consider the extended flow model and its optimization issue. At first, we show the results of the conveyor belt outflow for different initial data. Afterwards, we present the results of the black box optimization approach.
- In Subsection 3.5.6 we present an example and a numerical test case of a conveyor accumulation buffer system.

All computations are performed on the same platform, namely a 3.0 GHz Dualcore computer with 8 GB RAM. All algorithms are implemented in MATLAB [83].

3.5.1 Real World Validation

Real World Settings The experiments describe the transport of cargo on a conveyor belt redirected by a singularizer. To collect real world data, the upper side of the conveyor belt is filmed by a high speed camera, see Figure 3.10. Image processing tools use the camera data to determine the positions and velocity of each object. Obviously, the quality of the real world data depends on several factors, i.e., ambient light intensity, camera refraction and robustness of the image processing algorithms. Hence, measuring errors cannot completely excluded. We consider a total of $N_n = 192$ cargo in the shape of metal cylinders with a radius of $R = 0.012m$ and a height of $l = 0.008m$. The maximal cargo density is equal to the hexagonal packings of two dimensional spheres with radius R. Therefore the maximal density is about $\rho_{max} = 2004$ parts per m^2. The velocity of the conveyor belt is $v_T = 0.395m/s$.



(a) Experimental setup (b) Exposure of the high-speed-camera

Figure 3.10: The setup consists of a conveyor belt, a high speed camera, and a computer for evaluating experimental data (left picture). The exposure of the high speed camera shows cargo lying on the conveyor belt [67] (right picture).

Macroscopic Model Settings The solution of the macroscopic model is computed by the finite volume scheme with dimensional splitting introduced in Section 3.3.1. The step sizes are set to $\Delta x_1 = 5 \cdot 10^{-3}, \Delta x_2 = 5 \cdot 10^{-3}, \Delta t = 1.25 \cdot 10^{-3}$ in the following numerical computations. The mollifier η occurring in the operator $\boldsymbol{I}(\rho)$ is set as follows

$$\eta(\boldsymbol{x}) = \frac{\sigma}{2\pi} \exp\Big(-\frac{1}{2}\sigma \|\boldsymbol{x}\|_2^2 \Big), \qquad \sigma = 10000.$$

The influence of the operator $\boldsymbol{I}(\rho)$ is determined by the factor $\epsilon = 2v_T$. The initial density $\rho_0(\boldsymbol{x})$ is given by the origin position of the cargo at time $t = 0$. Since the vector $\boldsymbol{x}_{i,0} \in \mathbb{R}^2$ denotes the position of a cargo i at time $t = 0$, the initial density $\rho_0(\boldsymbol{x})$ can be modeled by

$$\rho_0(\boldsymbol{x}) = \frac{\sigma_0}{2\pi\rho_{max}} \sum_{i=1}^{N_n} \exp\Big(-\frac{1}{2}\sigma_0 \|\boldsymbol{x} - \boldsymbol{x}_{i,0}\|_2^2 \Big), \qquad \sigma_0 = 2500. \tag{3.23}$$

In addition, the total mass of ρ_0 yields $\int \rho_0(\boldsymbol{x}) d\boldsymbol{x} = 192$.

Example 1: Flow Model vs. Extended Flow Model

We start with the setting that the singularizer angle is α is set to 60 degree. The results are shown in Figure 3.11. The left column in Figure 3.11 shows the measurements of a conveyor experiment. The middle and right column show the

numerical results of the flow model and the extended flow model. Each Plot visualize the cargo position for different times. In particular, the yellow cylinders in the left column visualize the cargo objects. The pictures in the middle and right column show the density functions as a gray-scaled image plot. Each color specifies a density value. Therefore, a dark color represent a higher density (black represent the maximal density) and vice versa. In all results, we observe that the cargo are transported with the velocity v_T. A formation of congestion is observable in all results. In due of the flow model (middle column), the cargo moves enormously slower along the singularizer than in the other results. The emerging diffusion in the both macroscopic model plots is an numerical artifact. The diffusion results from the step sizes in the numerical schemes. However, from a qualitative point of view, the results of the extended flow model are remarkably good and promising. The results of the flow model reproduce the formation of congestion very well; nevertheless, the entire qualitative behavior is quite unrealistic.

Mass balance and outflow behavior

Let us analyze the experiment quantitatively. We are interested in the amount of cargo that pass the singularizer. A time-dynamic mass function $U(t)$ counts all cargo which have not passed the singularizer. The aim is to compare the amount of passed objects for both models and the real data. For the real data, the time-dynamic mass function $U(t)$ is defined as

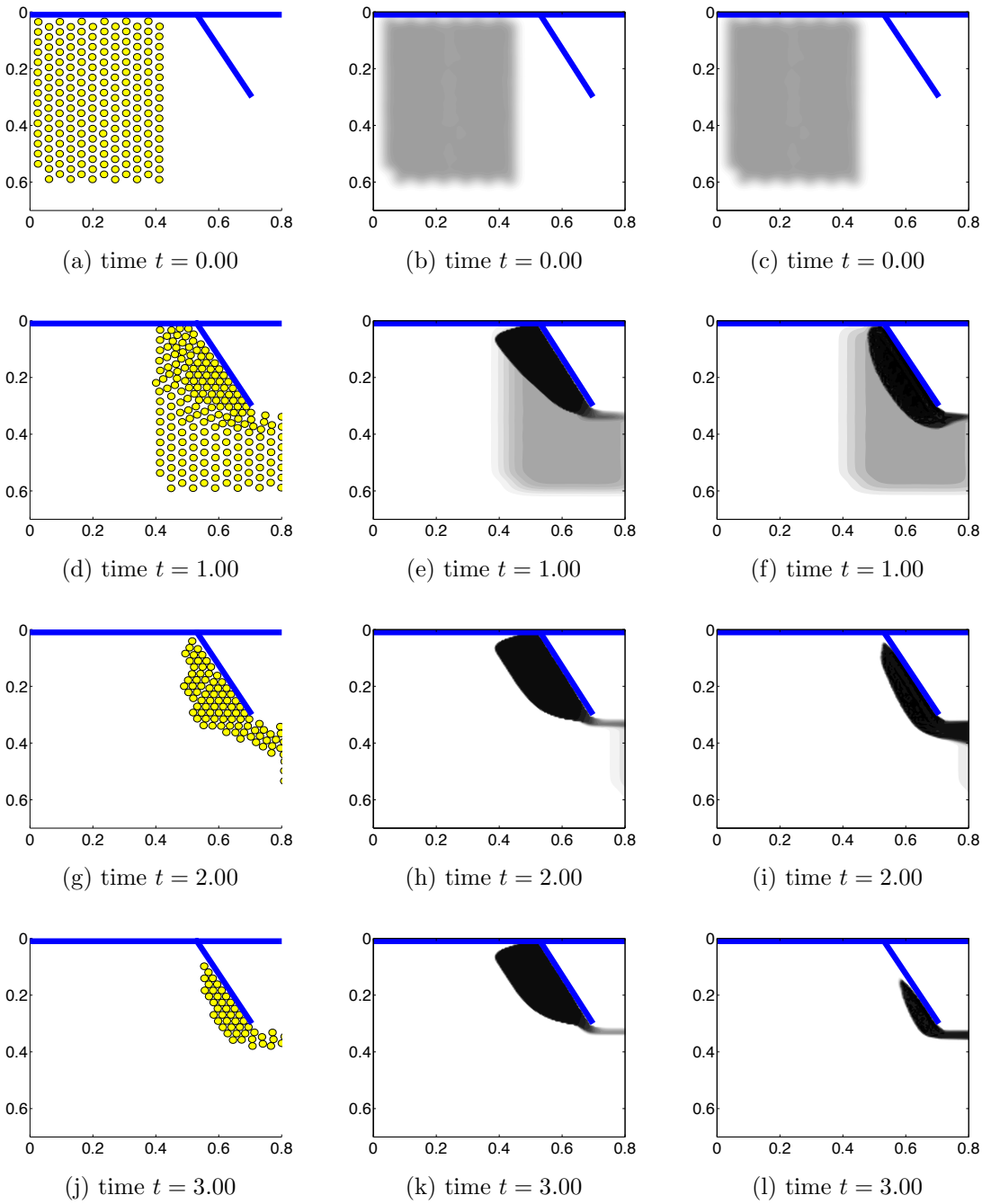
$$U(t) = \sum_{i=1}^{N_n} \chi_{\Omega_0}(\boldsymbol{x}_i(t)), \qquad \chi_{\Omega_0}(\boldsymbol{x}) = \begin{cases} 1 & \boldsymbol{x} \in \Omega_0 \\ 0 & \text{otherwise.} \end{cases}$$

where $\Omega_0 \subset \Omega$ is the left sided region in front of the obstacle, i.e., $\Omega_0 = \big\{(x_1, x_2) \in \mathbb{R}^2 \big| x_1 < 0.75\big\}$. The time-dependent mass function $U_\rho(t)$ for the macroscopic models is given by

$$U_\rho(t) = \int_{\Omega_0} \rho(\boldsymbol{x}, t) dx.$$

The evaluation of U and U_ρ is shown in Figure 3.12. At the beginning $t = 0$, the amount of cargo is 192. After a certain time, cargo pass the obstacle and the amount U, U_ρ decreases. We observe that macroscopic models fit quite well for $t < 2$. After time $t > 2$ a huge gap appears for the flow model, however, the amount U_ρ decrease slowly. There is a small gap between the extended flow model and the measurements from time $t = 2$ to time $t = 5$.

Remark 3.5.1. *The macroscopic model was used with ad hoc parameter choices and detailed parameter fits could significantly improve the results. However, in real applications, experimental data is not always available such that parameter fits cannot be performed.*



(a) time $t = 0.00$ (b) time $t = 0.00$ (c) time $t = 0.00$

(d) time $t = 1.00$ (e) time $t = 1.00$ (f) time $t = 1.00$

(g) time $t = 2.00$ (h) time $t = 2.00$ (i) time $t = 2.00$

(j) time $t = 3.00$ (k) time $t = 3.00$ (l) time $t = 3.00$

Figure 3.11: Real world data (left), flow model (middle), and extended flow model (right).

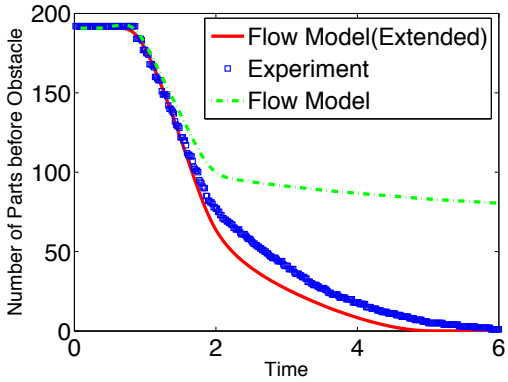
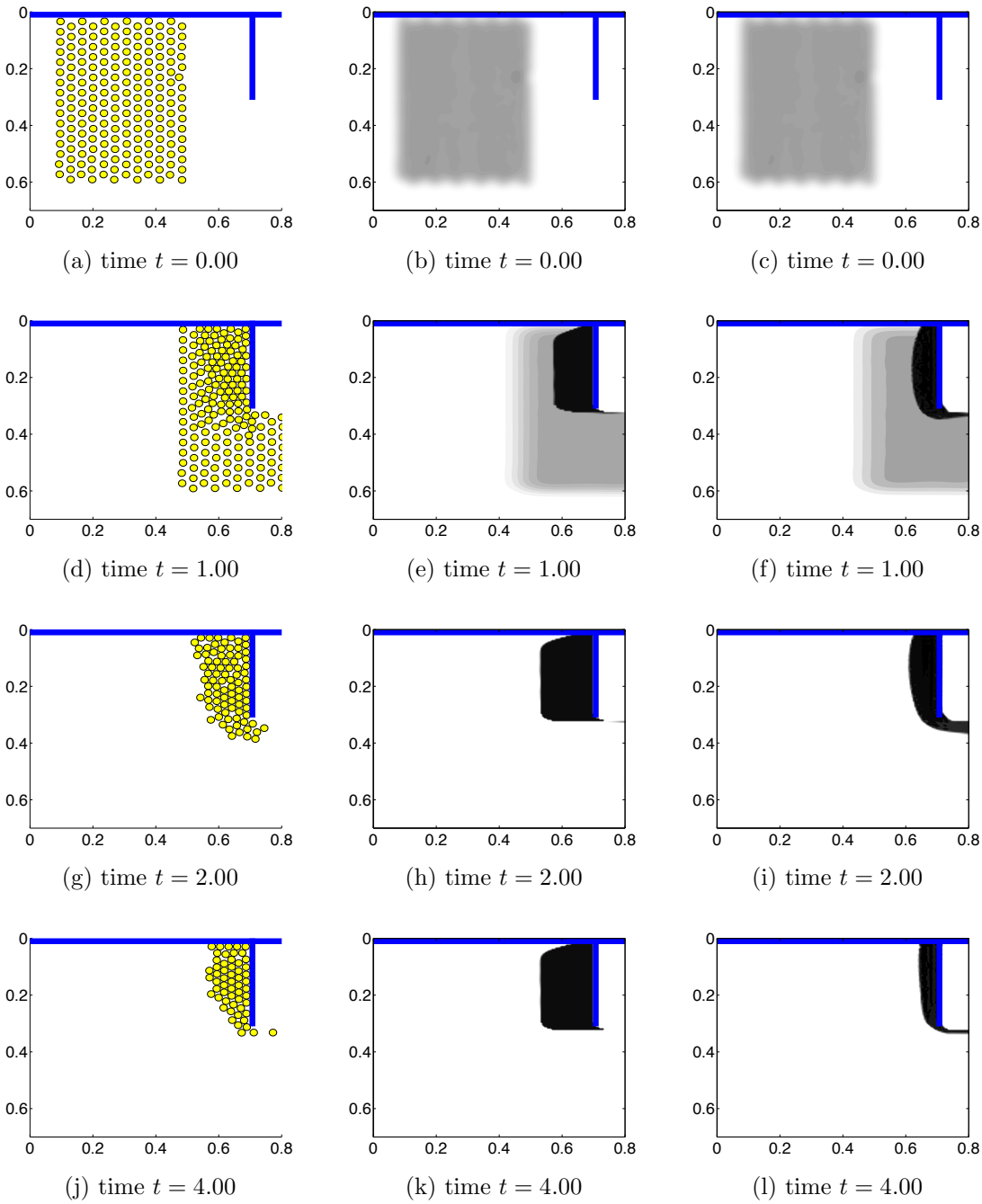


Figure 3.12: Comparison of the outflows over time. Each object and quantity is measured in the conveyor-region $x_1 < 0.75$. The dotted blue line represents the experimental data, while the dashed green and solid red lines correspond to the flow and extended flow model respectively.

Example 2: Flow Model vs. Extended Flow Model

Again, we use the same parameter as in Example 1 but with the difference that the singularizer angle is set to 90 degree now. The results are shown in Figure 3.13. The composition of the plots in Figure 3.13 is analogue to Figure 3.11. In all models, the cargo are transported with the conveyor belt velocity in direction of the singularizer. Due to the rectangular arrangement of the singularizer, we recognize more crowded regions and congestions. Note that the rounded shape of the congestion in Figure 3.13(f) is a result of the convolution $\nabla(\eta * \rho)$.
As in the previous comparison in Figure 3.12 the congestion at the singularizer dissolves slower in the flow model, than in the extended flow model and the experiment. This effect is emphasized in Figure 3.14. At time $t \approx 2$, an effect of tilting occurs in the real data, explaining the small plateau of the blue line in Figure 3.14. Note that this setting represents a very challenging experiment, since not all cargo can pass the singularizer. In the experimental setting, vibrations transmitted from the conveyor belt onto the cargo result in additional small contributions to the velocity of the objects. Due to the 90 degree angle of the singularizer, the overall outflow velocity of the cargo is lower than in Example 1, such that the effect of vibration is of higher influence in this setting. However, the additional contribution introduced by vibrations is not represented in the numerical models. Since these models tend to predict slightly too high throughput rates, but do not capture the additional velocity contribution, the gap in the outflow rate is reduced in this example.



(a) time $t = 0.00$ (b) time $t = 0.00$ (c) time $t = 0.00$

(d) time $t = 1.00$ (e) time $t = 1.00$ (f) time $t = 1.00$

(g) time $t = 2.00$ (h) time $t = 2.00$ (i) time $t = 2.00$

(j) time $t = 4.00$ (k) time $t = 4.00$ (l) time $t = 4.00$

Figure 3.13: Real world data (left), flow model (middle), and extended model (right).

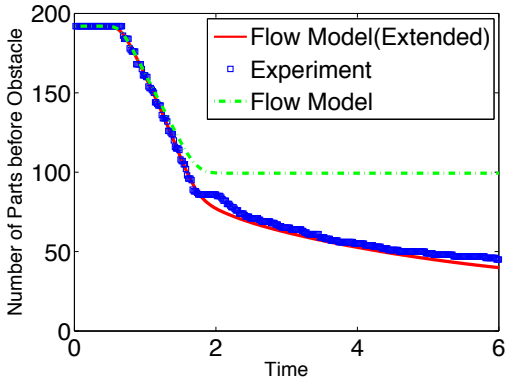


Figure 3.14: Comparison of the outflows over time. Each object and quantity is measured in the conveyor-region $x_1 < 0.75$. The dotted blue line represents the experimental data, while the dashed green and solid red lines correspond to the flow and extended flow model respectively.

3.5.2 Finite Volume vs. Discontinuous Galerkin

In this section, we compare the quality of the methods from Subsection 3.3.1 and 3.3.2. Additionally, we consider only solutions of the extended flow model. This model is based on a integral-differential equation, using a convolution term in the flux function. Similar models are already used for pedestrian flows [20]. However, certain lane or pattern artifacts are already observed for such pedestrian models. Also, lane artifacts appear in the extended flow model under certain assumptions. In this regard, it is not clearly understood why lane or pattern formation occurs. Thus, to investigate this phenomena (lane formation) in more details, we are motivated to reproduce these artifacts by a numerical scheme of higher order. A detailed discussion about the pattern or lane artifacts is found in the next section.
In the following, we present the numerical results of the extended flow model computed by the methods introduced in Section 3.3. The finite volume method and the discontinuous Galerkin method offer their benefits as well as drawbacks that are independently discussed in this section.

Finite Volume Settings The grid sizes of the finite volume approach with dimensional splitting are chosen as $\Delta x_1 = \Delta x_2 = 5 \cdot 10^{-3}$, $\Delta t = 1.25 \cdot 10^{-3}$ in the following computation.

Discontinuous Galerkin Settings The discontinuous Galerkin method uses a triangulation Ω_h with a maximal triangle edge length $h = 0.1$. The polynomial degree of each finite element is $N = 10$. The ODE system (3.16) is solved by

the explicit Euler method with a time step size $\Delta t = 10^{-3}$. Thereby the filter procedure is called in each computational step of the ODE solver. The filter settings are selected as $\beta = 36$, $s = 6$, and $N_c = 1$.

Macroscopic Model Settings As already mentioned in Remark 3.3.6, we choose a smooth modification of the dynamic velocity field $\boldsymbol{v}^{dyn}$, i.e.,

$$\begin{aligned} \boldsymbol{v}^{dyn} &= \tilde{H}(\rho - \rho_{max})\boldsymbol{I}(\rho), \\ \tilde{H}(u) &= \frac{1}{\pi}\arctan(25u) + \frac{1}{2}, \end{aligned}$$

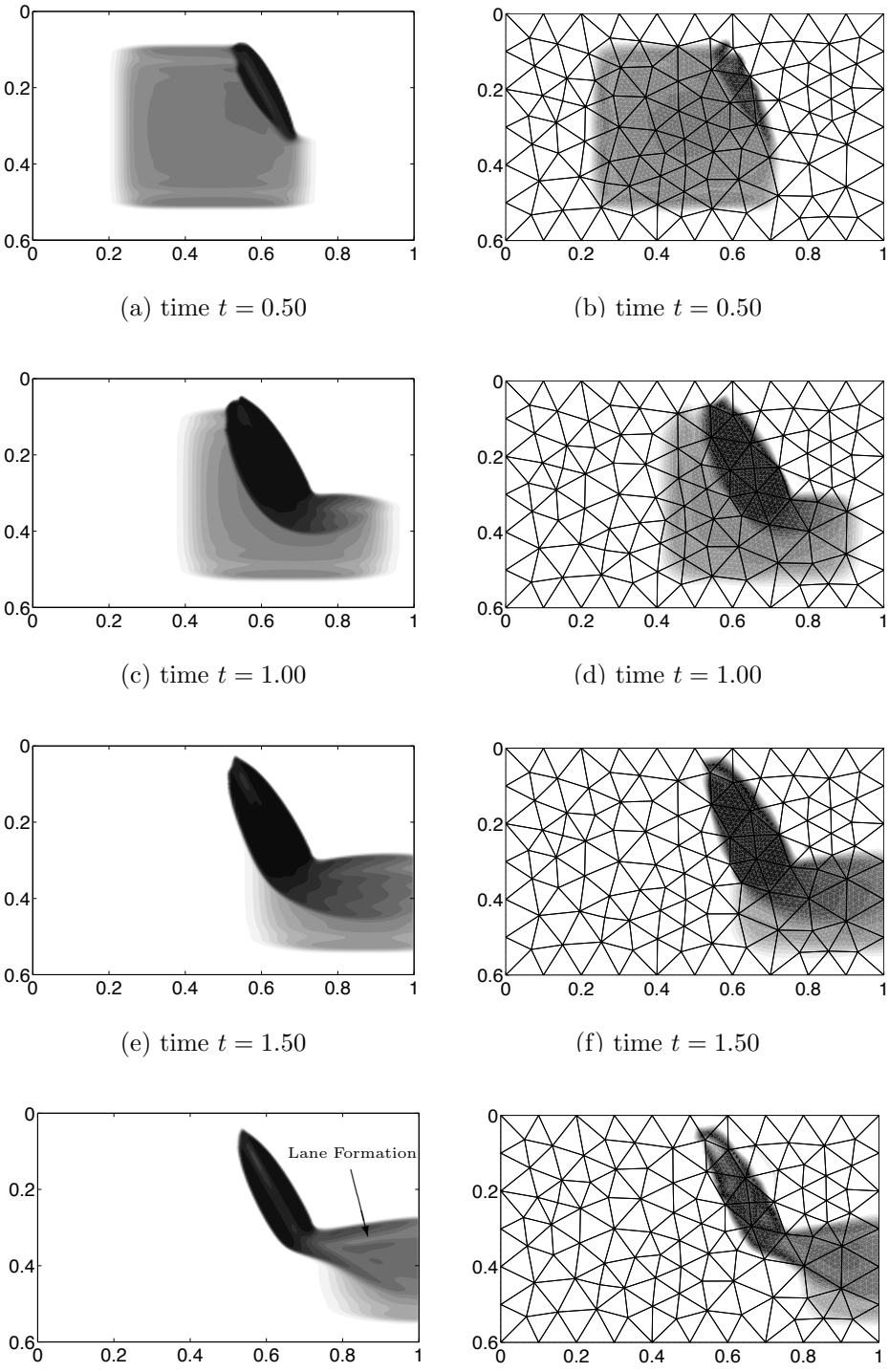
where $\tilde{H}$ is a smooth approximation of the Heaviside function. The mollifier η, occurring in the operator $\boldsymbol{I}(\rho)$, is defined as follows

$$\eta(\boldsymbol{x}) = \frac{\sigma}{2\pi}\exp\Big(-\frac{1}{2}\sigma\|\boldsymbol{x}\|_2^2\Big), \qquad \sigma = 2500.$$

In this example, the maximal density is set to $\rho_{max} = 1$. The strength of the term $\boldsymbol{I}(\rho)$ is selected as $\epsilon = 2v_T$. Furthermore, the time horizon is $T = 7$, and the singularizer angle α is set to 60 degree.

The results are shown in Figure 3.15. The left column shows the solution computed by the finite volume approach with splitting. The right column shows the results of the discontinuous Galerkin method. Each picture shows the density function as a gray-scaled image plot and each color specifies a density value. Thus, a dark color represent a higher density (black represent the maximal density) and vice versa. In all results, we observe that the cargo is transported by the conveyor belt velocity v_T. A formation of congestion is observable in all results.

In all plots, we recognize a weak dispersing of quantity, cf. Figure 3.15 (g), (h). This is caused by the term $\boldsymbol{v}^{dyn} = \tilde{H}(\rho - \rho_{max})\boldsymbol{I}(\rho)$. The smoothed modification $\tilde{H}(\rho - \rho_{max})$ is never zero for $\rho < \rho_{max}$. Consequently, the dispersing term $\boldsymbol{I}(\rho)$ is always activated and the quantity drifts apart all the time. This is also true, if the quantity has no connection to the singularizer, a dispersing effect is also recognizable, see Figure 3.15 (a), (b). Moreover, the term $\boldsymbol{I}(\rho)$ disperses the quantity with addition of artifacts (lane formation). Indeed, lane formations are observable, e.g. in Figure 3.15 (g). The solution of the discontinuous Galerkin method seems to be smooth and not accurate in contrast to the results of the finite volume method. This is mainly due to the fact that the DG method uses polynomials on triangle finite elements of degree $N = 10$. However, polynomials are inherently smooth, and it is impossible to approximate accurate shock solutions in due of the presented size of the finite elements. Indeed, the quality of the DG method can be improved by refining the triangle mesh grid. Compared to the DG method, the splitting method uses a 20 times higher discretization.



(a) time $t = 0.50$ (b) time $t = 0.50$

(c) time $t = 1.00$ (d) time $t = 1.00$

(e) time $t = 1.50$ (f) time $t = 1.50$

(g) time $t = 2.00$ (h) time $t = 2.00$

Figure 3.15: Results of the finite volume method with splitting - $\Delta x_1 = \Delta x_2 = 5 \cdot 10^{-3}$ (left), and results of the discontinuous Galerkin method - $h \approx 0.1$, $N = 10$ (right).

The question rises, what mesh grid sizes and what polynomial degrees are necessary to ensure good approximations in due of the discontinuous Galerkin method? In the following, the previous example is computed again by the DG method with different triangulations and polynomial degrees. We test our problem on 3 different mesh grid sizes $h = 0.1$, $h = 0.06$ and $h = 0.04$. The results are shown in Figure 3.16. For all grid sizes and polynomial degrees, the qualitative behavior of the solution is approximated quite well. A finer grid or a higher polynomial degree generates more precise solutions, i.e., quantity shocks and congested formations are drawn in an accurate way.
However, a rough triagulation or a low polynomial degree causes bad approximations, cf. Figure 3.16 (e). Compared to the other results, the congestion formation in Figure 3.16 (e) looks quite degenerated.

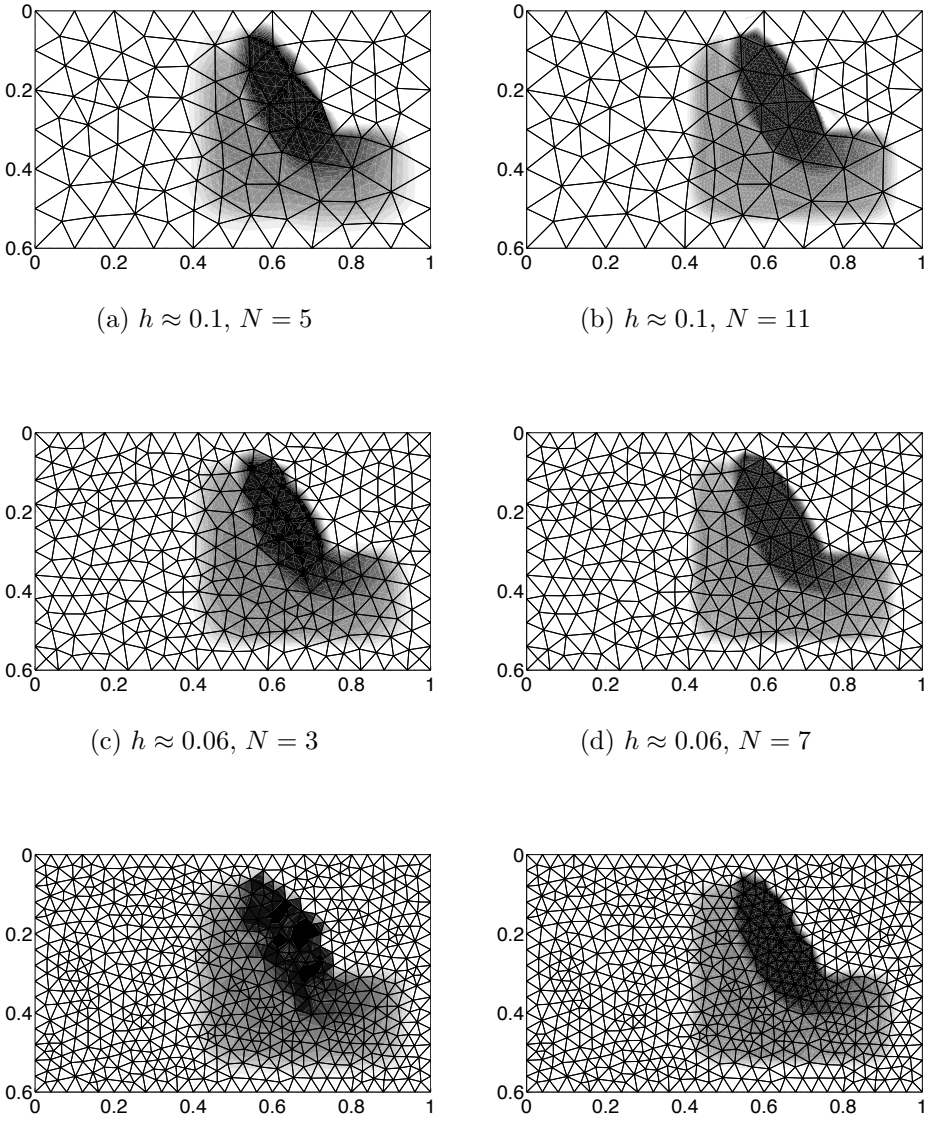
The computation times of the DG method with respect to the mesh-sizes and polynomial degrees are shown in Table 3.1 and Table 3.2. Furthermore, the computation times are distinguished into preprocessing time, cf. Table 3.2, and simulation time, cf. Table 3.1. Preprocessing contains the calculation of the coefficients of the convolution, see Remark 3.3.9. The simulation time contains the computation of the ODE system (3.16) by the explicit Euler method.

N	$h = 0.1$	$h = 0.06$	$h = 0.04$
1	7.14s	12.30s	13.42s
3	9.30s	18.63s	51.70s
5	17.31s	46.29s	-
7	30.10s	111.94s	-
9	49.58s	-	-
11	88.70s	-	-

Table 3.1: Computation times of the discontinuous Galerkin method (simulation process) with different grid sizes h and polynomial degrees N. The time is measured in seconds.

N	$h = 0.1$	$h = 0.06$	$h = 0.04$
1	0.06s	1.12s	1.73s
3	0.48s	3.88s	79.99s
5	2.01s	60.82s	-
7	5.44s	420.08s	-
9	47.11s	-	-
11	183.86s	-	-

Table 3.2: Computation times in seconds for the convolution preprocessing in due of the grid size h and polynomial degree N.



(a) $h \approx 0.1$, $N = 5$

(b) $h \approx 0.1$, $N = 11$

(c) $h \approx 0.06$, $N = 3$

(d) $h \approx 0.06$, $N = 7$

(e) $h \approx 0.04$, $N = 1$

(f) $h \approx 0.04$, $N = 3$

Figure 3.16: Results of the discontinuous Galerkin method with different triangulations ($h = 0.1, 0.06, 0.04$) and polynomials degrees N. All plots show the solution at time $t = 1$.

The computing time required for the calculation of the finite volume approach is about 788.21s. Consequently, the DG method is quite faster than the finite volume approach for all presented settings. However, the computing times and the memory requirements of the DG preprocessing increase enormously since the computation of the convolution in one nodal point requires at most $N_p \cdot K$ coefficients. Furthermore, there are $N_p \cdot K$ nodal points and the convolution is evaluated twice in each dimension. Thus, it is necessary to calculate and store about $2 \cdot (N_p \cdot K)^2$ coefficients. As a consequence, the computer was not able to run the preprocessing routine successfully for small h and a large N, for example, $N = 11$ and $h = 0.06$, see Table 3.2.
Let us summarize: The discontinuous Galerkin method is able to approximate accurately the extended flow equations on complex geometric domains. However, the presented example consists only a rectangle shaped domain and it is not necessary to use methods for complex geometries, cf. regular grids. As already seen, the DG method needs a very time and memory consuming preprocessing in due of the convolution. Hence, it is very expensive to apply small step sizes h for computation of accurate approximations and evaluating the corresponding convergence behavior.

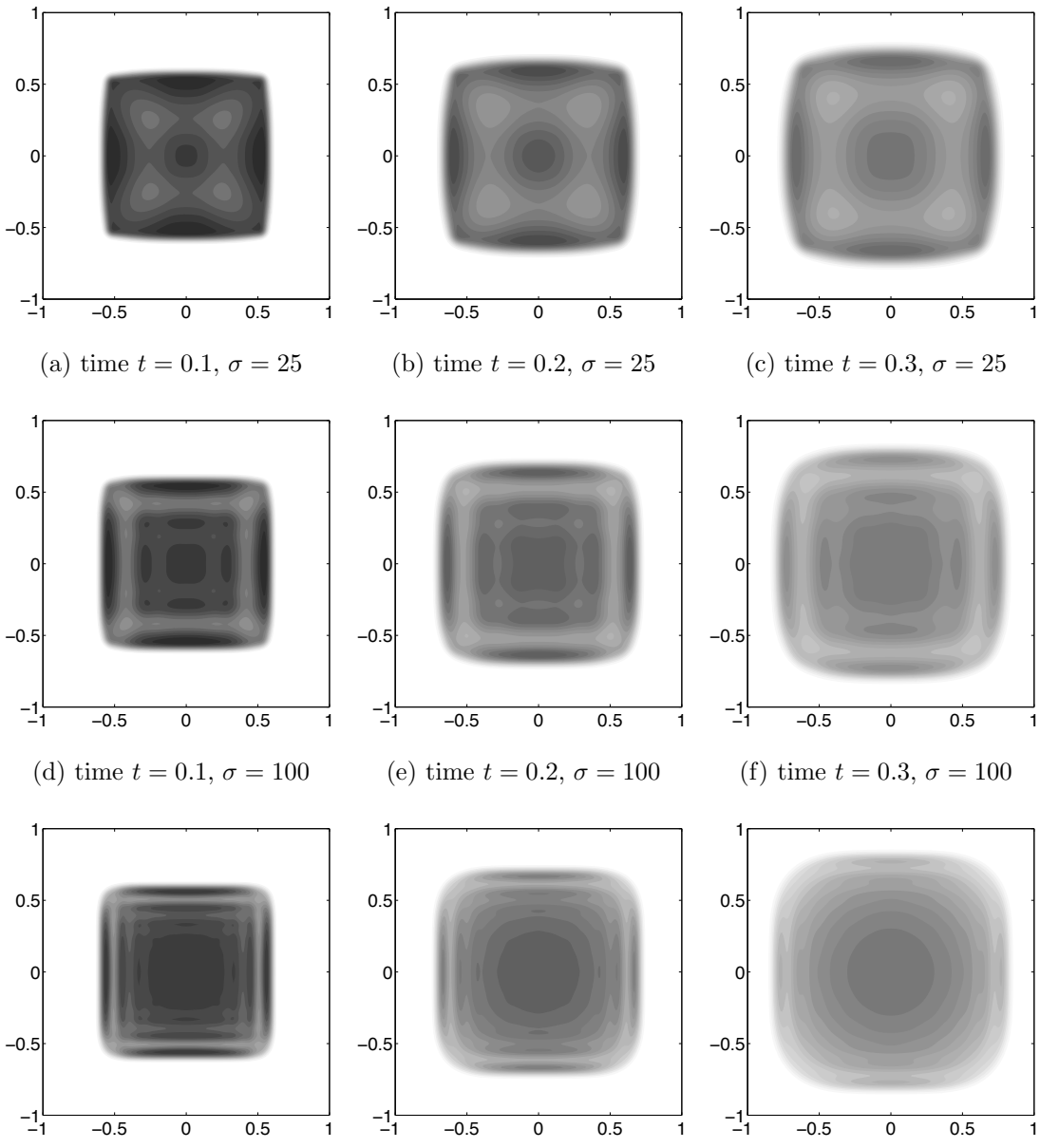
3.5.3 Lane and Pattern Formation

The extended flow model presented above has a discontinuous component, namely the Heaviside function H, see (3.5c). This decision part is included in the dispersive term $\boldsymbol{v}^{dyn}(\rho) = H(\rho - \rho_{max})\boldsymbol{I}(\rho)$. If $\rho > \rho_{max}$, the term $\boldsymbol{I}(\rho)$ will be active.
In the literature, models for pedestrian flow use a similar model [20], but do not limit the influence of the dispersive term to a maximum density. Therefore, these models do not contain the Heaviside function and set $\boldsymbol{v}^{dyn}(\rho) = \boldsymbol{I}(\rho)$. Additionally, we neglect the static velocity field $\boldsymbol{v}^{stat}$ of equation (3.5). This reduces to the following equation.

$$\begin{aligned} &\partial_t \rho + \nabla \cdot (\rho(\boldsymbol{I}(\rho)) = 0, \\ &\boldsymbol{I}(\rho) = -\frac{\nabla(\eta * \rho)}{\sqrt{1 + \|\nabla(\eta * \rho)\|_2^2}}. \end{aligned} \tag{3.24}$$

In [20], lane formation was observed for the pedestrian model with smooth dispersive term, whereas this effect seems to be much less present in the above presented non-smooth material flow model. Note that the extended flow model in Subsection 3.5.2 uses a smoothed switching function H. In particular, we observe already lane artifacts in Figure 3.15 (g).

We reproduce the appearance of the lane formation in the extended flow model and validate the results by both numerical methods (finite volume approach with

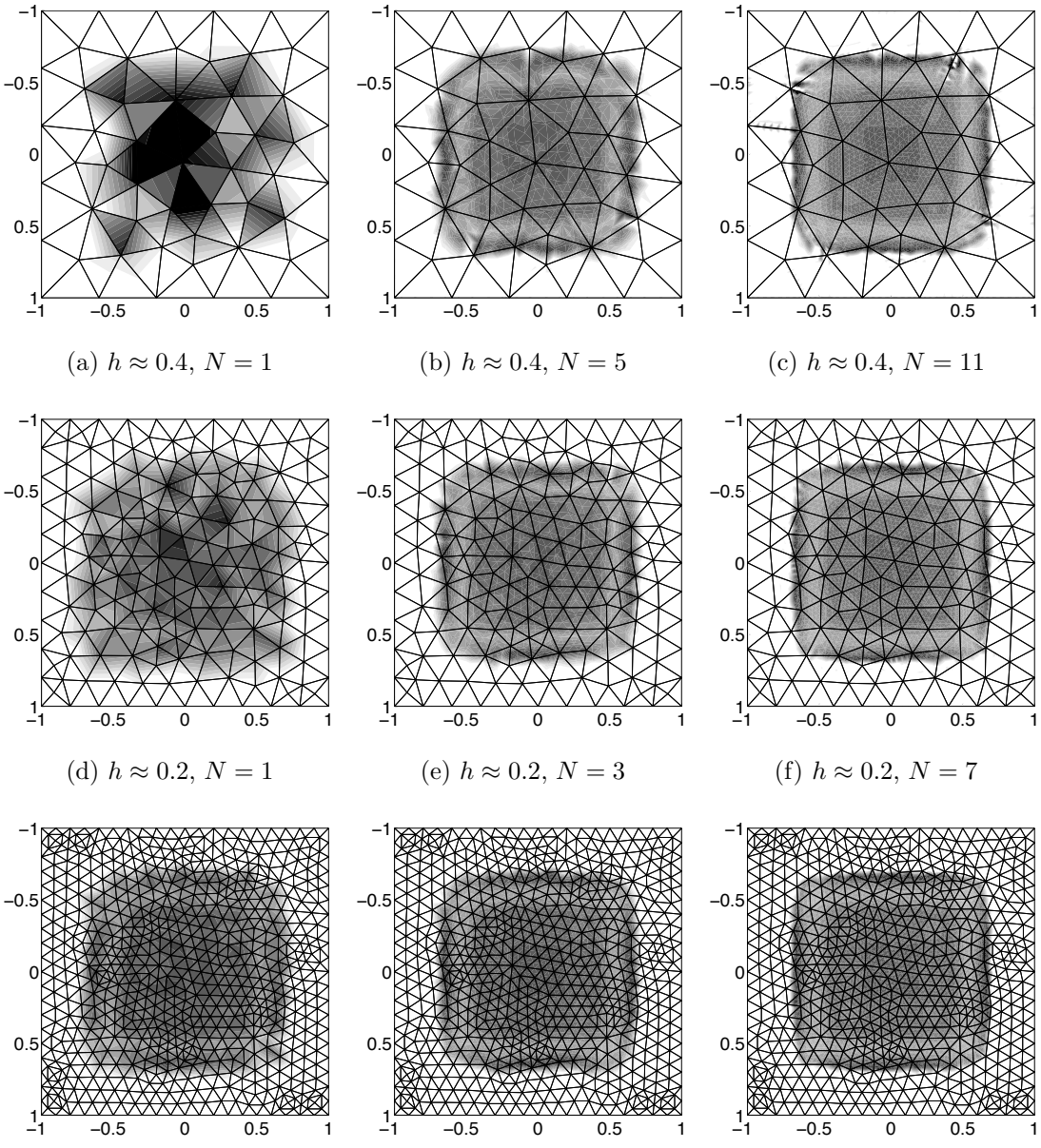


(a) time $t = 0.1$, $\sigma = 25$ (b) time $t = 0.2$, $\sigma = 25$ (c) time $t = 0.3$, $\sigma = 25$

(d) time $t = 0.1$, $\sigma = 100$ (e) time $t = 0.2$, $\sigma = 100$ (f) time $t = 0.3$, $\sigma = 100$

(g) time $t = 0.1$, $\sigma = 400$ (h) time $t = 0.2$, $\sigma = 400$ (i) time $t = 0.3$, $\sigma = 400$

Figure 3.17: Numerical solution of the simplified model (3.24) computed by the finite volume approach with dimensional splitting. Visualized for time $t = 0.1, 0.2, 0.3$ and smoothing function parameter $\sigma = 25, 100, 400$.



(a) $h \approx 0.4$, $N = 1$ (b) $h \approx 0.4$, $N = 5$ (c) $h \approx 0.4$, $N = 11$

(d) $h \approx 0.2$, $N = 1$ (e) $h \approx 0.2$, $N = 3$ (f) $h \approx 0.2$, $N = 7$

(g) $h \approx 0.1$, $N = 1$ (h) $h \approx 0.1$, $N = 2$ (i) $h \approx 0.1$, $N = 4$

Figure 3.18: Comparison of the results of the discontinuous Galerkin Method with different mesh-sizes h and polynomial degrees N. The plots show the solution at time $t = 0.2$ for a smoothness parameter $\sigma = 100$.

dimensional splitting and DG method). However, we solve the simplified equation (3.24) on the spatial domain $\Omega = [-1, 1]^2$. The initial density $\rho(\boldsymbol{x}, 0)$ is set to 1 for $\boldsymbol{x} \in [-\frac{1}{2}, \frac{1}{2}]^2$, otherwise $\rho(\boldsymbol{x}, 0) = 0$. Additionally, we compute the simplified model (3.24) for three different mollifiers, i.e., $\sigma = 25, 100, 400$. The step sizes of the finite volume approach are $\Delta x_1 = \Delta x_2 = 0.01$ and $\Delta t = 0.005$.

The results of the finite volume approach are shown in Figure 3.17. In all plots, we observe that the quantity spreads out in all directions. The top row corresponds to the setting with smoothing function parameter $\sigma = 25$. We recognize a squared shaped pattern in all time series. In the middle and lower row of plots, the smoothing function parameter $\sigma = 100, 400$ is used. Here, we observe a lane formation with a circular shape. A further increase of the mollifier parameter σ yields thiner lanes in the solution. However, we recognize the disappearance of the lanes in Figure 3.17 (h),(i). This is caused by the artificial numerical diffusion of the scheme which smears out the thin lanes in the solution.
Figure 3.18 shows the results of the discontinuous Galerkin Method for different triangulations and polynomial degrees; however, the results are plotted for the time $t = 2$ and $\sigma = 100$. All plots (exceptional (a) and (d)) lead to the same result and they are similar to the plot of Figure 3.17 (e). Indeed, a low triangulation and a low polynomial degree causes poor results, cf. Figure 3.18 (a) and (d). To get the most solution accuracy, the usage of filters for the DG computations is neglected. Therefore, some high frequent oscillations can appear, cf. Figure 3.17 (c).

3.5.4 Simulation

Comparison of the Outflow

We compare the behavior of the material outflow with respect to different choices of initial data. Therefore, we introduce three configurations of initial density values, i.e., we consider a higher, middle, and lower bulk of material as an initial density distribution. Each bulk is distributed to a density $\rho(x, 0) = 0.6$. Outside of a bulk, the initial density is 0. The shape and the size of each bulk is given in Figure 3.19. The material transport is simulated by the extended flow model for different singularizer angles $\alpha = 45, 60, 90$. The corresponding numerical method is the finite volume method with dimensional splitting. The following step sizes are used, i.e., $\Delta x_1 = \Delta x_2 = 10^{-2}$, and $\Delta t = 2.5 \cdot 10^{-3}$. The conveyor belt velocity is selected to $v_T = 0.395 \frac{m}{s}$. The smoothing parameter of the mollifier η is set to $\sigma = 2500$.

The outflow of the different bulks and different angles is shown in Figure 3.20. We observe, that the quantity of the lower bulk is transported by velocity v_T without any interaction of the singularizer with angle $\alpha = 45, 60$ degree. Hence,

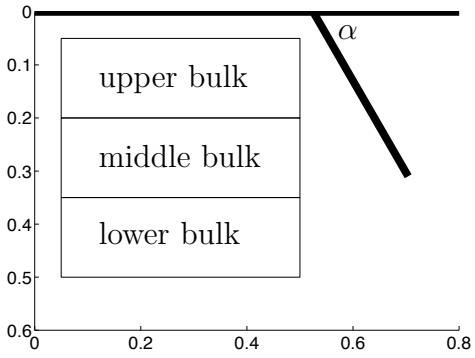
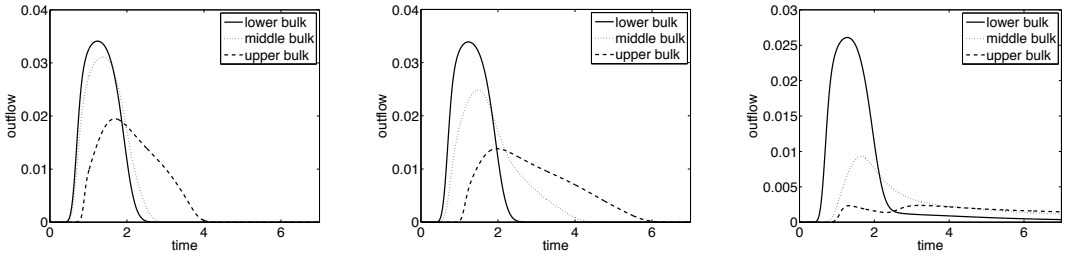


Figure 3.19: The initial density $\rho(x, 0)$ is divided into three bulks (lower, middle, and upper bulk).

the entire lower bulk flows directly out the domain without any additional delay. In all other cases, the bulks interact with the singularizer. Therefore, a significant delay of the mass transport is observable. In due of the case $\alpha = 90$ degree, the crowded quantity moves much slower than in the cases with $\alpha = 45, 60$. This results a very thin outflow rate for the lower and middle bulk, see Figure 3.20 (c).



(a) $\alpha = 45°$ (b) $\alpha = 60°$ (c) $\alpha = 90°$

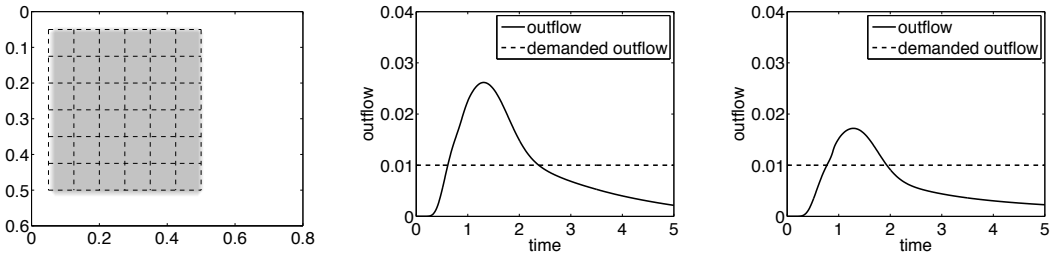
Figure 3.20: Cargo outflow for different singularizer angles (degrees). Each plot shows the outflow for different initial density values (lower, middle, and upper bulk).

3.5.5 Optimization

Optimal Outflow

The following numerical experiment leads to the optimal cargo position at starting time $t = 0$ such that the conveyor outflow fulfills the certain demand $f^*_{out} = 0.01$. Moreover, we apply the optimization model (3.22) in due of the black box

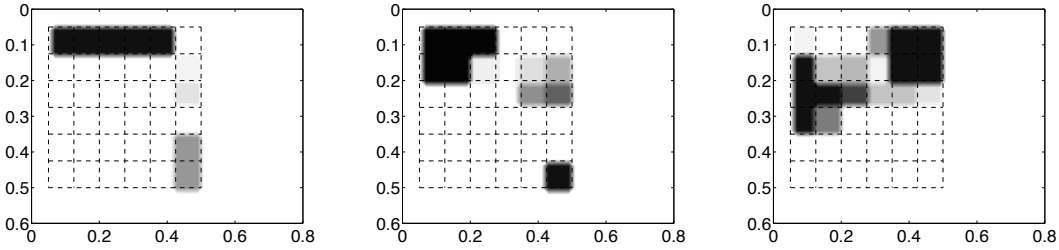
method in Subsection 3.4.1. The MATLAB routine `fmincon()` is used to solve approximately the optimization problem. The forward simulation is computed by the finite volume method with dimensional splitting. Additionally, we select a discretization with step sizes $\Delta x_1 = \Delta x_2 = 0.02$ and $\Delta t = 0.005$. Three scenarios are tested with different singularizer angles, i.e., $\alpha = 45, 60, 90$ (in degree). To reduce the complexity of optimization problem (3.22), we discretize the initial distribution of the cargo $u(\boldsymbol{x})$ into a squared 6×6 matrix which is prescribed by a vector $(u_1, ..., u_N)$ with $N = 36$. The values of the first iteration (start iteration) are set to $u_i = 0.5$. The corresponding results for $u_i = 0.5$ in due of the singularizer angle $\alpha = 60, 90$ are shown in Figure 3.21. Indeed, the cargo outflow of this scenario does not fulfill the demanded outflow f^*_{out}.



(a) Initial Value $u_i = 0.5$. (b) Outlflow ($\alpha = 60°$) (c) Outlflow ($\alpha = 90°$)

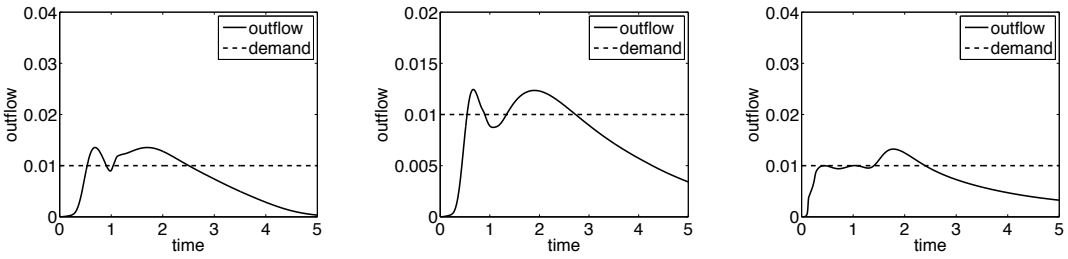
Figure 3.21: Outflow with cargo initial position $u = 0.5$.

The MATLAB optimization routine terminates for all scenarios and finds a local optimal solution for all test cases with angle $\alpha = 45, 60, 90$. The results of the optimal solution u are given in Figure 3.22. The corresponding outflow is plotted in Figure 3.23. The MATLAB routine `fmincon()` uses the active set algorithm in this computational example. The routines terminates in about 2 hours.
Indeed, the optimal outflow approximates the demand remarkably well.



(a) $\alpha = 45°$ (b) $\alpha = 60°$ (c) $\alpha = 90°$

Figure 3.22: Optimal cargo initial position. A dark color represents a higher density.



(a) $\alpha = 45°$ (b) $\alpha = 60°$ (c) $\alpha = 90°$

Figure 3.23: Outflow with optimal cargo initial position.

3.5.6 Conveyor Accumulation Buffer Systems

Often, production units do not work fluently or it is not possible to process the entire material flow. Therefore, unprocessed materials can be stacked in buffer systems to keep a steady production flow. In the following, we introduce a conveyor accumulation buffer system that is installed in front of a production unit. Concepts of the following accumulation buffer system are already offered by the Paxona AG *.
We reproduce the accumulation buffer system as a test case for the extended flow model.
The buffer system consists mainly of a primary and a secondary conveyor belt. As usual, the primary conveyor belt transport materials to a production unit. The secondary conveyor belt is installed parallel to the primary belt, and moves in the corresponding reverse direction. A layout of the accumulation buffer system is shown in Figure 3.24. The function of the secondary belt is an extension of the primary conveyor belt. As a consequence, additional cargo can be stored in the system to maintain a fluently production process.

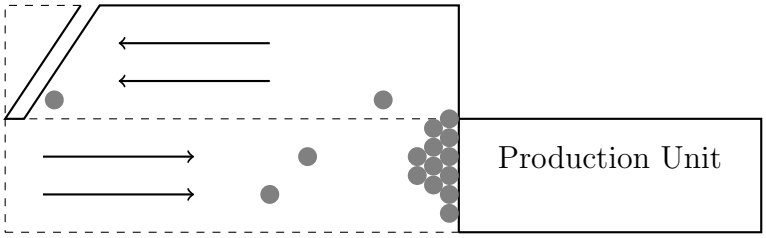
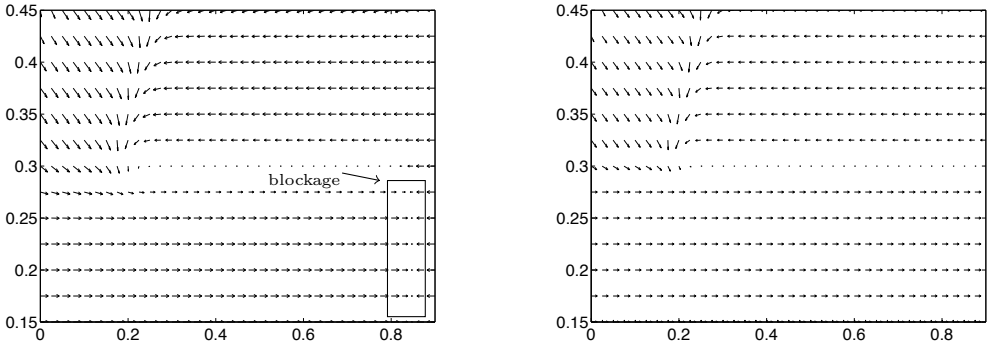


Figure 3.24: Conveyor accumulation buffer system

Next, we consider a production unit with an accumulation buffer system. The primary conveyor belt transports cargo to the production unit. Also, we assume

*www.paxona.de



(a) Static velocity field (blockage) (b) Static velocity field (free flow)

Figure 3.25: Static velocity fields for the extended flow model. The left velocity field models the conveyor accumulation buffer system with blocked production unit (blockage). The right velocity field simulates the same buffer system with working production unit (free flow).

that the production unit has a failure. Thus, the incoming cargo cannot be processed anymore and the cargo accumulates in front of the production unit. The congestion of cargo becomes larger until some cargo objects are pushed onto the secondary belt. Subsequently, cargo on the secondary belt is conveyed backwards to an angled obstacle (singlarizer) that redirects the cargo to the primary conveyor belt again. By that way, the system allows to stack additional cargo objects in a production line.

We reproduce the above introduced buffer system by the extended flow model. Furthermore, we simulate an increasing and decreasing of materials in the buffer. The production unit stops at time $0 \leq t \leq 2.5$ and continues its process at time $t > 2.5$. However, we extend the static velocity field of Subsection 3.2.1 by an additional conveyor belt with reverse direction. The failure and working of the production unit is realized by two separated static velocity fields, cf. Figure 3.25. The left static velocity field simulates the buffer system with a machine failure, i.e., the material cannot flow out of the right domain via primary conveyor belt. This velocity field is used for all times $t < 2.5$. However, the blockage is removed at the right static velocity field, cf. Figure 3.25 (b). As a consequence, the material can flow out of the conveyor system. After $t > 2.5$, the static velocity field $\boldsymbol{v}^{stat}$ switches from the blocked field (cf. Figure 3.25 (a)) to the free flow field (cf. Figure 3.25 (b)).

The velocities of both conveyor belts are set to $v_T = 1$. The maximal density is given by $\rho_{max} = 1$. The material inflow is described by the density 0.5 on the left

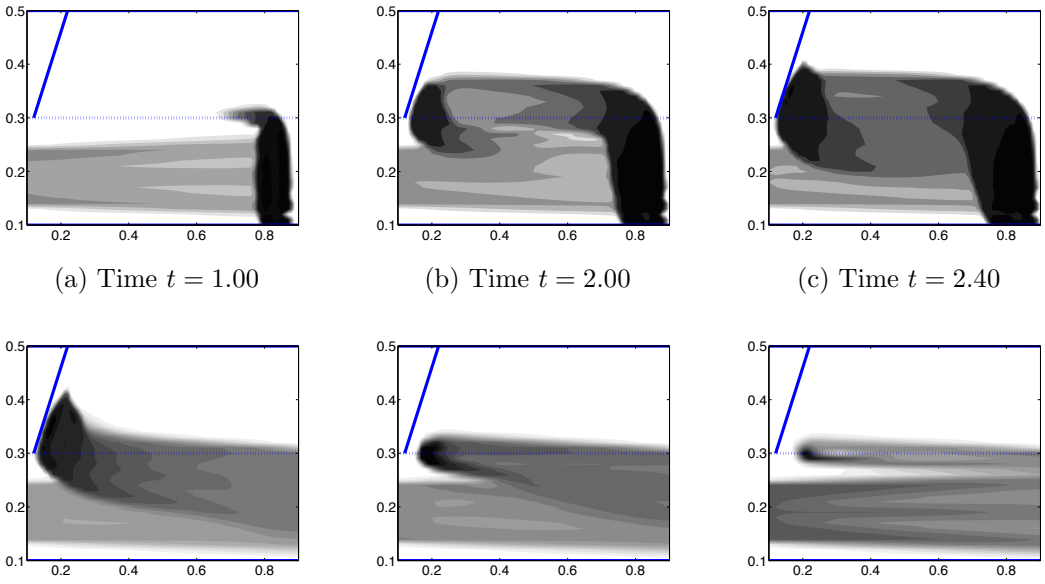
boundary and on the primary belt (lower belt).
We choose a smooth modification of the dynamic velocity field $\boldsymbol{v}^{dyn}$, i.e.,

$$\boldsymbol{v}^{dyn} = \tilde{H}(\rho - \rho_{max})\boldsymbol{I}(\rho),$$
$$\tilde{H}(u) = \frac{1}{\pi}\arctan(25u) + \frac{1}{2},$$

where $\tilde{H}$ is a smooth approximation of the Heaviside function. The mollifier η is defined by

$$\eta(\boldsymbol{x}) = \frac{\sigma}{2\pi}\exp\Big(-\frac{1}{2}\sigma\|\boldsymbol{x}\|_2^2\Big),$$

where $\sigma = 2500$. The simulation is computed by the finite volume approach with dimensional splitting. The step sizes are selected as $\Delta x_1 = \Delta x_2 = 0.01$ and $\Delta t = 0.001$.



(a) Time $t = 1.00$ (b) Time $t = 2.00$ (c) Time $t = 2.40$

(d) Time $t = 3.00$ (e) Time $t = 4.00$ (f) Time $t = 5.00$

Figure 3.26: Accumulation system on a conveyor belt.

The results are plotted in Figure 3.26. At time $t = 1$, the material is transported by the primary conveyor belt from left side to the right side. Also, we observe a congestion of material on the right side of the domain. For time $t = 2$ and $t = 2.4$, the size of congested material increases and a part of the material is pushed onto the secondary conveyor belt (upper belt). Thereby material on the secondary belt is transported to the singularizer, which redirect the material back

to the primary belt. After time $t > 2.5$, we observe that the congested material thins out and flows out of the domain. Note that the smoothed static velocity field is quite zero at the boundary layer (blue dotted line), i.e., for $x_2 = 0.3$. Consequently, the material flow in that region is quite low.

Conclusion

In many applications, simulation tools based on mathematical models are helpful to plan, organize and control manufacturing processes. In this work, we provided different models for production systems that are characterized by conservation laws (PDEs) with discontinuous flux functions. The first model describes an entire production flow on networks with finite buffers and deterministic machine break-downs. A scalar one-dimensional conservation law that is similar to the model of Armbruster, Göttlich, and Herty in [4] was extended to a novel network model. Therefore, the solution of the PDE was analyzed in detail by the wave front tracking approach. According to the propagation of shock waves, valid coupling conditions at the intersection could be found. Also, the wave front tracking method enables to establish the novel numerical scheme, namely the discontinuous flux Godunov (DFG) that is based on the finite volume method. The main advantage of this method is that no regularization of the flux are used and fast traveling shock waves are computed in an accurate way.
A further important application in manufacturing products is decision making with aid of optimization problems, for instance, reducing the buffer sizes, finding the optimal time interval of a maintenance, and more. To solve such optimization problems, we applied two different discrete approaches by using the novel DFG method. A reformulation of non linear parts in linear constraints and binary variable restrictions enables a mixed integer programming formulation (MIP). Additionally, we determined a discrete adjoint equation system of the PDE model on a single edge. We analyzed and compared the results and also the structure of the MIP and adjoint approach. Therefore, we could find a connection between both optimization approaches. Indeed, the MIP model requires an enormous computation time for large network problems. Due to accelerate the MIP computation, we investigated two approaches for a MIP presolving technique. The first presolving approach is based directly on the work of Dittel et al. in [33]. The second presolving approach is an extension of the first technique with an additional routine that is based on the discretized PDE network model.

The second application within this work is the simulation of material flow on conveyor belts. Therefore, we investigated two continuous flow models. The first model, so-called flow model, is a two-dimensional extension of the one-dimensional discontinuous conservation law of the presented network model. The second model (extended flow model) is similar to the pedestrian model in the work of Colombo et al. in [21]. Such models use a non local term including convolution

that is integrated in a flux function. Furthermore, the results of both flow models was validated against a real world experiment. The extended flow model reveals good results for practical applications. Moreover, the computational costs for simulations depend only on the grid size and is independent of the number of objects on the conveyor.
The extended flow model was tested by two different numerical methods, namely a finite volume method and a discontinuous Galerkin method. Solution artifacts (lane formation) were detected in the model of [21] and also in the extended flow model. We verified the appearance of such artifacts by the two different numerical solution approaches.
In summary, this work contains many fields of mathematics, i.e., in our case ordinary and partial differential equation systems, numerical methods and computations, and discrete optimization issues, that are connected to describe different production processes.
This research has thrown up some questions in need of further investigation.

- The presented network model is unable to reproduce random effects, for instance, random machine break-downs. How can stochastic processes be included into the model?
- An important limitation lies in the fact that only one kind of products is modeled. Is it possible to extend the presented model to a multiple commodity model?
- Due to the conveyor belt problem, how can further optimization issues be included into the model?
- How can the presented models be combined with existing production models?

Generally speaking, a further investigating of discontinuous conservation laws and its optimization issues in relation to production models is a worthwhile task with great potential.

Bibliography

[1] Adimurthi, J. Jaffre, and G. D. V. Gowda. Godunov-type methods for conservation laws with a flux function discontinuous in space. *SIAM J. Numer. Anal.*, 42(1):179–208, 2004.

[2] E. D. Anderson and K. D. Anderson. Presolving in linear programming. *Mathematical Programming*, 71(2):221–245, 1995.

[3] D. Armbruster, P. Degond, and C. Ringhofer. A model for the dynamics of large queuing networks and supply chains. *SIAM J. Appl. Math.*, 66(3):896–920, 2006.

[4] D. Armbruster, S. Göttlich, and M. Herty. A scalar conservation law with discontinuous flux for supply chains with finite buffers. *SIAM J. Appl. Math.*, 71(4):1070–1087, 2011.

[5] A. Aw, A. Klar, T. Materne, and M. Rascle. Derivation of continuum traffic flow models from microscopic follow-the-leader models. *SIAM J. Appl. Math.*, 63(1):259–278, 2002.

[6] A. Aw and M. Rascle. Resurrection of "second order" models of traffic flow. *SIAM J. Appl. Math.*, 60(3):916–938, 2000.

[7] F. Bachmann and J. Vovelle. Existence and uniqueness of entropy solution of scalar conservation laws with a flux function involving discontinuous coefficients. *Commun. Partial Differ. Equations*, 31(1-3):371–395, 2006.

[8] M. K. Banda, M. Herty, and A. Klar. Coupling conditions for gas networks governed by the isothermal Euler equations. *Netw. Heterog. Media*, 1(2):295–314, 2006.

[9] F. Bassi and S. Rebay. High-order accurate discontinuous finite element solution of the 2d euler equations. *J. Comput. Phys.*, 138(2):251–285, 1997.

[10] A. Beck, T. Bolemann, H. Frank, F. Hindenlang, M. Staudenmaier, G. Gassner, and C.D. Munz. *Discontinuous Galerkin for High Performance Computational Fluid Dynamics*. Springer-Verlag, 2013.

[11] B. J. Block, M. Lukácová-Medvid'ová, P. Virnau, and L. Yelash. Accelerated GPU simulation of compressible flow by the discontinuous evolution galerkin method. *Eur. Phys. J. Special Topics*, 210:119–132, 2012.

[12] D. Braess. *Finite Elemente: Theorie, schnelle Löser und Anwendungen in der Elastizitätstheorie*. Springer-Verlag, 2007.

[13] A. L. Brearley, G. Mitra, and H. P. Williams. Analysis of mathematical programming problems prior to applying the simplex algorithm. *Math. Program.*, 8:54–83, 1975.

[14] A. Bressan. *Hyperbolic systems of conservation laws. The one-dimensional Cauchy problem.*, volume 20. Oxford: Oxford University Press, 2000.

[15] C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang. *Spectral Methods in Fluid Dynamics.* Springer Series in Computational Physics. Springer-Verlag, 1988.

[16] J. Carillo. Conservation laws with discontinuous flux functions and boundary condition. In *Nonlinear Evolution Equations and Related Topics*, volume 3, pages 283–301. Birkhäuser Basel, 2003.

[17] B. Cockburn, G. Karniadakis, and C.W. Shu. *Discontinuous Galerkin Methods. Theory, Computation and Applications. Lecture Notes in Computational Science and Engineering*. Springer-Verlag, 2000.

[18] B. Cockburn and C. W. Shu. The runge-kutta discontinuous galerkin method for conservative laws v: Multidimentional systems. *J. Comput. Phys.*, 141:199–224, 1998.

[19] G. M. Coclite, M. Garavello, and B. Piccoli. Traffic flow on a road network. *SIAM J. Math. Anal.*, 36(6):1862–1886, 2005.

[20] R. M. Colombo, M. Garavello, and M. Lecureux-Mercier. A class of nonlocal models for pedestrian traffic. *Math. Models Methods Appl. Sci.*, 22:1150023, 2012.

[21] R. M. Colombo, M. Garavello, and M. Lecureux-Mercier. Nonlocal crowd dynamics models for several populations. *Acta Math. Sci. Ser. B Engl. Ed.*, 32:177–196, 2012.

[22] R. M. Colombo, G. Guerra, M. Herty, and V. Schleper. Optimal control in networks of pipes and canals. *SIAM J. Control Optim.*, 48(3):2032–2050, 2009.

[23] H. Crowder, E. Johnson, and M.W. Padberg. Solving large-scale zero-one linear programming problems. *Oper. Res.*, 31:803–834, 1983.

[24] P. A. Cundall and O. D. L. Strack. A discrete numerical model for granular assemblies. *Géotechnique*, 29:47–65, 1979.

[25] C. D'Apice, G. Bretti, R. Manzo, and B. Piccoli. A continuum-discrete model for supply chains dynamics. *Netw. Heterog. Media*, 2(4):661–694, 2007.

[26] C. D'Apice, S. Göttlich, M. Herty, and B. Piccoli. *Modeling, Simulation, and Optimization of Supply Chains: A Continuous Approach*. SIAM, Society for Industrial and Applied Mathematics, Philadelphia, 2010.

[27] C. D'Apice, R. Manzo, and B. Piccoli. Packet flow on telecommunication networks. *SIAM J. Math. Anal.*, 38(3):717–740, 2006.

[28] P. Degond, S. Göttlich, M. Herty, and A. Klar. A network model for supply chains with multiple policies. *Multiscale Model. Simul.*, 6(3):820–837, 2007.

[29] J. P. Dias and M. Figueira. On the Riemann problem for some discontinuous systems of conservation laws describing phase transitions. *Commun. Pure Appl. Anal.*, 3(1):53–58, 2004.

[30] J. P. Dias and M. Figueira. On the approximation of the solutions of the Riemann problem for a discontinuous conservation law. *Bull. Braz. Math. Soc. (N.S.)*, 36(1):115–125, 2005.

[31] J. P. Dias and M. Figueira. On the viscous Cauchy problem and the existence of shock profiles for a p-system with a discontinuous stress function. *Quart. Appl. Math.*, 63(2):335–341, 2005.

[32] J. P. Dias, M. Figueira, and J. F. Rodrigues. Solutions to a scalar discontinuous conservation law in a limit case of phase transitions. *J. Math. Fluid Mech.*, 7(2):153–163, 2005.

[33] A. Dittel, A. Fügenschuh, S. Göttlich, and M. Herty. MIP presolve techniques for a PDE-based supply chain model. *Optim. Methods Softw.*, 24(3):427–445, 2009.

[34] Q. Du, J. R. Kamm, R. B. Lehoucq, and M. L. Parks. A new approach for a nonlocal, nonlinear conservation law. *SIAM J. Appl. Math.*, 72(1):464–487, 2012.

[35] K. Ehrhardt and M. Steinbach. Nonlinear gas optimization in gas networks. *Modeling, Simulation and Optimization of Complex Processes (eds. H. G. Bock, E. Kostina, H. X. Pu, R. Rannacher), Springer Verlag, Berlin*, 2005.

[36] N. Foster and R. Fedkiw. Practical animation of liquids. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '01, pages 23–30, New York, NY, USA, 2001. ACM.

[37] A. Fügenschuh, S. Göttlich, M. Herty, A. Klar, and A. Martin. A discrete optimization approach to large scale supply networks based on partial differential equations. *SIAM J. Sci. Comput.*, 30(3):1490–1507, 2008.

[38] M. Garavello and B. Piccoli. Traffic flow on a road network using the Aw-Rascle model. *Comm. Partial Diff. Equ.*, 31(1-3):243–275, 2006.

[39] M. Garavello and B. Piccoli. *Traffic flow on networks*, volume 1 of *AIMS Series on Applied Mathematics*. American Institute of Mathematical Sciences (AIMS), Springfield, MO, 2006.

[40] T. Gaugele, F. Fleissner, and P. Eberhard. Simulation of material tests using meshfree lagrangian particle methods. In *Proc. IMechE*, volume 222, pages 327–338, 2008. Part K: Multi-body Dynamics.

[41] T. Gimse. Conservation laws with discontinuous flux functions. *SIAM J. Math. Anal.*, 24(2):279–289, 1993.

[42] T. Gimse and N. H. Risebro. Solution of the cauchy problem for a conservation law with a discontinuous flux. *SIAM J. Math. Anal.*, 23:635–648, 1992.

[43] S. Göttlich. *Continuous Models for Production Networks Including Optimization Issues*. PhD thesis, Technische Universität Kaiserslautern, 2007.

[44] S. Göttlich, M. Herty, and A. Klar. Network models for supply chains. *Commun. Math. Sci.*, 3(4):545–559, 2005.

[45] S. Göttlich, M. Herty, and A. Klar. Modelling and optimization of supply chains on complex networks. *Commun. Math. Sci.*, 4(2):315–330, 2006.

[46] S. Göttlich, M. Herty, and C. Ringhofer. Optimization of order policies in supply networks. *European J. Oper. Res.*, 202(2):456–465, 2010.

[47] S. Göttlich, S. Hoher, P. Schindler, V. Schleper, and A. Verl. Modeling, simulation and validation of material flow on conveyor belts. *Applied Mathematical Modelling*, 38(13):3295–3313, 2014.

[48] S. Göttlich, A. Klar, and P. Schindler. Discontinuous conservation laws for production networks with finite buffers. *SIAM J. Appl. Math.*, 73(3):1117–1138, 2013.

[49] S. Göttlich, A. Klar, and S. Tiwari. A model hierarchy for complex material flow problems: a mean field approach and particle methods. submitted, 2013. 14 pages.

[50] S. Göttlich and O. Kolb. A continuous buffer allocation model using stochastic processes. submitted, 2013.

[51] S. Göttlich, O. Kolb, and S. Kühn. Optimization for a special class of traffic flow models: combinatorial and continuous approaches. *accepted to Netw. and Heterog. Media*, pages 1–20, 2014.

[52] S. Göttlich, S. Kühn, J. Schwarz, and R. Stolletz. Approximations of time-dependent unreliable flow lines with finite buffers. submitted, 2013.

[53] S. Göttlich and P. Schindler. Optimal inflow control of production systems with finite buffers. submitted, 2013.

[54] S. Göttlich, U. Ziegler, and M. Herty. Numerical discretization of hamilton-jacobi equations on networks. *Netw. Heterog. Media*, 8(3):685–705, 2013.

[55] D. Gottlieb and C.-W. Shu. On the gibbs phenomenon and its resolution. *SIAM Rev.*, 39:644–668, 1997.

[56] V. Gowda and J. Jaffré. *A discontinuous finite element method for scalar nonlinear conservation laws.* Rapport de recherche INRIA. Institut National de Recherche en Informatique et en Automatique, 1993.

[57] D. Helbing. *Verkehrsdynamik: Neue physikalische Modellierungskonzepte.* Springer, 1997.

[58] D. Helbing, S. Lämmer, T. Seidel, P. Seba, and T. Platkowski. Physics, stability and dynamics of supply networks. *Physical Review E*, 70:066116, 2004.

[59] M. Herty, Ch. Joerres, and B. Piccoli. Existence of solution to supply chain models based on partial differential equation with discontinuous flux function. *J. Math. Analysis and Applications*, 401(2):510–517, 2013.

[60] M. Herty and A. Klar. Modeling, simulation, and optimization of traffic flow networks. *SIAM J. Sci. Comput.*, 25(3):1066–1087, 2003.

[61] M. Herty and A. Klar. Modelling and optimization of traffic networks. *SIAM J. Sci. Comp.*, 25:1066, 2004.

[62] M. Herty and A. Klar. Simplified dynamics and optimization of large scale traffic networks. *Math. Models Methods Appl. Sci.*, 14(4):579–601, 2004.

[63] M. Herty, R. Pinnau, and M. Seaid. Optimal control in radiative transfer. *Optimization Methods and Software*, 22(6):917–936, 2007.

[64] M. Herty and M. Rascle. Coupling conditions for a class of second-order models for traffic flow. *SIAM J. Math. Anal.*, 38(2):595–616, 2006.

[65] J. S. Hesthaven and S. M. Kirby. Filtering in legendre spectral methods. *Math. Comput.*, 77(263):1425–1452, 2008.

[66] J. S. Hesthaven and T. Warburton. *Nodal Discontinuous Galerkin Methods - Algorithms, Analysis, and Applications*. Springer-Verlag, 2008.

[67] S. Hoher, P. Schindler, S. Göttlich, V. Schleper, and S. Röck. System dynamic models and real-time simulation of complex material flow systems. In H. A. ElMaraghy, editor, *Enabling Manufacturing competitiveness and economic sustainability, Part 3*, pages 316–321. Springer, 2012.

[68] H. Holden and N. H. Risebro. A mathematical model of traffic flow on a network of unidirectional roads. *SIAM J. Math. Anal.*, 26(4):999–1017, 1995.

[69] H. Holden and N. H. Risebro. *Front Tracking for Hyperbolic Conservation Laws*. Springer-Verlag, 2002.

[70] H. Hoteit, P. Ackerer, R. Mosé, J. Erhel, and B. Philippe. New two-dimensional slope limiters for discontinuous galerkin methods on arbitrary meshes. *Int. J. Numer. Methods Eng.*, 61(14):2566–2593, 2004.

[71] IBM ILOG CPLEX. Optimization studio 12.5.0.0. Informations available at http://ibm.com/software/products/ibmilogcpleoptistud/.

[72] A. Jabbarzadeh and R. I. Tanner. Molecular dynamics simulation and its application to nano-rheology. *Rheology Reviews*, pages 165–216, 2006.

[73] J. Kallrath. *Gemischt-Ganzzahlige Optimierung: Modellierung in der Praxis*. Vieweg Verlag, Wiesbaden, 2002.

[74] S. Kantorovich, R. Weeber, J. J. Cerdà, and C. Holm. Magnetic particles with shifted dipoles. *Journal of Magnetism and Magnetic Materials*, 323(10):1269–1272, 2011.

[75] C. Kirchner, M. Herty, S. Göttlich, and A. Klar. Optimal control for continuous supply network models. *Netw. Heterog. Media*, 1(4):675–688, 2006.

[76] C. Klingenberg and N. H. Risebro. Convex conservation laws with discontinuous coefficients. existence, uniqueness and asymptotic behavior. *Commun. Partial Differ. Equations*, 20(11-12):1959–1990, 1995.

[77] J. W. Landry, G. S. Grest, L. E. Silbert, and S. J. Plimpton. Confined granular packings: Structure, stress, and forces. *Phys. Rev. E*, 67(4):041303, 2003.

[78] P. A. Langston, U. Tüzün, and D. M. Heyes. Discrete element simulation of granular flow in 2d and 3d hoppers: Dependence of discharge rate and wall stress on particle interactions. *Chemical Engineering Science*, 50(6):967–987, 1995.

[79] R. J. LeVeque. *Finite volume methods for hyperbolic problems*. Cambridge Texts in Applied Mathematics. Cambridge University Press, Cambridge, 2002.

[80] J. P Lions. *Optimal control of systems governed by partial differential equations*. Springer-Verlag, New York, 1971.

[81] Y. Lu, S. C. Wong, M. Wang, and C.-W. Shu. The entropy solutions for the Lighthill-Whitham-Richards traffic flow model with a discontinuous flow-density relationship. *Transportation Science*, 43(4):511–530, 2009.

[82] S. Martin and J. Vovelle. Convergence of implicit finite volume schemes for scalar conservation laws with discontinuous flux function. *ESAIM, Math. Model. Numer. Anal.*, 42(5):699–727, 2008.

[83] MATLAB. *Version 7.10.0 (R2010a)*. The MathWorks Inc., Natick, Massachusetts, 2010.

[84] G. F. Newell. A simplified theory of kinematic waves in highway traffic. *Transp. Research B*, 27:281–313, 1993.

[85] J. Nocedal and S. J. Wright. *Numerical optimization. 2nd ed.* Springer Series in Operations Research and Financial Engineering. New York, NY: Springer. xxii, 664 p., 2006.

[86] J. Palaniappan, S. T. Miller, and R. B. Haber. Sub-cell shock capturing and spacetime discontinuity tracking for nonlinear conservation laws. *Int. J. Numer. Methods Fluids*, 57(9):1115–1135, 2008.

[87] P. O. Persson and J. Peraire. Sub-cell shock capturing for discontinuous galerkin methods. *American Institute of Aeronautics and Astronautics*, 2006.

[88] V. L. Popov. *Contact Mechanics and Friction. Physical Principles and Applications*. Springer, 2010.

[89] W. T. Reeves. Particle systems–a technique for modeling a class of fuzzy objects. *SIGGRAPH Comput. Graph.*, 17(3):359–375, 1983.

[90] G. Reinhart and F.-F. Lacour. Physically based virtual commissioning of material flow intensive manufacturing plants. In H. A. Zaeh, M. F.; El-Maraghy, editor, *3rd International Conference on Changeable, Agile, Reconfigurable and Virtual Production (CARV 2009)*, pages 377–387, Munich, 2009.

[91] S. Röck. Hardware in the loop simulation of production systems dynamics. *Production Engineering*, 5:329–337, 2011.

[92] M. W. P. Savelsbergh. Preprocessing and probing techniques for mixed integer programming problems. *ORSA J. Comput.*, 6(4):445 – 454, 1994.

[93] P. Spellucci. *Numerical methods of nonlinear optimization. (Numerische Verfahren der nichtlinearen Optimierung.).* ISNM Lehrbuch. Basel: Birkhäuser Verlag. 576 S. , 1993.

[94] R. Stolletz and S. Weiss. Buffer allocation using exact linear programming formulations and sampling approaches. In *7th Conference on Manufacturing Modelling, Management, and Control*, 2013.

[95] U. H. Suhl and R. Szymanski. Supernode processing of mixed-integer models. *Comput. Optim. Appl.*, 3(4):317 – 331, 1994.

[96] J. Towers. Convergence of a difference scheme for conservation laws with a discontinuous flux. *SIAM J. Numer. Anal.*, 38(2):681–698, 2000.

[97] J. Towers. A difference scheme for conservation laws with a discontinuous flux - the nonconvex case. *SIAM J. Numer. Anal.*, 39(4):1197–1218, 2001.

[98] R. Trapp. Microscopic traffic flow modeling of large urban networks: Approach and techniques at the example of the city of cologne. *81st Annual Meeting of the Transportation Research Board (CD-ROM)*, 2002.

[99] F. Troeltzsch. *Optimal control of partial differential equations. Theory, procedures, and applications. (Optimale Steuerung partieller Differentialgleichungen. Theorie, Verfahren und Anwendungen.).* Wiesbaden: Vieweg. x, 297 p. , 2005.

[100] S. Ulbrich. Adjoint-based derivative computations for the optimal control of discontinuous solutions of hyperbolic conservation laws. *Syst. Control Lett.*, 48(3-4):313–328, 2003.

[101] S. Weiss and R. Stolletz. First results on: A benders decomposition approach for the optimization of flow lines with stochastic processing times. In *9th Conference on Stochastic Models of Manufacturing and Service Operations*, 2013.

[102] J. K. Wiens, J. M. Stockie, and J. F. Williams. Riemann solver for a kinematic wave traffic model with a discontinuous flux. *J. Comput. Phys.*, 242:1–23, 2013.

[103] G. Wünsch. Realtime collision detection and rigid body simulation for the digital assembly automation. In *CARV 2009 - 3rd International Conference on Changeable, Agile, Reconfigurable and Virtual Production*, pages 899–907, 2009.

[104] H. P. Zhu and A. B. Yu. Averaging method of granular materials. *Phys. Rev. E*, 66:021302, 2002.

[105] H. P. Zhu and A. B. Yu. Micromechanic modeling and analysis of unsteady-state granular flow in a cylindrical hopper. *J. Engrg. Math.*, 52(1-3):307–320, 2005.

[106] U. Ziegler. *Mathematical Modelling, Simulation and Optimisation of Dynamic Transportation Networks*. PhD thesis, RWTH Aachen Universität, 2012.