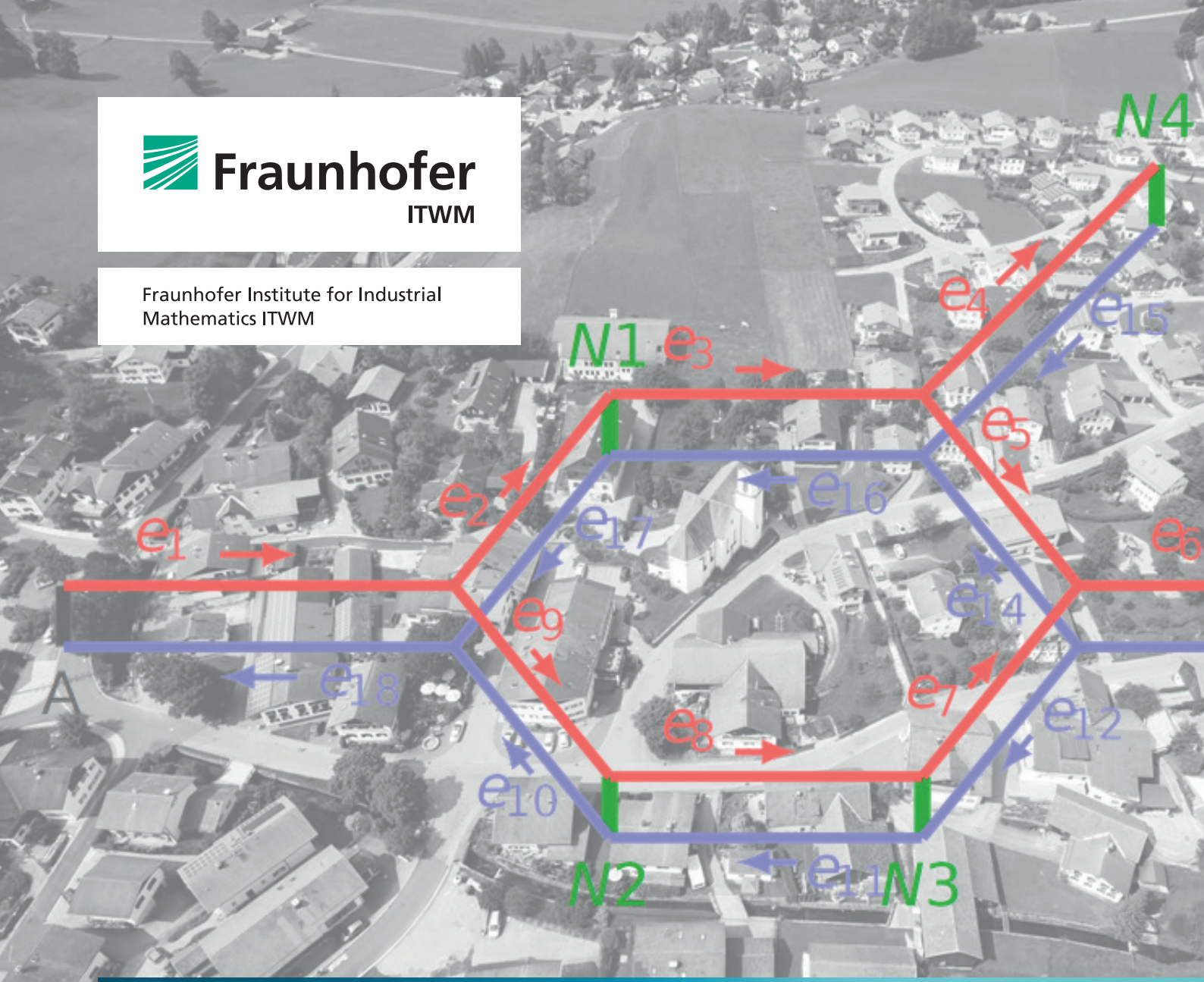

Fraunhofer ITWM

Fraunhofer Institute for Industrial Mathematics ITWM

Modeling and Efficient Simulation of District Heating Network

Matthias Eimer

TECHNISCHE UNIVERSITÄT KAISERSLAUTERN

Fraunhofer Verlag

Fraunhofer Institute for
Industrial Mathematics ITWM

Modeling and Efficient Simulation of District Heating Network

Matthias Eimer

Fraunhofer Verlag

Contact:
Fraunhofer Institute for
Industrial Mathematics ITWM
Fraunhofer-Platz 1
67663 Kaiserslautern
Germany
Phone +49 631/31600-0
info@itwm.fraunhofer.de
www.itwm.fraunhofer.de

Bibliographic information of the German National Library:
The German National Library has listed this publication in its Deutsche Nationalbibliografie; detailed bibliographic data is available on the internet at www.dnb.de.

ISBN (print version): 978-3-8396-1853-0
DOI (free PDF version): https://doi.org/10.24406/publica-187

DE-386

Zugl.: Kaiserslautern, TU, Diss., 2021

Print and finishing:
Fraunhofer Verlag, Mediendienstleistungen

The book was printed with chlorine- and acid-free paper.

Nobelstrasse 12
70569 Stuttgart
Germany
verlag@fraunhofer.de
www.verlag.fraunhofer.de

is a constituent entity of the Fraunhofer-Gesellschaft, and as such has no separate legal status.

Fraunhofer-Gesellschaft zur Förderung
der angewandten Forschung e.V.
Hansastrasse 27 c
80686 München
Germany
www.fraunhofer.de

Technische Universität Kaiserslautern

Dissertation

Modeling and Efficient Simulation of District Heating Networks

---

Autor:
Matthias Eimer

Gutachter:
PD. Dr. Raul Borsche
TU Kaiserslautern
DR CNRS Dr. Raphaël Loubère
University of Bordeaux

Vom Fachbereich Mathematik
der Technischen Universität Kaiserslautern
zur Verleihung des akademischen Grades
Doktor der Naturwissenschaften
(Doctor rerum naturalium, Dr. rer. nat.)
genehmigte Dissertation

Datum der Disputation: 28.10.2021

DE-386

Acknowledgements

First, I would like to thank my supervisor Raul Borsche for his support, his patience and the many fruitful discussions. His guidance and suggestions improved this thesis in many respects.

I want to thank the transport processes department at ITWM for the possibility to work on this project. Especially, I am thankful for the support of Norbert Siedow and Jan Mohring at ITWM and their valuable feedback during my research. Furthermore, I want to thank my colleagues from the EMO group, Dominik Linn, Markus Rein and Jaroslaw Wlazlo for the great working atmosphere and the many discussions. I extend my thanks to the whole TV Mensa group for their company during the last four years.

Finally, i want thank my family for always believing in me. I want to express my deepest gratitude to my partner Nina Baumgarten for her constant motivation, trust and love. Thank you so much for being by my side.

Abstract

The current trend towards renewable energy sources raises the importance of their efficient use. An accurate simulation of the energy transport through the distribution network is key to the emerging optimization task. District heating is a powerful technology that connects great flexibility in the used energy source with the much-needed buffering and storage effect. Heated water is distributed through a network of pipes to many different buildings providing space heat and domestic hot water.

This thesis tackles the modeling and efficient simulation of district heating networks. Due to the large scale and complex dynamical behavior of such systems, this is a challenging task. The obtained results build an important foundation for the upcoming optimization problems. The dynamical behavior of water in the pipes is described by the incompressible Euler equations. Conservation of the involved quantities determines the coupling at each junction in the network, leading to a system of partial differential algebraic equations. For that system, the unique existence of a solution is shown. Furthermore, a stability estimate for the dependence on important parameters is derived.

A new local time stepping algorithm is presented, that is perfectly suited for the solution of the transport problem involved. In comparison to generic high order ADER schemes its high efficiency outperforms the classical approach significantly. In order to enable computation of vary large time steps, implicit methods are investigated. A high order finite volume method is equipped with an *a posteriori* limiter. The superior behavior of the constructed hybrid scheme is shown in different numerical tests and applications.

Zusammenfassung

Durch den aktuellen Trend zu erneuerbaren Energiequellen richtet sich das Interesse auch verstärkt auf deren effizienten Nutzung. Eine genaue Simulation des Energietransports durch das Verteilungsnetz ist der Schlüssel zur entstehenden Optimierungsaufgabe. Fernwärme ist eine leistungsstarke Technologie, die eine große Flexibilität in der genutzten Energiequelle mit dem dringend benötigten Puffer- und Speichereffekt verbindet. Erhitztes Wasser wird über ein Rohrnetz an viele verschiedene Verbraucher verteilt, um Raumwärme und Warmwasser bereitzustellen.

Die vorliegende Arbeit beschäftigt sich mit der Modellierung und effizienten Simulation von Fernwärmenetzen. Aufgrund des komplexen dynamischen Verhaltens solcher Systeme ist dies eine anspruchsvolle Aufgabe. Die gewonnenen Erkenntnisse bilden eine wichtige Grundlage für die anstehenden Optimierungsprobleme. Das dynamische Verhalten von Wasser in den Rohren wird durch die inkompressiblen Euler-Gleichungen beschrieben. Die Rohrgleichungen koppeln über die Erhaltung der beteiligten Größen an jedem Knotenpunkt im Netzwerk, was zu einem System von partiellen algebraischen Differentialgleichungen führt. Für dieses System wird die eindeutige Existenz einer Lösung gezeigt. Weiterhin wird eine Stabilitätsabschätzung für die Abhängigkeit von wichtigen Systemparametern abgeleitet.

Das in der Arbeit vorgestellte lokale Zeitschrittverfahren eignet sich hervorragend für die Lösung des vorliegenden Transportproblems. Im Vergleich zu generischen ADER-Methoden hoher Ordnung übertrifft seine enorme Effizienz den klassischen Ansatz deutlich. Um die Berechnung von wesentlich größeren Zeitschritten zu ermöglichen, werden implizite Verfahren untersucht. Dafür wurde ein Finite-Volumen-Verfahren hoher Ordnung wird mit einem *a posteriori*-Limiter kombiniert. Die Vorteile des so konstruierten Hybridschemas werden in verschiedenen numerischen Testbeispielen und praktischen Anwendungsfällen gezeigt.

Publications

In the course of the research leading to this thesis, the following publications were written. Parts of the listed publications are contained in this thesis.

- R. Borsche, **M. Eimer**, N. Siedow; A local time stepping method for thermal energy transport in district heating networks, Applied Mathematics and Computation 19, https://doi.org/10.1016/j.amc.2019.01.072
- **M. Eimer**, R.Borsche, N.Siedow; Local time stepping method for district heating networks, Proceedings in Applied Mathematics and Mechanics 19, https://doi.org/10.1007/978-3-030-27550-1_50
- J. Mohring, D. Linn, **M. Eimer**, M. Rein, N. Siedow; District heating networks - dynamic simulation and optimal operation, Mathematical MSO for Power Engineering and Management, 2021, https://doi.org/10.1007/978-3-030-62732-4_14
- **M. Eimer**, R. Borsche, N. Siedow; Implicit finite volume method with a posteriori limiting for transport networks; Advances in Computational Mathematics, 2022, https://doi.org/10.1007/s10444-022-09939-1

Contents

Chapter 1

Introduction

Energy production, distribution and usage are important aspects of the political as well as technological development. Different new technologies are on the rise. The energy production by renewable energy sources such as solar and wind energy is pushed forward. This trend effects a structural change in the production and distribution infrastructure. One of the main challenges in achieving 100% green energy is the high volatility and weather dependence in most production forms. This means together with the production infrastructure, efficient storage and buffering technologies have to be developed.

On a local level, district heating (DH) is a powerful technology that might be able to solve some problems of the current energy transition. The power plant usually combines heat and power production and is much more efficient than pure power factories. Furthermore, located in industrial areas, the excess heat of nearby production facilities can be used in the heating network as well. That enables the use of energy that otherwise would be lost. Moreover the ability to filter exhaust gases is much better compared to the smaller scale heat boilers of single households. In that way the overall emissions of carbon and particular matter are reduced. In the near future, the ability to feed thermal energy to the network will also be assessed by the classical consumers. So-called 'prosumers' will be able to primarily use locally available heat from solar panels, where excess energy can be deposited in the network, while on the other hand the network can top up shortages. This system is already common practice in electrical power production with photovoltaics for electrical power.

Efficient use of district heating networks as energy storage and buffer requires precise knowledge of the dynamics involved. However, the optimization of such networks is a challenging task. The dynamic behaviour and direct coupling to certain events enforces real time optimization. A typical control interval in which new inputs need to be proposed is 15 minutes. The large number of degrees of freedom in the resulting system calls for efficient numerical methods. This work addresses both, the mathematical modeling of district heating networks and its efficient simulation.

The working principle of district heating is rather simple and shown in Figure 1.1. Thermal energy is transferred to water at a power plant. A network of pipes is responsible for the transportation of the heated water to different buildings, as for example

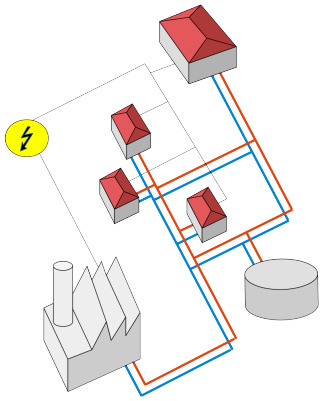

Figure 1.1: Schematic illustration of a district heating network: A power plant (bottom) is connected to a number of houses by two parallel sets of pipes (red and blue).

family houses, public buildings or office buildings. At the consumer site, heat exchangers withdraw the thermal energy from the district heating network to the local heating circuit. That way, no water is extracted from the DH network, instead it gets cooled down to a reference temperature. The cooler water then returns back to the power plant to be reheated again.

The large potential of DH systems is twofold: In most cases, the power plant is a so-called combined heat and power (CHP) plant, where the primary energy sources can be used either for heating the network or for generating electrical power in generators. The transition between the two is continuous, such that at any time the available energy can be split arbitrarily. This gives rise to an optimization problem, where the aim is to get the highest gain from selling electrical energy, while fulfilling all heat demands in the network.

Technische Werke Ludwigshafen is the DH operator that provided us with the necessary data. In their specific setting, there is another optimization potential. A waste incineration plant provides a relatively constant energy flow over the course of the day, but consumer behaviour is in general highly volatile and depends on daytime and current weather conditions. Whenever the energy provided by waste incineration does not suffice the demand, additional gas boilers compensate the difference. Due to the time delay between the injection of a specific temperature and its actual usage at the customers, it is possible to 'preheat' the network. The energy deposition at the consumers contains the product of temperature difference times flow velocity. Some hours before the peak load in the consumer demand, the operator feeds higher temperatures to the network. That way, the additional burning of gas can be avoided, saving a large fraction of operational cost.

Most commercial tools for the simulation and planning of district heating networks use a stationary or quasistationary formulation of the energy transport [32],[57]. However for live optimization tasks, a full dynamical model is required. The flow of inviscid fluids through a pipe can be modeled by the Euler equations

$$\begin{aligned} \partial_t \rho + \nabla \cdot (\rho \vec{v}) &= 0 \\ \partial_t (\rho \vec{v}) + \nabla \cdot (\rho \vec{v} \otimes \vec{v}) + \nabla p &= f \\ \partial_t E + \nabla \left((E + p) \vec{v} \right) &= S, \end{aligned} \tag{1.1}$$

where ρ is the fluid density, $\vec{v}$ the flow velocity, p the pressure and E the energy of the fluid. The terms f and S are used to model friction terms and energy dissipation. Especially in the context of gas networks, the Euler equations are well-studied [33],[50],[77]. When the ratio between the flow velocity of the gas and the propagation speed of pressure differences, also called Mach number, is small it behaves similar to an incompressible fluid. In recent years, many schemes that have an asymptotic preserving property have been developed [24],[63]. That means that they are able to capture the correct limits, if the Mach number converges to zero. While it might have advantages to use such schemes also in the district heating context, mostly the incompressible Euler equations are used. The general form of the resulting hydrodynamic system is very similar to the ones used to model water supply networks [15],[74] with the additional energy information that is transported.

The most crucial part of district heating dynamics is the possible change of flow directions inside the network. Time varying consumer behaviour and non constant feed temperatures lead to unavoidable zero crossings of velocity. Not only does that introduce contact discontinuities into otherwise possibly smooth solutions, it also raises the question of existence of solutions, that is not trivial to answer. For the pure hydrodynamic part, existence results are available via DAE theory [36]. In [62], local existence for the non isothermal, non isenthropic Euler equations on networks has been proven, using the notion of renormalized solutions [26]. In some specific cases, e.g. specific network topologies or certain friction models, also proving global existence was possible. However, due to the lack of a Lipschitz-type stability estimate for their setting, uniqueness of solutions could not be guaranteed.

Most publications on the topic of simulating district heating networks use either first order discretizations to model the advective transport e.g. [38],[72],[71] or node based methods, where no spatial discretization along the pipes is used [8]. A contribution to second order simulation of the energy transport part is given in [44]. When it comes to the simulation of district heating networks, especially in combination with an optimization task, computational efficiency is crucial. In [59],[61] model order reduction is applied to the system arising from upwind discretization. Those reduced models are used to solve optimization problems in [60].

The contribution of this work addresses both the modeling and the simulation of district heating networks. In a first step, the shortcoming of pure algebraic coupling of transport equations is shown by some analytical examples. By the introduction of small

volumes at the nodes, the stability of solutions in the space of functions of bounded variations is ensured. We are able to show unique existence of solutions to the two separate systems, together with a stability estimate on the dependence of parameters. A sequence of solutions to the separate systems is shown to converge towards a unique solution of the coupled problem by a contraction argument. Altogether we were able to show uniqueness of solutions to the full problem with Lipschitz-continuous stability in the parameters. This emerged from joint work together with Mauro Garavello, Elena Rossi and Raul Borsche. A separate publication on that matter is currently in preparation.

Secondly, different possibilities in the numerical simulation of the transport part were explored. Concerning explicit methods, a new local time stepping scheme is presented. The special structure of the scheme allows very efficient evaluation and the computational complexity is by one order of magnitude better than e.g. generic ADER finite volume schemes. The corresponding results have already be published in [10].

Furthermore, implicit methods have been studied in order to enable the computation over larger time horizons. We present a generic construction of high order finite volume schemes and analyze their stability. The oscillations near sharp gradients are tackled with an a posteriori limiting technique. Originally designed for multidimensional finite volume schemes for systems of conservation laws, the MOOD limiter [20] is perfectly suited for this application. It is able to eliminate most disadvantages of classical limiters, when applied to large time steps. This is only possible due to a marching formulation and thus sequential evaluation of the scheme. We show the superior behavior of the new hybrid scheme in contrast to first order methods and its applicability for large time steps. A publication on that matter is submitted to "Advances in computational mathematics", a preprint version is available in [30].

This thesis is structured as follows. In Chapter 2 a mathematical model describing the dynamics in district heating networks is presented. Starting with a brief motivation of the Euler equations, the network structure as a graph is introduced. We describe all important components of the district heating network together with their respective governing equations. Then a splitting of the complete system into a hydrodynamic part and a transport part is motivated. The analytical properties of the model are investigated in Chapter 3. For both split parts a proof for unique existence of solutions is given. Furthermore, we show that the solutions to the subsystems fulfill a stability estimate. This enables the construction of a unique solution to the coupled system by Banach's fixed point theorem. The Chapters 4 to 5 discuss numerical methods for the solution of transport problems on networks. Different explicit schemes are explored at first, where the focus lies on efficient computation of high order accurate solutions. ADER schemes as well as a local time stepping scheme are described, where a suitable coupling transfers their classical formulation to the network setting. In order to expand the time steps beyond the classical CFL condition, implicit methods are discussed. Starting with two first order methods, the accuracy is increased by the use of high order implicit finite volume schemes. The novel application of a posteriori limiting

to this type of problem reduces the amount of spurious oscillations significantly. The simulation of the hydraulic part is briefly covered in Chapter 6. Chapter 7 collects the numerical results for the presented explicit and implicit schemes. Different test cases are used to emphasize specific properties of the selected methods. Furthermore, the methods are applied to real district heating networks, illustrating realistic operation settings. Finally a conclusion of the findings in this work is given in Chapter 8. Several aspects are suggested that might be worth investigating in the future.

Chapter 2

Model

In this chapter we derive a model describing the dynamics in a district heating network. Starting from the formulation with the full Euler equations for single pipes, the transition to incompressible one dimensional flow is motivated. Using some basic graph theory notations we define a set of incidence matrices describing the network topology. By providing suitable coupling and boundary conditions the one dimensional formulation extends to a system of equations on a network. Additionally to the Euler equations on the pipes we describe the dynamics of other components involved such as consumers and the power plant. Due to some special properties of the resulting system, the model is split into a transport equation for internal energy density and a hydraulic part responsible for the evaluation of flow velocities.

2.1 Euler Equations

First of all, we derive the basic equations for the dynamics of a fluid in a pipe. They state the conservation of the three essential quantities of fluid flow, namely mass, momentum and energy. We give a brief derivation of the essential parts, for more details on the Euler equations we refer to textbooks as e.g. [18],[49],[64].

2.1.1 Continuity Equation

Consider a fixed domain $\Omega \in \mathbb{R}^d$. The total mass of a continuous fluid for any time t is the volume integral over its density ρ

$$M(t) = \int_\Omega \rho(t, \vec{x}) \mathrm{d}V.$$

The change of mass over time is given by the flow of mass over the surface of the domain

$$\frac{d}{dt} M(t) = \frac{d}{dt} \int_\Omega \rho \mathrm{d}V = -\int_{\partial\Omega} \rho \vec{v} \cdot \vec{n} \mathrm{d}S,$$

where $\vec{v}$ is the flow velocity and $\vec{n}$ is the outer normal vector of the domain. Since the domain does not change, the time derivative can be written inside the integral.

Application of Gauss divergence theorem leads to

$$\int_\Omega \frac{d}{dt}\rho + \nabla \cdot (\rho\vec{v})\mathrm{d}V = 0.$$

As this equation holds for all Ω, the integrand has to be zero. The resulting equation is called **continuity equation** and states the conservation of mass

$$\partial_t \rho + \nabla \cdot (\rho\vec{v}) = 0. \tag{2.1}$$

2.1.2 Momentum Equation

The second quantity of the fluid we are looking at is the momentum

$$\int_\Omega \rho\vec{v}\mathrm{d}V.$$

The momentum of a given fluid element is changing due to the forces acting on it. Volume forces, or body forces, act on the whole mass element and are proportional to the density

$$\mathrm{d}F_1 = \int_\Omega \rho\vec{F}\mathrm{d}V. \tag{2.2}$$

Surface forces just act on the boundary of the domain. Here we will assume inviscid fluids, which means that all those forces will act perpendicular to the surface. In this case, the only contribution is given by the pressure p which leads to

$$\mathrm{d}F_2 = -\int_{\partial\Omega} p\vec{n}\mathrm{d}S. \tag{2.3}$$

The general case incorporating shear stress would lead to the Navier-Stokes equations [14], which are not in the scope of this thesis. Furthermore, as above in the continuity equation, momentum can enter the considered domain by a flow over the boundary. The surface integrals can again be rewritten in their volume integral form using the divergence theorem, we get the balance of momentum as

$$\frac{d}{dt}\int_\Omega \rho\vec{v}\mathrm{d}V = \int_\Omega \rho\vec{F} - \nabla p - \nabla \cdot (\rho\vec{v} \otimes \vec{v})\mathrm{d}V.$$

The time derivative can be written inside the integral again and due to generality of Ω we get

$$\partial_t(\rho\vec{v}) + \nabla p + \nabla \cdot (\rho\vec{v} \otimes \vec{v}) = \rho\vec{F}. \tag{2.4}$$

This is the second equation of the Euler system and called the **momentum equation**.

2.1.3 Energy Equation

At last, we have the conservation of energy. In Section 2.1.2 we already noted the forces acting on the fluid element. The change of energy is the scalar product of force times velocity per time frame. This means we get contributions similar to 2.2 and 2.3. Furthermore, the thermal energy loss/gain is given by the heat flux $\vec{q} \cdot \vec{n}$ across the surface. Altogether, the change of energy is

$$\mathrm{d}E = \left(\int_\Omega \rho \vec{F} \cdot \vec{v} \mathrm{d}V - \int_{\partial\Omega} p\vec{n} \cdot \vec{v} \mathrm{d}S - \int_{\partial\Omega} \vec{q} \cdot \vec{n} \mathrm{d}S \right) \mathrm{d}t.$$

Similar to the continuity equation, we can apply Gauss' divergence theorem and get the energy balance of the form

$$\partial_t E + \nabla \left((E + p)\vec{v} \right) = \rho \vec{F} \cdot \vec{v} - \nabla \cdot \vec{q}. \tag{2.5}$$

The energy has two different contributing parts, namely the inner energy density e and the kinetic energy $\frac{1}{2}\rho v^2$

$$E = \rho e + \frac{1}{2}\rho v^2.$$

Together, the three equations (2.1),(2.4),(2.5) form the Euler equations. In their classical form, when no external forces act on the system, the full Euler system reads

$$\begin{aligned} \partial_t \rho + \nabla \cdot (\rho \vec{v}) &= 0 \\ \partial_t (\rho \vec{v}) + \nabla \cdot (\rho \vec{v} \otimes \vec{v}) + \nabla p &= f \\ \partial_t E + \nabla \left((E + p)\vec{v} \right) &= S. \end{aligned} \tag{2.6}$$

The exact form of the right hand sides is introduced when the equations and parameters of real pipelines are motivated in Section 2.2.2. System (2.6) consists of three equations depending on the four different variables ρ, v, p, e. Depending on the specific setting, it has to be closed by a suitable equation of state. The classical closure relation for polytropic gas is

$$E = \frac{p}{1 - \gamma} + \frac{1}{2}\rho v^2. \tag{2.7}$$

For water in its liquid phase, the ideal gas law does not apply. Most equations of state for liquids involve interpolation of experimental data in the problem specific operation conditions [75],[55]. In the following section we introduce the incompressible system, which we will close by either a constant density $\rho = const$ or a pure energy dependent density function $\rho = \rho(e)$.

2.1.4 One Dimensional, Incompressible Flow

Compared to the total length of the computational domain, the diameter is very small. Typically the lengths are in the range of 100-1000 meters, while the diameter is between 0.05-0.4 meters. As dynamics on the length scales of the diameter do not play

a large role, the 3d system is reduced to a system of one dimensional averaged values. Furthermore we ignore the pressure contribution as well as the kinetic energy part in the energy equation, i.e. we use the energy formulation

$$E = \rho e.$$

This leads to the following one dimensional system

$$\begin{aligned} \partial_t \rho + \partial_x(v\rho) &= 0 \\ \partial_t(v\rho) + \partial_x(v^2\rho) + \partial_x p &= f(\rho, v) \\ \partial_t e + \partial_x(ve) &= S(e) \end{aligned} \tag{2.8}$$

where f is a friction term and S is a term modeling a heat sink.
The density of water changes depending on its current temperature. We therefore close above model with an equation of state of the form $\rho = \rho(e)$ and we get

$$\begin{aligned} \rho(e)_t + (v\rho(e))_x &= 0 \\ \Rightarrow \rho' e_t + v_x \rho(e) + v\rho' e_x &= 0. \end{aligned}$$

We insert the energy equation

$$e_t + (ve)_x = S(e)$$

and get

$$v_x = \frac{\rho'}{\rho} S(e).$$

The derivative $\frac{\partial \rho}{\partial e}$ is denoted by ρ'. For water, the derivative of the density is much smaller than the density itself, leading to a right hand side that is very small. Setting it equal to zero leads to the incompressibility constraint

$$v_x = 0.$$

Insertion into (2.8) leads to the incompressible Euler equations

$$\begin{aligned} \partial_x v &= 0 \\ \rho \partial_t v + \partial_x p &= f(\rho, v) \\ \partial_t e + v \partial_x e &= S(e). \end{aligned} \tag{2.9}$$

In order to better understand the value ranges of the involved variables, they can be transformed to dimensionless form. This means every physical quantity is scaled by a reference value. The dimensionless quantities should then all be of a similar order of magnitude in their usual range. Especially for the numerical solution of the equations this procedure is advisable, as large discrepancies in variable ranges can lead to badly

conditioned systems and large round-off errors. In order to transform to a dimensionless system, we define:

$$\begin{aligned} \tilde{x} &= \frac{x}{x_r}, & \tilde{v} &= \frac{v}{v_r} = \frac{v x_r}{t_r}, \\ \tilde{t} &= \frac{t}{t_r}, & \tilde{\rho} &= \frac{\rho}{\rho_r}, \\ \tilde{p} &= \frac{p}{p_r}, & \tilde{e} &= \frac{e}{e_r}. \end{aligned} \tag{2.10}$$

Inserting the scaled variables in (2.9) and dropping the tildes we end up with a system that looks very similar:

$$\begin{aligned} \partial_x v &= 0 \\ \rho \partial_t v + \frac{1}{\epsilon^2} \partial_x p &= f(\rho, v) \\ \partial_t e + v \partial_x e &= S(e), \end{aligned} \tag{2.11}$$

with the dimensionless parameter

$$\epsilon^2 = \frac{v_r^2 \rho_r}{p_r}.$$

This means that except the pressure, all variables have the same order of magnitude.

2.2 Network Model

The Euler equations derived in Section 2.1 describe the fluid flow in a single pipeline. In order to formulate the full model, we need to connect the different pipelines by suitable coupling conditions. Furthermore, there are also non-pipe components in the network, with different equations describing their flow behavior. First of all, some basic definitions from graph theory are given, then all the network components are presented. In the end the full system of equations describing the dynamics of district heating networks is shown.

2.2.1 Basic Graph Notations

For the description of the network as a graph and for some reformulations and proofs later, we need some basic definitions [39].

Definition 2.2.1 *A **graph** is a tuple $\mathcal{G} = (\mathcal{V}, \mathcal{J})$ where*

1. *$\mathcal{V} = \{V_1, \ldots, V_{N_\mathcal{V}}\}$ is the node set with $N_\mathcal{V}$ nodes.*
2. *$\mathcal{J} = \{J_1, \ldots, J_{N_\mathcal{J}}\} \subset \mathcal{V} \times \mathcal{V}$ is the edge set with $N_\mathcal{J}$ edges.*

3. *For an edge $J = (V_1, V_2)$ we call V_1 **starting node** of J and V_2 **end node** of J.*

4. *For a node V, the set $\mathcal{A}(V) = \{J = (V_1, V_2) \in \mathcal{J} | V_1 = V \text{ or } V_2 = V\}$ is called its incidence set.*

For the formulation of coupling conditions later, we need some formulation of orientation of the edges in the graph. We use two different notions, namely the topological orientation and the flow orientation. In order to define the flow orientation, we need an orienting function

$$\sigma : \mathcal{J} \to \{-1, 1\}.$$

Definition 2.2.2 *For a graph $\mathcal{G}$ and a given node V we define the sets of topologically incoming edges and outgoing edges as*

$$\begin{aligned} \mathcal{J}_{in}(V) &= \{J = (V_1, V_2) \in \mathcal{J} | V = V_2\} \\ \mathcal{J}_{out}(V) &= \{J = (V_1, V_2) \in \mathcal{J} | V = V_1\} \end{aligned} \tag{2.12}$$

and for a flow orientation σ we define the sets of flow oriented incoming and outgoing edges as

$$\begin{aligned} \mathcal{I}_\sigma(V) &= \{J \in \mathcal{J}_{in}(V) | \sigma(V) = 1\} \cup \{J \in \mathcal{J}_{out}(V) | \sigma(V) = -1\} \\ \mathcal{O}_\sigma(V) &= \{J \in \mathcal{J}_{in}(V) | \sigma(V) = -1\} \cup \{J \in \mathcal{J}_{out}(V) | \sigma(V) = 1\}. \end{aligned} \tag{2.13}$$

For a velocity field $v : [0, T] \to \mathbb{R}^N$, that for any time t assigns a real velocity to every edge in the network we denote the sets induced by $\sigma = \text{sign}(v(t))$ with $\mathcal{I}_{v(t)}(V)$ and $\mathcal{O}_{v(t)}(V)$ respectively. If the orientation function is obvious, the subscripts are dropped.

The next definitions depend on the whole graph topology, not only on local considerations.

Definition 2.2.3 *For a graph $\mathcal{G}$*

1. *A walk is a sequence $(J_1, J_2, \ldots, J_n)$ with $J_i = (V_{i_1}, V_{i_2}) \in \mathcal{J}$, or $(V_{i_2}, V_{i_1}) \in \mathcal{J}$.*

2. *A path $\mathcal{P}$ is a finite walk, where all edges and nodes are distinct. If V_i is the starting node of $\mathcal{P}$ and V_j is its end node, we call $\mathcal{P}$ a $V_i \to V_j$ path.*

3. *A circle is a finite walk, where all edges and nodes are distinct except the first and last node are equal.*

4. *A graph $\mathcal{G}$ is called connected, if for every pair of nodes $V_i, V_j \in \mathcal{V}$ there exists a path from V_1 to V_2.*

5. *A graph $\mathcal{G}$ is called a tree, if it is connected and does not contain any circles.*

6. *A spanning tree $\mathcal{T}$ of a graph $\mathcal{G}$ is a subgraph that contains all nodes and is a tree.*

For any connected graph, a spanning tree can be constructed by iteratively deleting edges that close a circle. When there are no such edges left, the remaining subgraph is a spanning tree of $\mathcal{G}$. The set of removed circles is denoted by $\mathcal{C}$ and any element $C \in \mathcal{C}$ is composed by one deleted edge $J = (V_1, V_2)$ and the unique $V_2 \to V_1$ path in $\mathcal{T}$. A spanning tree has always $N_\mathcal{V} - 1$ edges, i.e. there are $N_\mathcal{C} = N_\mathcal{J} - N_\mathcal{V} + 1$ linearly independent circles in $\mathcal{C}$.

For computations with graphs, a matrix representation of their topology is helpful. In the incidence matrix $\mathbf{A}^I \in \mathbb{R}^{N_\mathcal{V} \times N_\mathcal{J}}$, the connection information between nodes and edges in the network are stored

$$(\mathbf{A}^I)_{i,j} = \begin{cases} 1, & \text{if } J_j = (V_i, *) \\ -1, & \text{if } J_j = (*, V_i) \\ 0, & \text{else} \end{cases} .$$

Analogously, we define the tree incidence matrix $\mathbf{A}^\mathcal{T}$ as the incidence matrix of the spanning tree $\mathcal{T}$ of $\mathcal{G}$. We partition $\mathbf{A}^I$ into a reduced incidence matrix $\mathbf{A}_r$ and a complementary reduced incidence matrix $\mathbf{A}_r^p$ such that the columns of the latter correspond to the nodes where we later prescribe a pressure p, namely both nodes of the source edge

$$(\mathbf{A}_r \ \mathbf{A}_r^p) = \mathbf{A}^I.$$

In other words, $\mathbf{A}_r^p$ contains the columns of the incidence matrix that correspond to the source nodes. Furthermore, for the cycle set $\mathcal{C} = \{C_1, \dots\}$ we define the matrix $\mathbf{A}^C \in \mathbb{R}^{N_\mathcal{C} \times N_\mathcal{J}}$

$$(\mathbf{A}^C)_{i,j} = \begin{cases} 1, & \text{if } J_j \text{ is in cycle } C_i \\ -1, & \text{if the opposite of } J_j \text{ is in cycle } C_i \\ 0, & \text{else} \end{cases} .$$

For the systems we solve, it is advantageous to have circles with as few edges as possible, increasing the sparsity structure of $\mathbf{A}^\mathcal{C}$. In [44] a heuristic algorithm for the computation of a favorable spanning tree resulting in small circles is presented.

To conclude this section, we want to discuss some general properties of district heating network graphs. When district heating pipes are constructed, typically two parallel sets are built simultaneously, one for the flow, one for the return. This leads to rather special topologies: The full graph can be decomposed into two identical subgraphs and a set of connecting edges. The first subgraph is called the flow network and responsible for the transport of the hot water from the source to the consumers. The second one is the return network and transports the cooled water back to the power plant such that it can be reheated again. Both subgraphs are connected by a set of edges that are either consumers or the source (see Section 2.2.2). Furthermore, the flow direction of those edges is always the same: The source will always draw from the return network and feed the flow network and the consumers operate vice versa. An example network

is shown in Figure 2.1, where the flow part is drawn in red and the return part in blue. The flow in the connecting edges, where green stands for consumers and gray represents the source, is always directing from the red side to the blue side. Note that for other pipes in general, the flow direction can change over time. Most of the important dynamic happens in the flow part of the network, as in most cases, the temperatures in the return part are at the same constant level. Therefore in most considerations and examples in this work we drop the return network. In order to keep a network topology that remains closed in terms of water circulation, we can compress the whole return network into one single node, compare Figure 2.2. Since the length of source and consumer does not appear in their respective equations, the different scalings in the figures have no impact on the resulting system of the flow network.

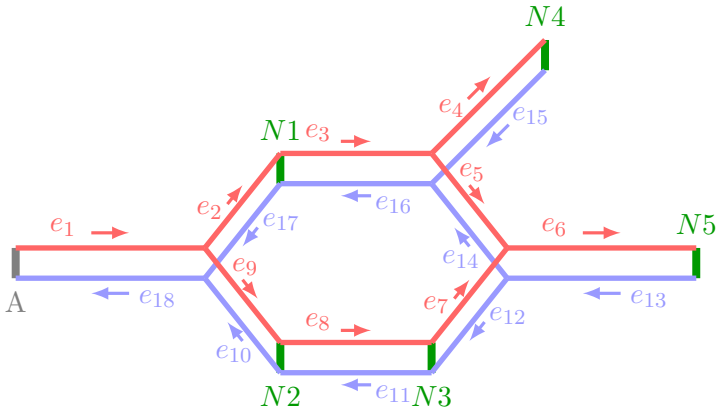

Figure 2.1: Graph of a district heating network

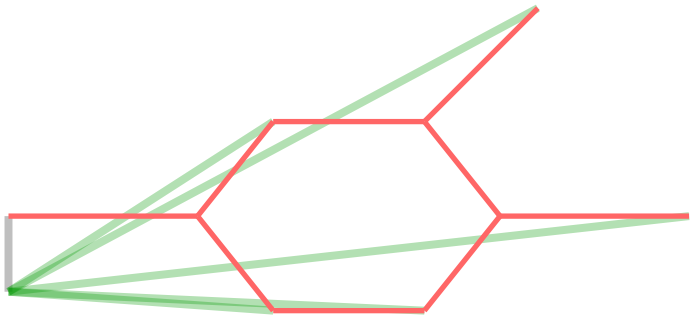

Figure 2.2: Graph of a district heating network with compressed return

2.2.2 Network Components

In this section, we introduce all components of a district heating network.
The components are represented by the edges of the graph, where every edge, depending on its type, adds a set of equations to the full system. The components are then coupled via suitable coupling conditions at the nodes.

Pipes

The most important components are the pipes. They are responsible for the transport of the hot water to all parts in the network and transporting back the cool water to be reheated again. The dynamics inside the pipes are modeled by the Euler equations, derived in Section 2.1. For the description of the transport medium water, we use the incompressible equations (2.9). The high pressure inside the pipes ensures that the water stays in the liquid phase for all temperatures within the operation range (up to $140°C$).

We now introduce the right hand side of the momentum equation. The two terms influencing the momentum by additional forces are friction and gravity. Friction results from viscous effects from boundary layer theory [46] and in the section averaged case can be modeled by the Darcy-Weisbach law

$$F_{fr} = \frac{\lambda}{2d} v|v|,$$

with diameter of the pipe d and friction coefficient λ. The friction factor depends on material properties of the pipe as well as viscosity and flow velocity of the medium. A popular model for λ is the so-called Colebrook-White equation [41], where λ is the solution of

$$\frac{1}{\sqrt{\lambda}} = -2\log_{10}\left(\frac{\nu}{3.7d} + \frac{2.51}{\mathrm{Re}\sqrt{\lambda}}\right).$$

Here, ν is the pipe roughness and the Reynolds number, depending on velocity and viscosity μ,

$$\mathrm{Re} = \frac{|v|\rho d}{\mu},$$

is a dimensionless indicator for the typical flow conditions. A Reynolds number below 2300 means laminar flow, while $\mathrm{Re} > 3000$ typically is a turbulent regime with a transition zone in between. District heating networks are operated in the turbulent regime in most cases. That further justifies the one dimensional considerations and the perfect mixing assumption, as no distinct layers can develop.

Furthermore we need a gravity term modeling the effects of different elevations in the network. The gravitational acceleration on the fluid has the form

$$F_g = g\partial_x b,$$

with gravitational constant g and the space derivative of the elevation b. Altogether, the momentum equation for the incompressible system reads

$$\partial_t v + \frac{1}{\rho}\partial_x p = \frac{\lambda}{2d} v|v| + g\partial_x b. \tag{2.14}$$

For the heating purposes, we are only interested in the internal energy part e of the total energy. This is why we drop the kinetic part in the energy equation. The thermal losses in a pipeline due to insufficient insulation are given by

$$q(e) = \frac{4k}{d}(T(e) - T_\infty(e)),$$

where k is the heat transmission coefficient and T_∞ the temperature of the ground or air outside the pipe. Note, that here we need a relation between temperature T and internal energy e. For water, experimental data can be used to approximate this relation by polynomials up to a desired order. We restrict ourselves to the first order approximation

$$T(e) = 1.94 + 220.54\ e.$$

The values are achieved by interpolations of data from [55]. Furthermore, we assume the density to be constant, i.e. the continuity equation becomes trivial.
Using the length L of the pipe as reference length x_r as in (2.10), we can transform the computational domain to be $\Omega = [0, 1]$.
Altogether, the incompressible system for water flow in a pipe is modeled by

$$\begin{aligned} \partial_x v &= 0 \\ \partial_t v + \frac{1}{\rho}\partial_x p &= \frac{\lambda}{2d} v|v| + g\partial_x b \qquad && (x,t) \in \Omega \times [0, t_{end}]. \\ \partial_t e + v\partial_x e &= -\frac{4k}{d}(T(e) - T_\infty(e)) \end{aligned}$$

When the equations are lifted to the full network, we equip the variables with superscripts indicating the corresponding edge.

Source

The source is located at the position of the power plant and responsible for the heating of the water. The source will act as the hydraulic driver of the network and produces the necessary pressure gradient accelerating the water in the pipes. Furthermore, it generates the thermal energy and transfers it to the water in the network. The exact dynamics of the power plant is not in the scope of this work, so we model it by a single pipe connecting return to flow part of the net. Rather than having PDEs on an edge, it provides the network with boundary values

$$p^s(0,t) = p_0(t), \quad p^s(1,t) = p_1(t), \quad e^s(1,t) = e_{in}(t).$$

In this work, we restrict ourselves to a single source in the network, but in general more sources are possible. In fact, there is current research on having each consumer acting as a source. The so called "prosumers" are able to drain from and feed energy to the network by the use of decentralized solar heating for example.

Consumers

The consumers extract the thermal energy from the water in heat exchangers. Besides the source, they are the only edges connecting the flow part with the return part of the network. Every consumer has a time dependent energy demand $Q(t)$ it wants to drain from the system. The heat exchangers will drop the incoming temperature to a contractually determined [2] return temperature T_{ret}. The flow velocity will be regulated to provide the exact power needed

$$Q(t) = v(t) \cdot (e(0,t) - e_{ret}). \tag{2.15}$$

As already mentioned above, the flow direction for the consumer edges has to be positive for all times. In real networks, this is achieved by a sufficiently large margin of the difference between the feed temperature e_{in} and the desired return temperature e_{ret}. Typical values are $e_{in} \in [80°C, 140°C]$, $e_{ret} = 60°C$. For analytical purposes, the consumer equation can be modified to

$$Q(t) = v(t) \cdot (\max\{e(0,t), e^c_{min}\} - e_{ret}),$$

with a fixed energy $e^c_{min} > e_{ret}$. This ensures a minimal energy difference and conserves the positivity of the velocity in any cases.

There is a large variety of different consumers in a district heating network, each having unique behavior. Since their real time consumption is not known in detail, consumers are grouped in different classes, e.g. one family houses, apartment houses and industrial consumers. Members of the same class share the same form function weighted with an individual annual consumption factor. The consumption of $c \in \mathcal{J}_C$ then has the following form [7]:

$$Q^c(t) = KW^c \cdot h(T_d, m(c)) \cdot \eta_{m(c)}(t, T_d).$$

The factor KW^c is the mean daily power drainage, $h(T_d, m(c))$ is a profile function depending on the outside temperature T_d and the consumer class $m(c)$ and finally $\eta_{m(c)}$ is an hourly distribution function of the daily consumption values. Note, that the outside temperature T_d is in general different from T_∞ in the energy transport equation, since the first one describes the air temperature and the latter the temperature of the ground. The shape of the consumption profiles is shown in Figure 2.3 for different outside temperatures T_d over the course of three days. The general shape of the three curves is similar, while the mean demand for temperatures of $0°C$ are about nine times higher, than at temperatures of $25°C$. The profiles show two distinct peaks of consumption, a sharp one in the morning and another, slightly smaller one in the afternoon. At nighttime, the consumption is the lowest, with about half the amount of the maximum.

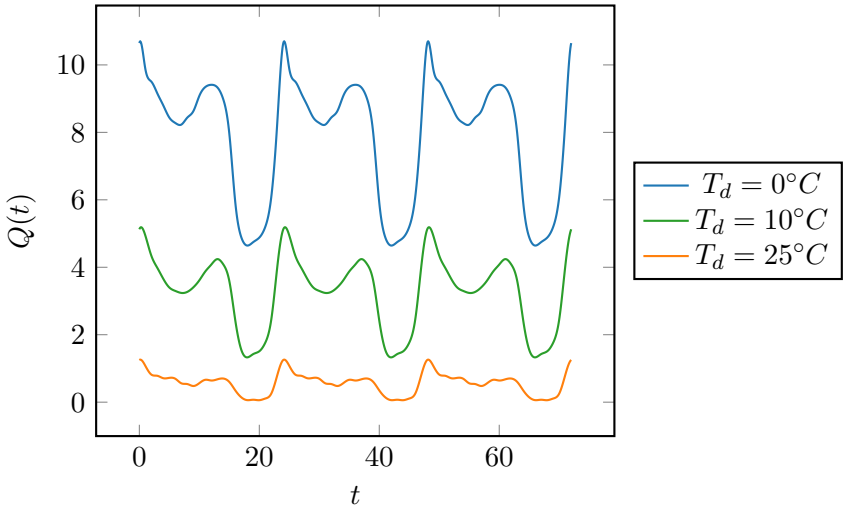

Figure 2.3: Consumption profiles for different outside temperatures

Nodes

The different components are coupled in nodes to form the network. We now formulate the coupling conditions that provide boundary values to the PDEs on the edges. In each node, the relevant quantities should be conserved. For the mass flow ρv this means the sum of all the flows on edges connected to the node must add up to zero

$$\sum_{J_i \in \mathcal{A}(V)} A^i \rho^i v^i(t) = 0, \qquad \forall V \in \mathcal{V},\ \forall t. \tag{2.16}$$

Here, $A^i = \pi(\frac{d^i}{2})^2$ is the cross sectional area of the pipe. This expression is straight forward, since the involved values are not space dependent and thus the edge orientation can be neglected.

For the coupling of p we assume the pipes to come together at a node in one point, meaning that they effectively share the same pressure value. So for a node V we have

$$p^V(t) = p^i(1,t) = p^j(0,t) \qquad \forall J_i \in \mathcal{J}_{in}(V),\ \forall J_j \in \mathcal{J}_{out}.$$

If we assume the vertical slope of each pipe to be constant, the only space dependent variable remaining in the momentum balance is the pressure, which means $\partial_x p$ has to be constant in space. We can integrate the momentum equation over the spatial domain $[0,1]$ leading to

$$\partial_t v + \frac{1}{\rho}\Delta p = \frac{\lambda}{2d} v|v| + g\Delta b.$$

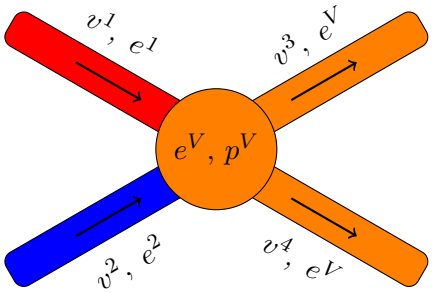

Figure 2.4: Perfect node mixing

We can see that only the pressure difference $\Delta p = p(0,t) - p(1,t)$ is needed, values in between can be linearly interpolated. It is not necessary to compute the full spatial resolution of p, instead we just compute the pressure for every node p^V.

In order to formulate the conservation of energy in the nodes, the flow orientation is important such that the energy is picked from the right side of the edge. A perfect mixing of the involved flows is assumed, this means that all outgoing edges share the same energy, the flow-weighted mean of all incoming energies. Note here, that flow directions may change over time and therefore the mixing dependencies are time dependent. For edge J_i we define the outflow boundary as

$$c^i(t) = \begin{cases} 1, & \text{if } v^i(t) \geq 0 \\ 0, & \text{if } v^i(t) < 0. \end{cases}$$

The inflow boundary is defined analogously and denoted by $\neg c^i(t)$.
The energy coupling with perfect mixing then reads

$$\sum_{J_i \in \mathcal{I}(V)} A^i |v^i(t)| e^i(c^i(t), t) - \sum_{J_i \in \mathcal{O}(V)} A^i |v^i(t)| e_V(t) = 0 \tag{2.17a}$$

$$e^i(\neg c^i(t), t) = e_V(t) \qquad \forall V \in \mathcal{V},\ \forall t, \tag{2.17b}$$

with the time dependent node energy e_V.

If the densities are considered constant, they can be eliminated from (2.16), effectively leading to a volume conservation formula. Since temperature differences in the node mixing can get relatively large, a more precise formulation includes the energy dependence of the water density. That way, the velocity coupling cannot be solved independently of the energy.

2.2.3 Full Network Model

Now we have all the necessary equations to formulate the full model on the network. It consists of an inflow boundary condition given by the source, outflow boundary

conditions implicitly given by the consumers and a set of (P)DEs on the pipelines. The spatial domain of every pipe has been scaled to the interval $[0, 1]$. Due to different pipe lengths, this leads to different scalings in each pipe that has to be compensated in the coupling. The rescaling is done by a modified cross-section $\tilde{A}^i = A^i v_r^i$ bringing all velocities to their corresponding range. The tildes are then dropped again for clarity This means we can formulate the whole system as a set of equations on $\Omega = [0, 1]$. We split the set of all edges into the subsets induced by the three edge categories source (s), consumer (c) and pipe (p)

$$\mathcal{J}_s \cup \mathcal{J}_c \cup \mathcal{J}_p = \mathcal{J}.$$

All relevant variables on the edges are collected in vectors $\mathbf{u} = (u_i)_i = (u^i)_{J_i \in \mathcal{J}} \in \mathbb{R}^{N_\mathcal{J}}$. Additionally, for a vector $\mathbf{u}$ created that way, let the corresponding diagonal matrix be $\mathbf{U} = \mathrm{diag}(\mathbf{u}) \in \mathbb{R}^{N_\mathcal{J} \times N_\mathcal{J}}$. We collect all the above defined models in one large system

$$\partial_x \mathbf{v} = 0 \tag{2.18a}$$

$$\mathbf{I}_{\mathcal{J}_p} \left(\partial_t \mathbf{v} + \frac{1}{\rho} \Delta \mathbf{p} \right) = \mathbf{I}_{\mathcal{J}_p} \left(\frac{1}{2} \mathbf{\Lambda} \mathbf{D}^{-1} \mathbf{V} |\mathbf{v}| + g \Delta \mathbf{b} \right) \tag{2.18b}$$

$$\mathbf{I}_{\mathcal{J}_p} \left(\partial_t \mathbf{e} + \mathbf{V} \partial_x \mathbf{e} \right) = \mathbf{I}_{\mathcal{J}_p} \left(-4 \mathbf{k} \mathbf{d}^{-1} (T(\mathbf{e}) - T_\infty(\mathbf{e})) \right) \tag{2.18c}$$

$$\mathbf{I}_{\mathcal{J}_c} \mathbf{V} \cdot (\mathbf{e}(0, \cdot) - e_{ret}) = \mathbf{Q} \tag{2.18d}$$

$$\mathbf{I}_{\mathcal{J}_s} \mathbf{e}(1, \cdot) = e_{in} \tag{2.18e}$$

$$\mathbf{I}_{\mathcal{J}_s} \mathbf{p} = p^s \tag{2.18f}$$

$$\mathbf{A}^I \mathbf{v} = 0 \tag{2.18g}$$

$$\left.\begin{aligned} \sum_{J_i \in \mathcal{I}(V)} A^i |v^i(t)| e^i(c^i(t), t) &= \sum_{J_i \in \mathcal{O}(V)} A^i |v^i(t)| e^V(t) \\ e^j(\neg c^j(t), t) &= e^V(t) \end{aligned}\right\} \forall V \in \mathcal{V}, \forall J_j \in \mathcal{O}(V), \tag{2.18h}$$

for $(t, x) \in [0, t_{end}] \times \Omega$. The first three equations (2.18a)-(2.18b) are the vectorized Euler equations on the pipes. The relevant entries are selected by the indicator matrices $\mathbf{I}_{\mathcal{J}_p} \in \mathbb{R}^{N_p \times N_\mathcal{J}}, \mathbf{I}_{\mathcal{J}_c} \in \mathbb{R}^{N_c \times N_\mathcal{J}}, \mathbf{I}_{\mathcal{J}_s} \in \mathbb{R}^{N_s \times N_\mathcal{J}}$. Without loss of generality we can assume the edge ordering to start with the pipes, followed by the consumers and the source at last. Composed, they build the identity matrix

$$\mathbf{I} = \begin{bmatrix} \mathbf{I}_{\mathcal{J}_p} \\ \mathbf{I}_{\mathcal{J}_c} \\ \mathbf{I}_{\mathcal{J}_s} \end{bmatrix}.$$

The pressure and height differences Δp and Δb can alternatively written as a linear form with the incidency matrix

$$\Delta \mathbf{p} = (\mathbf{A}^I)^T \mathbf{p}, \qquad \Delta \mathbf{b} = (\mathbf{A}^I)^T \mathbf{b}.$$

Equation (2.18d) is the vectorized version of (2.15), with the vector of consumptions $\mathbf{Q}$ and the consumer indicator matrix $\mathbf{I}_{\mathcal{J}_c}$. The boundary conditions at the source are

formulated in (2.18e) and (2.18f). In our setting, where the case of a single source is considered, those equations are scalar. The mass conservation is the linear equation (2.18g) with the assumption of constant density. Finally, the energy conservation is described by (2.18h). Here we stay with the nodal formulation for now as a more compact notation would involve more formal definitions.

2.3 Splitting

The full system (2.18) is a PDAE that becomes large once it is fully discretized. The special structure of the network and the system of equations allows for efficient solving techniques by decomposing the hydraulic part into linear and nonlinear elements. This motivates a splitting of the full equation set into two separate ones, the hydraulic equation and the thermal equation. Due to the fact that energy influences the fluid velocity only at the boundary, namely when reaching the consumers in (2.18d), the splitting is straight forward:
The hydraulic system can be solved for velocity and pressure, given an energy at the consumers. We introduce the modified demand $\tilde{\mathbf{Q}}(\mathbf{e}, t)$ with

$$\tilde{\mathbf{Q}}_i = \frac{Q^i(t)}{e^i(0,t) - e_{ret}},$$

for the consumers $J_i \in \mathcal{J}_c$. Then the decoupled hydraulic problem is:

$$\begin{aligned} \partial_x \mathbf{v} &= 0 \\ \mathbf{I}_{\mathcal{J}_p}\left(\partial_t \mathbf{v} + \frac{1}{\rho}(\mathbf{A}^I)^T \mathbf{p}\right) &= \mathbf{I}_{\mathcal{J}_p}\left(\frac{1}{2}\boldsymbol{\Lambda}\mathbf{D}^{-1}\mathbf{V}|\mathbf{v}| + g(\mathbf{A}^I)^T \mathbf{b}\right) \\ \mathbf{A}^I \mathbf{v} &= 0 \\ \mathbf{I}_{\mathcal{J}_c} \mathbf{v} &= \tilde{\mathbf{Q}}(\mathbf{e}, t) \\ \mathbf{I}_{\mathcal{J}_s} \mathbf{p} &= p^s, \end{aligned} \tag{2.19}$$

which only implicitly depends on the energy via the modified demands at the consumers. Systems of the form (2.19) are used in water supply networks [1],[74], where there is no temperature information needed and the consumers have pure volume flow demands as a function of time. Using the information of the network structure, such systems can be solved very efficiently. A decomposition of the system into a pure algebraic part and a set of ODEs of smaller dimension is shown in [44] and [36].
The general idea is the decomposition of the graph into a spanning tree and a set of circles. For tree-networks, the algebraic part

$$\begin{aligned} \mathbf{A}^I \mathbf{v} &= 0 \\ \mathbf{I}_{\mathcal{J}_c} \mathbf{V} &= \tilde{\mathbf{Q}}(\mathbf{e}, t) \end{aligned} \tag{2.20}$$

can be uniquely solved for v. The momentum equation only acts as a definition of the pressures in the network. When there exist circles in the network, we can make use

of Kirchhoffs voltage laws. Originally formulated for electrical circuits, the law can be transferred to hydraulic systems by replacing voltage with pressure. It states that the sum of all pressure drops along a closed circle have to sum up to zero.

$$\sum_{J_i \in C_i} \Delta p^i = 0 \qquad \forall C_i \in \mathcal{C}. \tag{2.21}$$

Since (2.20) is uniquely solvable for trees and for each circle, we can add an additional equation with one new velocity component of the form (2.21), the resulting system stays uniquely solvable. More details on the exact form of the new system are investigated in the analysis Chapter 3. We just want to emphasize here, that instead of solving the full nonlinear system of size $N_{\mathcal{J}}$, we solve a linear one of size $N_{\mathcal{V}} - 1$ and the nonlinear one just for the remaining $N_{\mathcal{C}}$. In general the number of circles in district heating networks is relatively small compared to the total number of edges, so this leads to a much simpler problem.

The remaining set of equations form the energy transport system

$$\mathbf{I}_{\mathcal{J}_p}\left(\partial_t \mathbf{e} + \mathbf{V}\partial_x \mathbf{e}\right) = \mathbf{I}_{\mathcal{J}_p}\left(-4\mathbf{k}\mathbf{d}^{-1}(T(\mathbf{e}) - T_\infty(\mathbf{e}))\right) \tag{2.22a}$$

$$\mathbf{I}_{\mathcal{J}_s}\mathbf{e}(1,\cdot) = e_{in} \tag{2.22b}$$

$$\left.\begin{aligned} \sum_{J_i \in \mathcal{I}(V)} A^i |v^i(t)| e^i(c^i(t),t) &= \sum_{J_i \in \mathcal{O}(V)} A^i |v^i(t)| e^V(t) \\ e^j(c^j_\neg(t),t) &= e^V(t) \end{aligned}\right\} \forall V \in \mathcal{V}, \forall J_j \in \mathcal{O}(V). \tag{2.22c}$$

The parts (2.22b) and (2.22c) are responsible for providing the boundary data for the PDE (2.22a). The latter is a system of pure advection equations with a source term. The main focus of this thesis lies on the numerical simulation of such networks of advection equations.

2.4 Conclusion

District heating networks are modeled by a set of Euler equations on a graph. The typical operation conditions force the water into its liquid phase, even for temperatures higher than $140°C$, allowing to use the incompressible equations. Those equations hold on all pipes of the network, where a suitable friction model and a cooling term model the interaction of the fluid with the pipe wall. Further components of the network are consumers and source, responsible for heat transfer onto the network and its drainage at the desired locations. With some basic graph notations, we can formulate the full network system in a very compact way. Hereby all relevant components are coupled by basic conservation properties and ideal mixing laws. By applying a splitting to the full system, separating hydraulic and thermal transport part, we can use efficient solving strategies exploiting their specific structures.

Chapter 3

Analysis

In this chapter, some analytic properties of the system are explored. Starting with some examples, we show scenarios where even for very smooth initial conditions the regularity of the solution of the energy advection is reduced. This might be surprising, since the underlying PDE is linear. The network structure of the full system however introduces a nonlinear behavior. With an adjustment of the node coupling equation, we regain stability of solutions in the BV sense. Finally, the wellposedness of the district heating model is shown in three steps. First the existence of solutions as well as their Lipschitz continuous dependence on the parameters for the hydraulic as well as the transport equation is shown. With those results we construct an operator in the specifically chosen solution space and show its contractive property. Application of Banachs fixed point theorem yields unique existence of solutions to our problem. These stability results emerged from joint work together with Raul Borsche[1], Mauro Garavello[2] and Elena Rossi[3]. The corresponding paper is already submitted [9].

3.1 Analytical examples

Before coming to the theoretical results some analytical examples are shown. We want to emphasize the impact of the network structure on solutions to the transport problem. In district heating networks, the flow direction of edges inside the network can change over time. In general, the sign of velocities is known and fixed in some parts of the network, for example at the power plant and on the consumer edges. If then the network graph is a tree, the positivity directly transfers to all the other edges. However, if there are circles in the graph, the velocity can change sign on the edge closing the circle, depending on the energy distribution and the consumer behaviour. This can already happen in the smallest possible example as shown in the following. Each change of flow direction might introduce a new contact discontinuity in the solution.

[1]Technische Universität Kaiserslautern, borsche@mathematik.uni-kl.de
[2]University of Milano Bicocca, mauro.garavello@unimib.it
[3]University of Modena and Reggio Emilia, elena.rossi13@unimore.it

3.1.1 Formation of Contact Discontinuities

Even for smooth initial data and velocities, contact discontinuities can form inside the network. This is due to the fact, that the sign of velocity can change. When flow direction changes in an edge J_i and the node temperature is not equal to the energy values inside the pipe, a discontinuous energy signal will travel through the pipe. This can happen in general in any network that contains a cycle. A minimal example is a triangle network, containing a source, 3 pipes and 2 consumers. Further details about this example can be found in [51]. The geometry of the triangle network is shown in Figure 3.1. We assume that all three edges have the same parameters as length, diameter and friction coefficient. Furthermore we assume perfect thermal insulation i.e. there is no heat loss and we ignore the acceleration term in the momentum equation. While those simplification allow for a simple example, where an analytic solution for the full system can be given, they do not change the general behavior of the system. Including all the mentioned effects would also lead to a solution with the desired properties but an analytic formulation would be much harder.

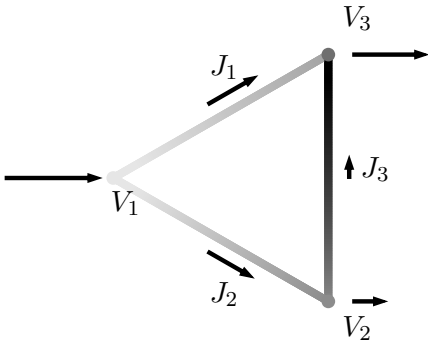

Figure 3.1: Triangle network

Analyzing the symmetry of the geometry, we observe that if the demands of the two consumers produce the same velocities, the pressure drops along the edges J_1 and J_2 are equal and thus the velocity on edge J_3 will be zero. If then one of the demands rises, v_3 will grow in the corresponding direction. With suitable energy configurations in the pipes, a contact discontinuity forms. The exact setting of an example for such a case is the following:
Let $t \in [-1, 1]$ and

$$\begin{aligned} v_1 &= 1 + \frac{t}{6} \\ v_2 &= 1 - \frac{t}{6} \\ e_b(t) &= 9 + \frac{t}{6}. \end{aligned}$$

Using the momentum equation, we can calculate the velocity of the connecting edge v_3 as

$$v_3|v_3| = v_1|v_1| - v_2|v_2| = (1+\frac{t}{6})|1+\frac{t}{6}| - (1-\frac{t}{6})|1-\frac{t}{6}| = \frac{4t}{6}$$
$$\rightarrow v_3 = 2sign(t)\sqrt{|\frac{t}{6}|}.$$

Knowing the velocities, the energies in the pipes can be calculated by the method of characteristics as

$$e_1(t,x) = e_b(t_1(x)) = 3 + 6\sqrt{(1+\frac{t}{6}) - \frac{x}{3}}$$
$$e_2(t,x) = e_b(t_2(x)) = 15 - 6\sqrt{(1+\frac{t}{6}) + \frac{x}{3}},$$

where t_1 and t_2 are the origin times at the left boundary of the characteristics of v_1 and v_2 respectively. The energy e_3 is getting its values, depending on the sign of v_3 from either edge 1 or edge 2. At $t = 0$ the sign of v_3 changes and the corresponding characteristic travels into the domain. This results in the piecewise defined signal

$$e_3(t,x) = \begin{cases} 15 - 6\sqrt{\left[1 - \left(|\frac{t}{6}|^{\frac{3}{2}} + \frac{x-1}{8}\right)^{\frac{2}{3}}\right]^2 + \frac{1}{3}}, & \text{if } t \geq 0 \text{ and } x \geq 1 - \left(\frac{2t}{3}\right)^{\frac{3}{2}} \\ 3 + 6\sqrt{\left[1 - \left(|\frac{t}{6}|^{\frac{3}{2}} + \frac{x}{8}\right)^{\frac{2}{3}}\right]^2 - \frac{1}{3}}, & \text{else.} \end{cases}$$

The corresponding values at the discontinuity are

$$e_3(0^+,1) = 15 - 6\sqrt{\frac{4}{3}} \approx 8.07$$
$$e_3(0^-,1) = 3 + 6\sqrt{\left(\frac{3}{4}\right)^2 - \frac{1}{3}} \approx 5.87.$$

This discontinuity then travels along the pipe. The resulting signal at $e(t,0)$ is shown in Figure 3.2.

3.1.2 Violation of BV-stability

In a second example, we want to show a rather artificial scenario. In theory it is possible that the sign of velocity changes infinitely many times in a finite time horizon. We will later show, that the velocity as solution of the hydraulic system will lie in the space of absolutely continuous functions. Additionally, when at a junction two pipes feed a

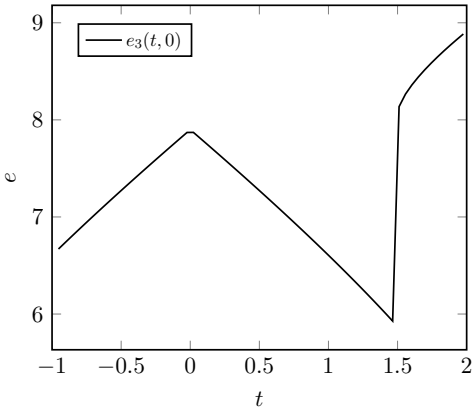

Figure 3.2: Temperature signal $e_3(0,t)$

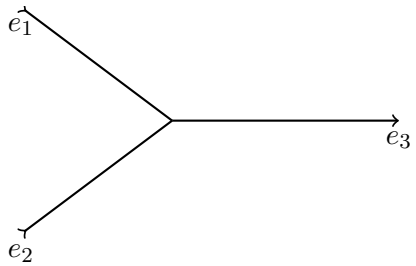

Figure 3.3: 3-way junction with two edges feeding a third one

third one (compare Figure 3.3), it is possible that the side from which the outgoing edge is fed changes infinitely many times in an interval. Consider the following setting:

$$\begin{aligned} e_1(0,x) &= 1 \\ e_2(0,x) &= 0 \\ e_3(0,x) &= 0 \end{aligned}$$

and

$$v_1 = \begin{cases} e^{-\frac{1}{(t_0-t)^2}} \sin^2(\frac{1}{(t_0-t)}), \text{ if } \frac{1}{(t_0-t)} \in [2k\pi, (2k+1)\pi], k \in \mathbb{N} \\ 0, \text{ else} \end{cases}$$

$$v_1 = \begin{cases} e^{-\frac{1}{(t_0-t)^2}} \sin^2(\frac{1}{(t_0-t)}), \text{ if } \frac{1}{(t_0-t)} \in [(2k+1)\pi, 2k\pi], k \in \mathbb{N} \\ 0, \text{ else} \end{cases}$$

$$v_3 = v_1 + v_2 = e^{-\frac{1}{(t_0-t)^2}} \sin^2(1/(t_0-t)).$$

Pipe J_3 will now draw alternately the values 1 and 0. Each switch raises the total variation of the solution by 1, so for $t = t_0$ the total variation is unbounded. A schematic illustration of the solution is shown in Figure 3.4. Note, that the drawn

values are not the real solution but one that more visibly displays the true behavior. The zeros do coincide the ones of the true solution.

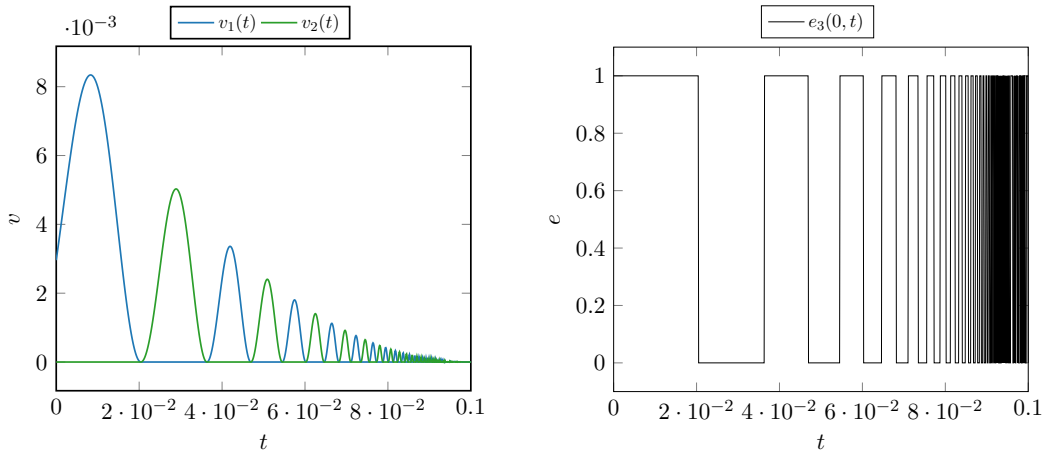

Figure 3.4: Schematic velocities and energy violating BV-stability

Scenarios like that will not play a role in real scenarios, as a fixed discretization grid limits the number of switches. For analytical considerations however, we need to somehow deal with those extreme cases. In order to do that, we introduce a small volume $\tilde{V}$ in the nodes. This adds some inertia to the node mixing energy and therefore smoothens the boundary data that edge three draws in above example. For every node V we introduce a new node energy variable e_V. The change of node energy within a small time step Δt can be expressed by

$$\begin{aligned} e_V(t) + \Delta e_V = \frac{1}{\tilde{V}_V} \sum_{J_i \in \mathcal{I}^t(V)} A^i |v^i(t)| e^i(t, b_i) \Delta t \\ + \frac{1}{\tilde{V}_V} \left(\tilde{V}_V - \sum_{J_i \in \mathcal{O}^t(V)} A^i \big| v^i(t) \big| \Delta t \right) e_V(t). \end{aligned}$$

As it can be seen in Figure 3.5, the change in node energy is the sum of all incoming volumes $A^i v^i$ minus the sum of all out flowing volumes weighted with their respective energies. Passing the limit $\Delta t \to 0$ we end up with an ODE describing the dynamics of the node energy e_V

$$\dot{e}_V = \frac{1}{\tilde{V}_V} \left(\sum_{J_i \in \mathcal{I}^t(V)} A^i |v^i(t)| e^i(t, c_i(t)) \quad - \sum_{J_i \in \mathcal{O}^t(V)} A^i \big| v^i(t) \big| e_V(t) \right). \tag{3.1}$$

This leads to higher regularity in the resulting solution and thus to a bounded total variation. With the choice $\tilde{V} = 10^{-5}$ the solution of above example is shown in Figure 3.6.

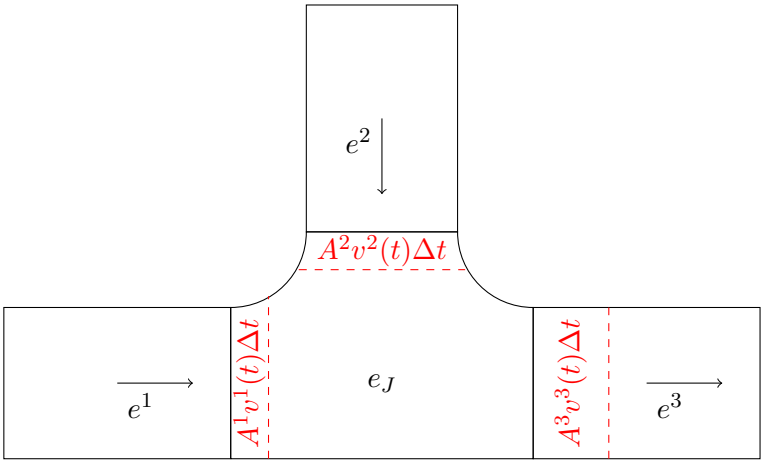

Figure 3.5: Schematic model of the node V with two incoming and one outgoing pipes in the derivation of the coupling condition (3.1)

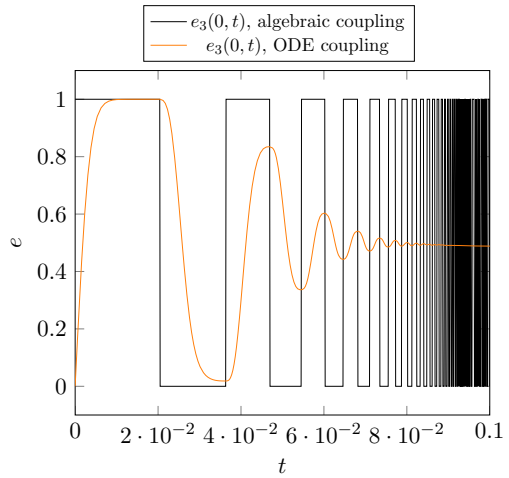

Figure 3.6: Node energy with new coupling formulation and $V = 10^{-5}$

3.1.3 Trace of the Solution

The important detail, why we are able to get stability estimates for the transport part is that in all coupling terms of the energy we consider the trace of the fluxes $v(t)e(t, c_i(t))$ instead of the trace $e(t, c_i(t))$. For the traces of solutions to

$$\begin{aligned} \partial_t e + v\partial_x e &= 0 \\ e(0,x) &= e_0(x), \end{aligned} \tag{3.2}$$

there exist no such estimates. Consider the following setting: Let $\Omega = [0,1]$ and

$$e_0(x) = \begin{cases} 1, & \text{if } x < 1, \\ 0, & \text{if } x = 1. \end{cases}$$

Consider two different velocities $v_1(t) = 0$ and $v_2(t) = \epsilon$. Then the traces of the corresponding solutions are $e_1(t,1) = 1$ and $e_2(t,1) = \begin{cases} 1, & t = 0 \\ 0, & t > 0 \end{cases}$ and therefore

$$\|e_1(\cdot,1) - e_2(\cdot,1)\|_{\mathbf{L^1}(0,t)} = t,$$

but

$$\|v_1 - v_2\|_{\mathbf{L^1}(0,t)} = \epsilon t$$

for all ϵ. So the difference in the traces is the same, even for arbitrarily small distance between v_1 and v_2. One possibility to get stability of traces is to prescribe a lower bound on the velocity.

Theorem 3.1.1 *Let $\Omega = [-\infty, 1]$, $v_1, v_2 \in [v_{min}, \infty]$ and $e_0 \in \mathbf{L^1}_{loc} \cap \mathbf{BV}(-\infty, 1)$, then for the two solutions to* (3.2)

$$\|e_1(\cdot,1) - e_2(\cdot,1)\|_{\mathbf{L^1}(0,t)} \leq Lt|v_1 - v_2|.$$

Proof. By solution of characteristics and using Lemma 3.6.3

$$\begin{aligned} \|e_1(\cdot,1) - e_2(\cdot,1)\|_{\mathbf{L^1}(0,t)} &= \int_0^t |e_0(1 - v_1\tau) - e_0(1 - v_2\tau)| \\ &\leq \frac{TV(e_0)}{v_{min}} t|v_1 - v_2|. \end{aligned}$$

□

This result can aswell be extended to non-smooth time dependent velocities. Many stability properties for velocity fields with low regularity have been shown in [23]. Those results however require a positive lower bound on the velocity, similar to the one above. As we have seen in the first example, flow direction changes cannot be avoided in district heating networks.

In (3.1), the traces never appear alone, but always multiplied with the corresponding velocity. This fact will save our results and enables the necessary stability estimates in Section 3.4. Above example would then read

$$\begin{aligned} \|v_1 e_1(\cdot,1) - v_2 e_2(\cdot,1)\|_{\mathbf{L^1}(0,t)} &= \|v_2 e_2(\cdot,1)\|_{\mathbf{L^1}(0,t)} \\ &\leq \epsilon t = \|v_1 - v_2\|_{\mathbf{L^1}(0,t)}, \end{aligned}$$

so we can in fact bound the difference of fluxes by the difference in velocities in that specific case. We will show in the following, that a stability estimate of that form holds for arbitrary time varying velocities in $\mathbf{L^1}$ without prerequisites on its sign. This will be the main tool in proving uniqueness of the coupled system.

3.2 Wellposedness Results

We finally come to the main results of this chapter, where the wellposedness of the coupled system is shown. We start by showing the existence of solutions to the hydraulic system. This system is very similar to those for water supply networks and we can use similar decomposition techniques bringing the system in a form where an explicit solution can be constructed.

After that, the required stability results for the transport equation are shown. In the main theorem of this part, the Lipschitz continuous dependence of solutions to the advection equation on non-smooth velocities is shown. Furthermore, in a second part a similar L-continuous dependence of the fluxes at the boundaries is proven. This is essential, as the pipes in the DH network couple over the fluxes and therefore we can extend these results to the whole network.

Combining those two results enables the construction of a fixed point iteration. The Lipschitz continuity gives rise to a contraction property and with Banach's fixed point theorem the wellposedness of the coupled problem is shown.

3.3 Hydrodynamic System

In the following, we analyze the wellposedness of hydrodynamic equations on the network. This system has already been studied in different applications, most notably in the context of water supply networks. First of all, the system and all necessary requirements are stated. After that, we show that there is a reformulation of the system allowing for direct construction of a solution. This procedure was already presented in [36] and we just emphasize the differences to their system of equations. In the end, we show that the solution fulfills a stability condition with respect to the input parameters, most notably the energies at the consumer sites.

Having a closer look at the consumer equation (2.15), the smoothness of the energies at the consumers directly transfers to the smoothness of the vector of velocities. For the stability estimate and some technical details later in the coupled setting, additional smoothness of the solution v to the hydrodynamic system is needed. We modify the algebraic consumer equation and instead use the ODE formulation

$$\partial_t v^k = \frac{1}{\alpha}\left(Q_k(t) - A^k v^k (e_{V_k} - e(T_{out})\right), \tag{3.3}$$

with relaxation parameter $\alpha > 0$. Similar to the modified node mixing in (3.1), this adds regularity to the system. In particular the solution v to (3.3) is continuous even if demand and energy are just $\mathbf{L^1}$ functions. Note that each consumer is connected to a node in the network and therefore using the new coupling defined in (3.1) the corresponding node energy has to be considered.

We recall the hydraulic system from (2.19):

$$\begin{aligned} \partial_x \mathbf{v} &= 0 \\ \mathbf{I}_{\mathcal{J}_p}\left(\partial_t \mathbf{v} + \frac{1}{\rho}(\mathbf{A}^I)^T \mathbf{p}\right) &= \mathbf{I}_{\mathcal{J}_p}\left(\frac{1}{2}\mathbf{\Lambda}\mathbf{D}^{-1}\mathbf{V}|\mathbf{v}| + g(\mathbf{A}^I)^T \mathbf{b}\right) \\ \mathbf{I}_{\mathcal{J}_c}\partial_t \mathbf{v} &= \tilde{\mathbf{Q}}(\mathbf{v}, \mathbf{e}, t) \\ \mathbf{I}_{\mathcal{J}_s}\mathbf{p} &= p^s \\ \mathbf{A}^I \mathbf{v} &= 0 \end{aligned} \tag{3.4}$$

and incorporate the new formulation of the coupling condition (3.1) and consumer equation (3.3) by

$$\tilde{Q}_i = \frac{1}{\alpha}\left(Q^i(t) - v^i\left(\max\{e_{V_{i_1}}(t), e_{min}\} - e_{ret}\right)\right)$$

for each consumer $J_i = (V_{i_1}, V_{i_2})$.

Definition 3.3.1 *Fix $T > 0$ and, for every $J_k \in \mathcal{J}_C$, functions Q_k that are in $\mathbf{L^1}\left([0, +\infty); \left[\overline{Q}_{min}, \overline{Q}_{max}\right]\right)$. Assume that, for every $J_k = (V_{k_1}, V_{k_2}) \in \mathcal{J}_C$, the function $t \mapsto e_{V_{k_1}}(t)$ is in $\mathbf{L^1}([0,T];\mathbb{R})$. A couple $(p, v) = \left(\left(p^1, \dots, p^{N_\mathcal{V}}\right), \left(v^1, \dots, v^{N_\mathcal{J}}\right)\right)$ is a hydraulic solution to (3.4) on the time interval $[0,T]$ if the following conditions are satisfied.*

1. *For every $i \in \{1, \dots, N\}$, the functions p^i, and v^i satisfy the regularity assumptions:*
   - *(a) $p^i \in \mathbf{AC}([0,T];\mathbb{R})$*
   - *(b) $v^i \in \mathbf{AC}([0,T];\mathbb{R})$.*
2. *For every $J_i = (V_{i_1}, V_{i_2}) \in \mathcal{J}_P$, p and v satisfy*
$$p^{i_2} = p^{i_1} - \rho\left(\frac{\lambda^i}{2d^i} v^i |v^i| + \partial_t v^i(t) + g\left(b^{i_1} - b^{i_2}\right)\right).$$
3. *For every $J_i = (V_{i_1}, V_{i_2}) \in \mathcal{J}_C$, for every $t \in [0,T]$,*
$$v^i(t) = v^i(0) + \int_0^t \frac{1}{\alpha}\left(Q^i(s) - v^i(s)\left(\max\{e_{V_{i_1}}(s), e_{min}\} - e_{ret}\right)\right) \mathrm{d}s\,.$$
4. *The boundary condition $p^1(t, a_1) = p^{1,b}(t)$ holds.*
5. *For every junction $V \in \mathcal{V}$,*
$$\sum_{J_h \in \mathcal{A}(V)} A^h v^h(t) = 0.$$

For the proof of the next lemma we use several properties of the topology matrices defined in Section 2.2.1. We just state them here, a proof can be found in [36].

1. $\begin{pmatrix} \mathbf{A}_t \\ \mathbf{A}_{PC} \end{pmatrix}$ and $\mathbf{A}_t\mathbf{A}_r$ are nonsingular
2. $\mathbf{A}_{PC}\mathbf{A}_r = 0$

Lemma 3.3.2 *Let $\mathcal{G} = (\mathcal{V}, \mathcal{J})$ be a connected graph with at least one pressure node. Then the system*

$$\begin{aligned} \dot{v}_0 &= \tilde{\boldsymbol{Q}}(v_0, \boldsymbol{e}, t) \\ v_1 &= \left(\boldsymbol{A}^I \boldsymbol{A}_t^T\right)^{-1} \left(\boldsymbol{A}^I \boldsymbol{I}_{\mathcal{J}_C}\right) v_0 \\ \dot{v}_2 &= \left(\boldsymbol{A}_{PC} \boldsymbol{A}_{PC}^T\right)^{-1} \boldsymbol{A}_{PC} \left(r + \boldsymbol{f}\left(\boldsymbol{A}_t^T v_1 + \boldsymbol{A}_{PC}^T v_2 - \boldsymbol{A}_t^T \dot{v}_1\right)\right) \\ \boldsymbol{p} &= \rho \left(\boldsymbol{A}_t (\boldsymbol{A}_r)^T\right)^{-1} \boldsymbol{A}_t \left(r + \boldsymbol{f}\left(\boldsymbol{A}_t^T v_1 + \boldsymbol{A}_{PC}^T v_2\right) - \left(\boldsymbol{A}_t^T \dot{v}_1 + \boldsymbol{A}_{PC}^T \dot{v}_2\right)\right), \end{aligned} \tag{3.5}$$

is equivalent to (3.4).

Proof. A detailed proof can be found in [36]. We restrict ourselves to the main ideas of the proof and highlight the differences of our system to theirs.

Using the incidence matrices defined in Section 2.2.1 we perform the coordinate transformation

$$v = \begin{pmatrix} \mathbf{I}_{\mathcal{J}_C} & \mathbf{A}_t^T & \mathbf{A}_{PC}^T \end{pmatrix} \begin{pmatrix} v_0 \\ v_1 \\ v_2 \end{pmatrix} = \mathbf{I}_{\mathcal{J}_C} v_0 + \mathbf{A}_t^T v_1 + \mathbf{A}_{PC}^T v_2. \tag{3.6}$$

Due to the third equation of (3.4), the v_0 component is independent of the others and follows the ODE

$$\dot{v}_0 = \tilde{\mathbf{Q}}(v_0, \mathbf{e}, t).$$

Then the mass conservation reads

$$\begin{aligned} \mathbf{A}^I \mathbf{v} &= \mathbf{A}^I \begin{pmatrix} \mathbf{I}_{\mathcal{J}_C} & \mathbf{A}_t^T & \mathbf{A}_{PC}^T \end{pmatrix} \begin{pmatrix} v_0 \\ v_1 \\ v_2 \end{pmatrix} \\ &= \mathbf{A}^I \mathbf{I}_{\mathcal{J}_C} v_0 + \mathbf{A}^I \mathbf{A}_t^T v_1 \\ &= 0 \end{aligned}$$

due to $\mathbf{A}^I \mathbf{A}_{PC}^T = 0$ and therefore

$$v_1 = \left(\mathbf{A}^I \mathbf{A}_t^T\right)^{-1} \left(\mathbf{A}^I \mathbf{I}_{\mathcal{J}_C}\right) v_0.$$

We multiply the second equation of (3.4) with $\begin{pmatrix} \mathbf{A}_t \\ \mathbf{A}_{PC} \end{pmatrix}$ and with (3.6) we get

$$
\begin{aligned}
&\mathbf{I}_{\mathcal{J}_p}\left(\partial_t \mathbf{v} + \frac{1}{\rho}(\mathbf{A}^I)^T\mathbf{p}\right) = \mathbf{I}_{\mathcal{J}_p}\left(\frac{1}{2}\mathbf{\Lambda}\mathbf{D}^{-1}\mathbf{V}|\mathbf{v}| + g(\mathbf{A}^I)^T\mathbf{b}\right)\\
\Leftrightarrow &\begin{cases}
\mathbf{A}_t\left(\mathbf{A}_t^T\dot{v}_1 + \mathbf{A}_{PC}^T\dot{v}_2\right) + \mathbf{A}_t\frac{1}{\rho}(\mathbf{A}^I)^T\mathbf{p} = \mathbf{A}_t\mathbf{f}\left(\mathbf{A}_t^T v_1 + \mathbf{A}_{PC}^T v_2\right) + \mathbf{A}_t g(\mathbf{A}^I)^T\mathbf{b}\\
\mathbf{A}_{PC}\left(\mathbf{A}_t^T\dot{v}_1 + \mathbf{A}_{PC}^T\dot{v}_2\right) + \mathbf{A}_{PC}\frac{1}{\rho}(\mathbf{A}^I)^T\mathbf{p} = \mathbf{A}_{PC}\mathbf{f}\left(\mathbf{A}_t^T v_1 + \mathbf{A}_{PC}^T v_2\right) + \mathbf{A}_{PC} g(\mathbf{A}^I)^T\mathbf{b}
\end{cases}\\
\Leftrightarrow &\begin{cases}
\frac{1}{\rho}\mathbf{A}_t(\mathbf{A}^I)^T\mathbf{p} = \mathbf{A}_t\left(\mathbf{f}\left(\mathbf{A}_t^T v_1 + \mathbf{A}_{PC}^T v_2\right) - \left(\mathbf{A}_t^T\dot{v}_1 + \mathbf{A}_{PC}^T\dot{v}_2\right) + g(\mathbf{A}^I)^T\mathbf{b}\right)\\
\mathbf{A}_{PC}\mathbf{A}_{PC}^T\dot{v}_2 = \mathbf{A}_{PC}\left(-\frac{1}{\rho}(\mathbf{A}^I)^T\mathbf{p} - \mathbf{A}_t^T\dot{v}_1 + \mathbf{f}\left(\mathbf{A}_t^T v_1 + \mathbf{A}_{PC}^T v_2\right) + g(\mathbf{A}^I)^T\mathbf{b}\right)
\end{cases}\\
\Leftrightarrow &\begin{cases}
\frac{1}{\rho}\mathbf{A}_t(\mathbf{A}_r)^T\mathbf{p} = \mathbf{A}_t\left(\mathbf{f}\left(\mathbf{A}_t^T v_1 + \mathbf{A}_{PC}^T v_2\right) - \left(\mathbf{A}_t^T\dot{v}_1 + \mathbf{A}_{PC}^T\dot{v}_2\right) + g(\mathbf{A}^I)^T\mathbf{b} - \frac{1}{\rho}\mathbf{A}_r^p p^s\right)\\
\mathbf{A}_{PC}\mathbf{A}_{PC}^T\dot{v}_2 = \mathbf{A}_{PC}\left(-\frac{1}{\rho}\mathbf{A}_r^p p^s - \mathbf{A}_t^T\dot{v}_1 + \mathbf{f}\left(\mathbf{A}_t^T v_1 + \mathbf{A}_{PC}^T v_2\right) + g(\mathbf{A}^I)^T\mathbf{b}\right)
\end{cases}
\end{aligned}
$$

with $\mathbf{f}(\mathbf{v}) = \frac{1}{2}\mathbf{\Lambda}\mathbf{D}^{-1}\mathbf{V}|\mathbf{v}|$. Finally, with the abbreviation $r = g(\mathbf{A}^I)^T\mathbf{b} - \frac{1}{\rho}\mathbf{A}_r^p p^s$ we get for the full system

$$
\begin{aligned}
\dot{v}_0 &= \tilde{\mathbf{Q}}(v_0, \mathbf{e}, t)\\
v_1 &= \left(\mathbf{A}^I\mathbf{A}_t^T\right)^{-1}\left(\mathbf{A}^I\mathbf{I}_{\mathcal{J}_C}\right)v_0\\
\dot{v}_2 &= \left(\mathbf{A}_{PC}\mathbf{A}_{PC}^T\right)^{-1}\mathbf{A}_{PC}\left(r - \mathbf{A}_t^T\dot{v}_1 + \mathbf{f}\left(\mathbf{A}_t^T v_1 + \mathbf{A}_{PC}^T v_2\right)\right)\\
\mathbf{p} &= \rho\left(\mathbf{A}_t(\mathbf{A}_r)^T\right)^{-1}\mathbf{A}_t\left(r + \mathbf{f}\left(\mathbf{A}_t^T v_1 + \mathbf{A}_{PC}^T v_2\right) - \left(\mathbf{A}_t^T\dot{v}_1 + \mathbf{A}_{PC}^T\dot{v}_2\right)\right),
\end{aligned}
$$

which concludes the proof. □

For this system we formulate the following theorem:

Theorem 3.3.3 *Let $T > 0$, $p^s \in \mathbf{C}([0,T],\mathbb{R})$ and $\boldsymbol{Q} \in \mathbf{L^1}([0,T],\mathbb{R}^{N_C})$ such that $\|\boldsymbol{Q}(t)\|_\infty < Q_{max}$ for almost every t. Then for each node energy function $\boldsymbol{e}_V$ that lies in $\mathbf{L^1}([0,T],\mathbb{R}^{N_\mathcal{V}})$ with $\|\boldsymbol{e}_V(t)\|_\infty < e_{max}$ for almost every t, there exists a unique solution $(\boldsymbol{v},\boldsymbol{p}) \in \mathbf{AC}([0,T],\mathbb{R}^{N_\mathcal{J}}) \times \mathbf{L^1}([0,T],\mathbb{R}_+^{N_\mathcal{V}})$ to (3.4) in the sense of Definition 3.3.1 with the following properties:*

1. *$\boldsymbol{v}$ and $\boldsymbol{p}$ are bounded, i.e. $\|\boldsymbol{v}(t)\|_\infty \le v_{max}$ and $\|\boldsymbol{p}(t)\|_\infty \le p_{max}$ for almost every t.*

2. *The solution depends L-continuously on the parameters, i.e. there exists a positive constant $L_v > 0$ such that, for every two sets of initial conditions $\bar{p}^{i,o}$, $\tilde{p}^{i,o}$, $\bar{v}^{i,o}$, and $\tilde{v}^{i,o}$ ($i \in \mathcal{J}$), two boundary data $\bar{p}^s$, $\tilde{p}^s$, two sets of power demands $\bar{\boldsymbol{Q}}$, $\tilde{\boldsymbol{Q}}$, and two sets of node energy functions $\bar{\boldsymbol{e}_\mathcal{V}}$, $\tilde{\boldsymbol{e}_\mathcal{V}}$ in $\mathbf{L^1}\left(0,T;\mathbb{R}^{N_\mathcal{V}}\right)$, the corresponding solutions $(\bar{\boldsymbol{p}},\bar{\boldsymbol{v}})$ and $(\tilde{\boldsymbol{p}},\tilde{\boldsymbol{v}})$ satisfy for a.e. $t \in [0,T]$*

$$
\sum_{J_i\in\mathcal{J}}\left\|\bar{v}^i - \tilde{v}^i\right\|_{\mathbf{C^0}([0,t])} \le L_v\left[\sum_{V\in\mathcal{V}}\|\bar{e}_V - \tilde{e}_V\|_{\mathbf{L^1}([0,t])} + \sum_{J_k\in\mathcal{J}_C}\left\|\bar{Q}_k - \tilde{Q}_k\right\|_{\mathbf{L^1}(0,T)}\right]. \tag{3.7}
$$

Proof. Due to Lemma 3.3.2 we can show the results for the equivalent system (3.5). Under above assumptions, the function $\tilde{\mathbf{Q}}(v_0, \mathbf{e}, t)$ with
$\tilde{Q}_i = \frac{1}{\alpha}\left(Q^i(t) - v^i\left(\max\{e_{V_{i_1}}(t), e_{min}\} - e_{ret}\right)\right)$ is globally Lipschitz continuous in v_0 as well as measurable in t. This means the ODE

$$\dot{v}_0 = \tilde{\mathbf{Q}}(v_0, \mathbf{e}, t)$$

fulfills the requirements of the Caratheodory existence theorem and there exists a unique solution $v_0(t) \in \mathbf{AC}([0, t^\star], \mathbb{R}^{N_Q})$. Furthermore using Gronwalls inequality we get an a priori bound on each component of the solution.

$$|v_0^i| \leq a(t) e^{\int_0^t b(\tau) d\tau} =: v_{0,max}(t),$$

where $a(t) = \int_0^t \frac{Q^i(\tau)}{\alpha} d\tau$ and $b(t) = \int_0^t \frac{\max\{e_{V_{i_1}}(\tau), e_{min}\} - e_{ret}}{\alpha} d\tau$.

After that, the component $v_1(t) = \left(\mathbf{A}^I \mathbf{A}_t^T\right)^{-1} \left(\mathbf{A}^I \mathbf{I}_{\mathcal{J}_C}\right) v_0$ is just a linear function of v_0 and consequently also unique with the same regularity as v_0.

Similarly to v_0, v_2 is again the solution of a Caratheodory type ODE with locally Lipschitz right hand side and there exists a locally unique solution. It remains to show, that this solution can be extended to the full time interval $[0, T]$. For this proof we refer to the proof of theorem 5.3 in [36], where they show that there exist $k, c > 0$ such that

$$\|v_2(t)\|_\infty \leq k e^{ct}$$

and with this a priori estimate the solution can be uniquely extended to the full time interval. Note that the constants k, c depend on $v_{0,max}$ in our case. With those estimates we can recover a bound on $\mathbf{v}$ by

$$\|\mathbf{v}\|_\infty \leq (1 + \|M_1\|_\infty) \|v_0\|_\infty + \|M_2\|_\infty \|v_2\|_\infty =: v_{max},$$

with $M_1 = \mathbf{A}_t^T \left(\mathbf{A}^I \mathbf{A}_t^T\right)^{-1} \left(\mathbf{A}^I \mathbf{I}_{\mathcal{J}_C}\right)$ and $M_2 = \mathbf{A}_{PC}^T$. The boundedness of $\mathbf{p}$ then is trivial as all the parameters it depends on are bounded.

The stability estimates directly follow from basic ODE theory, e.g. [22].
The L-continuity of the right hand side of the ODE on the parameters $\mathbf{Q}(t)$ and $e_{\mathcal{V}}(t)$ directly transfers to the solution. The proof is similar to showing continuous dependence on initial values. In fact, if the right hand side of the ODE is continuous, continuous dependence on parameters is equivalent to continuous dependence on initial values. □

Remark 3.3.4 *In Lemma 3.3.2 we showed, that the hydraulic system has a triangular structure in terms of (v_0, v_1, v_2, p). In fact, the velocities can be computed completely independent of p. The pressure is only of interest, when the system is used within an optimization, where e.g. box constraints on p are prescribed to avoid bursting pipes.*

This is why from now on, we drop the pressure from all solution notations and stability estimates. Due to the form of (3.5)*, especially the similarity of the third and fourth equation, all estimates that can be derived for v will hold in a similar form also for p. For clarity and simplicity of notations, when talking about the solution of the hydraulics, we refer to it by just v.*

3.4 Energy Network

In this section we study the wellposedness of the transport problem on networks. Similar existence results have been shown in [62] by using the notion of renormalized solutions. In our case, a constructive proof by the method of characteristics is possible. Using the modified node coupling, we are able to get powerful additional stability results. The main ingredient in the proof is Lemma 3.4.2, where we show unique existence of solutions to one dimensional advection problems for $\mathbf{L^1}$-velocities together with a Lipschitz continuous dependence of solutions on v. This result is then extended to the network setting by incorporating the coupling ODE.
We recall the PDE part of system (2.22) from Section 2.3

$$\begin{aligned} I_{\mathcal{J}_p}\left(\partial_t \mathbf{e} + \mathbf{V}\partial_x \mathbf{e}\right) &= I_{\mathcal{J}_p}\mathbf{g(e)} \\ I_{\mathcal{J}_s}\mathbf{e}(1,\cdot) &= e_{in} \end{aligned} \tag{3.8}$$

and add the new node coupling formulation

$$\begin{aligned} \dot{e}_V &= \frac{1}{V_V}\left(\sum_{J_i\in\mathcal{I}^t(V)} A^i|v^i(t)|e^i(t,c_i(t)) \quad - \sum_{J_i\in\mathcal{O}^t(V)} A^i\big|v^i(t)\big|e_V(t)\right) \\ e_V(0) &= \frac{1}{\sum_{J_i\in\mathrm{out}^t(V)} A^i|v^i(0)|}\left(\sum_{J_i\in\mathrm{out}^t(V)} A^i\big|v^i(0)\big|e^i(0,c_i(t))\right) \end{aligned} \tag{3.9}$$

$$e^i(t,\neg c_i(t)) = e_V(t), \qquad \forall J_j \in \mathcal{O}(V) \tag{3.10}$$

for all nodes $V \in \mathcal{V}$.

Definition 3.4.1 *Fix $T > 0$ and assume $e^{i,0} \in \mathbf{L^1}([a,b];\mathbb{R})$, $\forall J_i \in \mathcal{J}$. Furthermore let g be Lipschitz continuous. Assume that, for every $J_i \in \mathcal{J}$, the function $t \mapsto v^i(t)$ is in $\mathbf{L^1}((0,+\infty);\mathbb{R})$. A function $e = (e^{\mathcal{J}}, e_{\mathcal{V}}) = \left(\left(e^1,\dots,e^{N_{\mathcal{J}}}\right), \left(e_{J_1},\dots,e_{J_{N_{\mathcal{V}}}}\right)\right)$ is called an energy solution to* (3.8),(3.9),(3.10) *on the time interval $[0,T]$ if the following conditions are satisfied.*

1. *For every $i \in \{1,\dots,N\}$, the function $e^i \in \mathbf{C^0}\left([0,T];\mathbf{L^1}\left([a_i,b_i];\mathbb{R}\right)\right)$ satisfies the regularity assumption and, for every $t \in [0,T]$, $e^i(t)$ has finite total variation.*

2. *For every $V \in \mathcal{V}$, the function $e_V \in \mathbf{C^0}\left([0,T];\mathbb{R}\right)$ has finite total variation.*

3. *For every* $i \in \{1, \dots, N\}$, e^i *is a broad solution to*

$$\begin{aligned} \partial_t e^i + v \partial_x e^i &= g(e^i) \\ e^i(0, x) &= e^{i,0}(x). \end{aligned}$$

4. *For every junction* $V \in \mathcal{V}$, *for a.e.* $t \in [0, T]$ *and for all* $J \in \mathcal{A}(V)$

$$e^i(t, \neg c_i(t)) = e_V(t),$$

where e_V *is the Caratheodory solution to* (3.9).

In the following we will show that there exists a unique solution in the sense of Definition 3.4.1 and furthermore this solution depends Lipschitz continuously on the velocity vector $\mathbf{v}$. This is done in several steps. The main ingredient is to show unique existence and L-continuous dependence on v for an isolated pipe with given boundary data, i.e. for a one dimensional domain $\Omega = [a, b]$

$$\begin{aligned} \partial_t e + v \partial_x e = g(e) \qquad & x \in (a, b),\ t > 0 \\ e(t, a) = e_L(t) & \\ e(t, b) = e_R(t) & \\ e(0, x) = \bar{e}(x). & \end{aligned} \tag{3.11}$$

The important part here is that we show the stability property not only for the solution e itself, but also for its flux $v^+ e(t, 1)$ and $v^- e(t, 0)$ over the boundary. The latter is essential due to the dependence of the node coupling ODE on the fluxes.
In a second part, we show that the node energies conserve the desired stability property. The connection of this result with the boundary data for the problem on one segment gives rise to a Gronwall estimate for the full network case.

Lemma 3.4.2 ***(Stability of solutions on segments)***
Fix $a, b \in \mathbb{R}$, *with* $a < b$, $v_1, v_2, e_L^1, e_L^2, e_R^1, e_R^2 \in \mathbf{L^1}(\mathbb{R}^+)$ *and* $\bar{e}_1, \bar{e}_2 \in \mathbf{L^1}(a, b)$, *all with bounded total variation. Further let* $g \colon \mathbb{R} \to \mathbb{R}$ *be a Lipschitz continiuous function with* $|g(y_1) - g(y_2)| \leq G|y_1 - y_2|$. *We consider* e_1 *and* e_2 *the solutions to the following IBVP problems:*

$$P_1 \begin{cases} \partial_t e_1 + v_1 \partial_x e_1 = g(e_1) \qquad & x \in (a, b),\ t > 0 \\ e_1(t, a) = e_L^1(t) & \\ e_1(t, b) = e_R^1(t) & \\ e_1(0, x) = \bar{e}_1(x) & \end{cases} \tag{3.12}$$

and

$$P_2 \begin{cases} \partial_t e_2 + v_2 \partial_x e_2 = g(e_2) \qquad & x \in (a, b),\ t > 0 \\ e_2(t, a) = e_L^2(t) & \\ e_2(t, b) = e_R^2(t) & \\ e_2(0, x) = \bar{e}_2(x)\ . & \end{cases} \tag{3.13}$$

Then for a.e. t the following stability inequality holds:

$$\begin{aligned}\|e_1(t) - e_2(t)\|_{\mathbf{L^1}(a,b)} \leq & K_v \|v_1 - v_2\|_{\mathbf{L^1}(0,t)} + K_I \|\bar{e}_1 - \bar{e}_2\|_{\mathbf{L^1}(a,b)} \\ & + K_L \left\|e_L^1 - e_L^2\right\|_{\mathbf{L^1}(0,t)} + K_R \left\|e_R^1 - e_R^2\right\|_{\mathbf{L^1}(0,t)},\end{aligned} \tag{3.14}$$

where the Ks depend on the total variations and $\mathbf{L}^\infty$-norms of $\bar{e}_1$, $\bar{e}_2$, e_L, e_R and on G and t.

Furthermore, if $|v_1|, |v_2| \leq v_{max}$ and $t \leq \frac{b-a}{v_{max}}$, for the fluxes at the boundaries we have

$$\begin{aligned}\left\|v_1^- e_1(\cdot, a) - v_2^- e_2(\cdot, a)\right\|_{\mathbf{L^1}(0,t)} \leq & K_{v,a} \|v_1 - v_2\|_{\mathbf{L^1}(0,t)} + K_{I,a} \|\bar{e}_1 - \bar{e}_2\|_{\mathbf{L^1}(a,b)} \\ & + K_{L,a} \left\|e_L^1 - e_L^2\right\|_{\mathbf{L^1}(0,t)},\end{aligned} \tag{3.15}$$

$$\begin{aligned}\left\|v_1^+ e_1(\cdot, b) - v_2^+ e_2(\cdot, b)\right\|_{\mathbf{L^1}(0,t)} \leq & K_{v,b} \|v_1 - v_2\|_{\mathbf{L^1}(0,t)} + K_{I,b} \|\bar{e}_1 - \bar{e}_2\|_{\mathbf{L^1}(a,b)} \\ & + K_{R,b} \left\|e_R^1 - e_R^2\right\|_{\mathbf{L^1}(0,t)}.\end{aligned} \tag{3.16}$$

A sequential consideration with (3.14) *can remove the bound on t.*

The proof of this lemma is given in Section 3.6.1.
We now come to the main theorem of this section.

Theorem 3.4.3 *Fix $T > 0$. Then system* (3.8)-(3.10) *admits a unique solution, in the sense of Definition 3.4.1.*

Moreover the following stability estimate holds: There exists a positive constant $L > 0$ such that, for every two sets of initial conditions $\bar{e}^{i,o}$, $\tilde{e}^{i,o}$, ($J_i \in \mathcal{J}$) and two sets of velocity functions $\bar{v}$, $\tilde{v}$ fulfilling the requirements in Definition 3.4.1 the corresponding solutions $\bar{e} = (\bar{e}^{\mathcal{J}}, \bar{e}_{\mathcal{V}})$ and $\tilde{e} = (\tilde{e}^{\mathcal{J}}, \tilde{e}_{\mathcal{V}})$ satisfy for a.e. $t \in [0, T]$

$$\begin{aligned}&\sum_{V \in \mathcal{V}} \|\bar{e}_V - \tilde{e}_V\|_{\mathbf{C^0}(0,t)} + \sum_{J_i \in \mathcal{J}} \left\|\bar{e}^i(t) - \tilde{e}^i(t)\right\|_{\mathbf{L^1}(J_i)} \\ \leq & L \sum_{J_i \in \mathcal{J}} \left(\left\|\bar{v}^i - \tilde{v}^i\right\|_{\mathbf{L^1}(0,t)} + \left\|\bar{e}^{i,o} - \tilde{e}^{i,o}\right\|_{\mathbf{L^1}(I_i)} \right).\end{aligned} \tag{3.17}$$

We define by $\mathcal{T}_e$ the operator producing a solution according to Theorem 3.4.3.

Proof. Existence of solutions to similar transport problems on networks have already been proven, e.g. in [62]. The main difference is the new ODE coupling formulation we use compared to the algebraic one in their case. We focus on the stability estimate that to our knowledge has not been shown before. Nevertheless the main ingredient we use in this proof, Lemma 3.4.2, also contains the construction of a solution. The proof of the stability estimate consists of two steps. In the first one, we decompose the network problem into shorter time intervals. Those intervals are chosen in such a way that information does not travel from one node to another. Effectively the network problem then decouples into a sequence of localized problems on segments.

In the second step, the interconnection between the node energies and the energy fluxes at the boundaries are resolved. The right hand side of the node coupling ODE depends on the fluxes of solutions at the boundary. Whenever the flow velocity is negative, that node energy travels back into the domain and introduces a self dependence. By applying Gronwalls inequality we get a bound on e_V of the desired form.

Both of those steps depend on a stability condition for solutions and fluxes of solutions to the advection equation. The proof here is lengthy and since the result is rather general we have formulated it in the separate Lemma 3.4.2.

Step 1: Network decoupling
We want to prove the L-continuous dependence of the solution e as well as the node energies e_V on the velocity v:

$$\sum_{J_i\in\mathcal{J}} \left\|e^i(t,\cdot)-\tilde{e}^i(t,\cdot)\right\|_{\mathbf{L^1}(a,b)} \leq L_1 \sum_{J_k\in\mathcal{J}} \left(\left\|v^k-\tilde{v}^k\right\|_{\mathbf{L^1}(0,t)} + \left\|\bar{e}^{k,0}-\tilde{e}^{k,0}\right\|_{\mathbf{L^1}(a,b)}\right) \tag{3.18}$$

$$\sum_{V\in\mathcal{V}} \left\|e_V-\tilde{e}_V\right\|_{\mathbf{C^0}(0,t)} \leq L_2 \sum_{J_k\in\mathcal{J}} \left(\left\|v^k-\tilde{v}^k\right\|_{\mathbf{L^1}(0,t)} + \left\|\bar{e}^{k,0}-\tilde{e}^{k,0}\right\|_{\mathbf{L^1}(a,b)}\right) \tag{3.19}$$

For each individual edge $J_i \in \mathcal{J}$ we can apply Lemma 3.4.2 and get

$$\begin{aligned}\sum_{J_i\in\mathcal{J}} \left\|e^i(t,\cdot)-\tilde{e}^i(t,\cdot)\right\|_{\mathbf{L^1}(a,b)} \leq \sum_{J_k\in\mathcal{J}} \Bigg[& K_v^k \left\|v^k-\tilde{v}^k\right\|_{\mathbf{L^1}(0,t)} + K_I^k \left\|\bar{e}^{i,0}-\tilde{e}^{i,0}\right\|_{\mathbf{L^1}(a,b)} \\ & + K_L^k \left\|e_L^k-\tilde{e}_L^k\right\|_{\mathbf{L^1}(0,t)} + K_R^k \left\|e_R^k-\tilde{e}_R^k\right\|_{\mathbf{L^1}(0,t)}\Bigg],\end{aligned} \tag{3.20}$$

where the left and right boundary values e_L, e_R are exactly the node energies e_V due to (3.10). Let $deg(\mathcal{G})$ be the largest node degree in the network, then each node energy provides the boundary data to at most $deg(\mathcal{G})$ edges. Thus (3.20) gets

$$\begin{aligned}\sum_{J_i\in\mathcal{J}} \left\|e^i(t,\cdot)-\tilde{e}^i(t,\cdot)\right\|_{\mathbf{L^1}(a,b)} \leq & K_v \sum_{J_k\in\mathcal{J}} \left\|v^k-\tilde{v}^k\right\|_{\mathbf{L^1}(0,t)} + K_I \sum_{J_k\in\mathcal{J}} \left\|\bar{e}^{i,0}-\tilde{e}^{i,0}\right\|_{\mathbf{L^1}(a,b)} \\ & + K_V \sum_{V\in\mathcal{V}} \left\|e_V-\tilde{e}_V\right\|_{\mathbf{L^1}(0,t)} \\ \leq & K_v \sum_{J_k\in\mathcal{J}} \left\|v^k-\tilde{v}^k\right\|_{\mathbf{L^1}(0,t)} + K_I \sum_{J_k\in\mathcal{J}} \left\|\bar{e}^{i,0}-\tilde{e}^{i,0}\right\|_{\mathbf{L^1}(a,b)} \\ & + K_V t \sum_{V\in\mathcal{V}} \left\|e_V-\tilde{e}_V\right\|_{C^0(0,t)},\end{aligned} \tag{3.21}$$

with $K_v = \max_{J_k \in \mathcal{J}} K_v^k$, $K_I = \max_{J_k \in \mathcal{J}} K_I^k$ and $K_V = deg(\mathcal{G})(\max_{J_k \in \mathcal{J}} K_L + \max_{J_k \in \mathcal{J}} K_R)$.
Consequently, (3.18) follows directly from (3.19).

We now show (3.19) for one node $V \in \mathcal{V}$. Define the intermediate times $0 = t_0 < t_1 < \cdots < t_n = T$ with $\Delta t_i = (t_i - t_{i-1})$ such that $\max_{i=1,\dots,n} \Delta t_i \leq \frac{b-a}{v_{max}}$. Then in each interval $[t_{i-1}, t_i]$ the information coming from other nodes does not reach e_V. Assume

$$\begin{aligned} \|e_V - \tilde{e}_V\|_{\mathbf{C^0}(t_{i-1},t_i)} \leq A_V^i \sum_{J_k \in \mathcal{J}} \left\| v^k - \tilde{v}^k \right\|_{\mathbf{L^1}(t_{i-1},t_i)} \\ + B_V^i \sum_{J_k \in \mathcal{J}} \left\| e^k(t_{i-1},\cdot) - \tilde{e}^k(t_{i-1},\cdot) \right\|_{\mathbf{L^1}(a,b)} \\ + |e_V(t_{i-1}) - e_{\tilde{V}}(t_{i-1})| \end{aligned} \tag{3.22}$$

holds. We denote the maximum over those constants by $A = \max_{i,V} A_V^i$, $B = \max_{i,V} B_V^i$. Then by iterative insertion we get

$$\begin{aligned} \sum_{V \in \mathcal{V}} \|e_V - \tilde{e}_V\|_{\mathbf{C^0}(0,t)} &= \sum_{V \in \mathcal{V}} \sup_{i=0,\dots,n} \|e_V - \tilde{e}_V\|_{\mathbf{C^0}(t_{i-1},t_i)} \\ \text{(insert (3.22))} &\leq \sum_{V \in \mathcal{V}} \sup_{i=0,\dots,n} \Bigg(A \sum_{J_k \in \mathcal{J}} \left\| v^k - \tilde{v}^k \right\|_{\mathbf{L^1}(t_{i-1},t_i)} + B \sum_{J_k \in \mathcal{J}} \left\| e^k(t_{i-1},\cdot) - \tilde{e}^k(t_{i-1},\cdot) \right\|_{\mathbf{L^1}(a,b)} \\ &\qquad + |e_V(t_{i-1}) - e_{\tilde{V}}(t_{i-1})| \Bigg) \\ \text{(rearrange)} &\leq N_{\mathcal{V}} A \sum_{J_k \in \mathcal{J}} \left\| v^k - \tilde{v}^k \right\|_{\mathbf{L^1}(0,t)} + N_{\mathcal{V}} B \sup_{i=0,\dots,n-1} \sum_{J_k \in \mathcal{J}} \left\| e^k(t_i,\cdot) - \tilde{e}^k(t_i,\cdot) \right\|_{\mathbf{L^1}(a,b)} \\ &\qquad + \sum_{V \in \mathcal{V}} \|e_V - \tilde{e}_V\|_{\mathbf{C^0}(0,t_{n-1})} \\ \text{(insert (3.21)...)} &\leq N_{\mathcal{V}} A \sum_{J_k \in \mathcal{J}} \left\| v^k - \tilde{v}^k \right\|_{\mathbf{L^1}(0,t)} \\ &\qquad + N_{\mathcal{V}} B \sup_{i=0,\dots,n-1} \Bigg(K_v \sum_{J_k \in \mathcal{J}} \left\| v^k - \tilde{v}^k \right\|_{\mathbf{L^1}(0,t_i)} + K_V t \sum_{V \in \mathcal{V}} \|e_V - \tilde{e}_V\|_{\mathbf{C^0}(0,t_{n-1})} \\ &\qquad\qquad + K_I \sum_{J_k \in \mathcal{J}} \left\| e^k(t_{i-1},\cdot) - \tilde{e}^k(t_{i-1},\cdot) \right\|_{\mathbf{L^1}(a,b)} \Bigg) \\ &\qquad + \sum_{V \in \mathcal{V}} \|e_V - \tilde{e}_V\|_{\mathbf{C^0}(0,t_{n-1})} \\ \text{(rearrange)} &\leq N_{\mathcal{V}}(A + BK_v) \sum_{J_k \in \mathcal{J}} \left\| v^k - \tilde{v}^k \right\|_{\mathbf{L^1}(0,t)} \end{aligned}$$

$$+ (1 + N_{\mathcal{V}} B K_V t) \sum_{V \in \mathcal{V}} \|e_V - \tilde{e}_V\|_{\mathbf{C^0}(0,t_{n-1})}$$
$$+ N_{\mathcal{V}} B K_I \sup_{i=0,\dots,n-2} \sum_{J_k \in \mathcal{J}} \left\| e^k(t_i,\cdot) - \tilde{e}^k(t_i,\cdot) \right\|_{\mathbf{L^1}(a,b)}$$

We can insert (3.21) until the initial condition is reached. In each iteration, the resulting terms are multiplied by the constant K_I coming from the initial condition estimate of the previous step.

$$\begin{aligned}
{\scriptstyle (\dots\text{until } t_0)} \leq & N_{\mathcal{V}}(A + BK_v \frac{1-K_I^{n-1}}{1-K_I}) \sum_{J_k \in \mathcal{J}} \left\| v^k - \tilde{v}^k \right\|_{\mathbf{L^1}(0,t)} \\
& + N_{\mathcal{V}} B K_V t \sum_{i=0}^{n-1} \left((K_I)^i \sum_{V \in \mathcal{V}} \|e_V - \tilde{e}_V\|_{\mathbf{C^0}(0,t_{n-1-i})} \right) + \sum_{V \in \mathcal{V}} \|e_V - \tilde{e}_V\|_{\mathbf{C^0}(0,t_{n-1})} \\
& + N_{\mathcal{V}} B K_I^{n-1} \sum_{J_k \in \mathcal{J}} \left\| e^k(t_0,\cdot) - \tilde{e}^k(t_0,\cdot) \right\|_{\mathbf{L^1}(a,b)}
\end{aligned}$$

and finally with $\tilde{K}_I = \max(K_I, 1)$

$$\begin{aligned}
{\scriptstyle (\text{rearrange})} \leq & N_{\mathcal{V}}(A + BK_v \frac{1-K_I^{n-1}}{1-K_I}) \sum_{J_k \in \mathcal{J}} \left\| v^k - \tilde{v}^k \right\|_{\mathbf{L^1}(0,t)} \\
& + N_{\mathcal{V}} B K_I^{n-1} \sum_{J_k \in \mathcal{J}} \left\| e^{k,0} - \tilde{e}^{k,0} \right\|_{\mathbf{L^1}(a,b)} \\
& + (1 + N_{\mathcal{V}} B K_V \tilde{K}_I^{n-1} t) \sum_{i=0}^{n-1} \sum_{V \in \mathcal{V}} \|e_V - \tilde{e}_V\|_{\mathbf{C^0}(0,t_{n-1-i})} \,.
\end{aligned}$$

This is a recursive formula for the node energies $a(i) = \|e_V - \tilde{e}_V\|_{\mathbf{C^0}(0,t_i)}$ of the form

$$a(i) \leq c_1 + c_2 \sum_{l=0}^{i-1} a(l), \; a(1) = c_1,$$

with $c_1 = c_1^1 \sum_{J_k \in \mathcal{J}} \left\| v^k - \tilde{v}^k \right\|_{\mathbf{L^1}(0,t)} + c_1^2 \sum_{J_k \in \mathcal{J}} \left\| e^{k,0} - \tilde{e}^{k,0} \right\|_{\mathbf{L^1}(a,b)}$, where the coefficients are $c_1^1 = N_{\mathcal{V}} \left(A + BK_v \frac{1-K_I^{n-1}}{1-K_I} \right), c_1^2 = N_{\mathcal{V}} B K_I^{n-1}$ and $c_2 = (1 + N_{\mathcal{V}} B K_V \tilde{K}_I^{n-1} t)$. This recursion has the explicit formula

$$a(i) \leq c_1 \left(1 + c_2\right)^{(i-1)},$$

which leads to the explicit estimate of (3.4)

$$\begin{aligned}\sum_{V\in\mathcal{V}} \|e_V - \tilde{e}_V\|_{\mathbf{C^0}(0,t)} \leq (1+c_2)^{(n-1)} \Bigg(& c_1^1 \sum_{J_k\in\mathcal{J}} \left\|v^k - \tilde{v}^k\right\|_{\mathbf{L^1}(0,t)} \\ & + c_1^2 \sum_{J_k\in\mathcal{J}} \left\|e^{k,0} - \tilde{e}^{k,0}\right\|_{\mathbf{L^1}(a,b)} \Bigg) \\ \leq L \sum_{J_k\in\mathcal{J}} & \left(\left\|v^k - \tilde{v}^k\right\|_{\mathbf{L^1}(0,t)} + \left\|e^{k,0} - \tilde{e}^{k,0}\right\|_{\mathbf{L^1}(a,b)} \right).\end{aligned}$$

Thus the result (3.17) follows directly from (3.22), which we show in the second step.

Step 2: Stability of node values
From classical ODE stability we know for the junction that the solution of (3.9) fulfills

$$|e_V(t) - \tilde{e}_V(t)| \leq \hat{K} \left\|\hat{e} - \hat{\tilde{e}}\right\|_{L^1(0,t)} + K_{V,0}|e_V(0) - \tilde{e}_V(0)|,$$

where $\hat{e} = \sum_{J_i\in\mathcal{A}(V)} |v^i| e^i(\cdot, c_i(\cdot))$.

Each incoming flux can be estimated using (3.15) and (3.16) from Lemma 3.4.2 such that we have

$$\begin{aligned}\left\|\hat{e} - \hat{\tilde{e}}\right\|_{\mathbf{L^1}(0,t)} \leq K_V \|e_V - \tilde{e}_V\|_{\mathbf{L^1}(0,t)} & + K_v \sum_{J_i\in\mathcal{A}(V)} \left\|v^i - \tilde{v}^i\right\|_{\mathbf{L^1}(0,t)} \\ & + K_0 \sum_{J_i\in\mathcal{A}(V)} \left\|e^{i,0} - \tilde{e}^{i,0}\right\|_{\mathbf{L^1}(a,b)} .\end{aligned}$$

Inserting yields

$$\begin{aligned}|e_V(t) - \tilde{e}_V(t)| &\leq \alpha(t) + \hat{K}K_V \|e_V - \tilde{e}_V\|_{\mathbf{L^1}(0,t)} \\ &= \alpha(t) + \int_0^t \hat{K}K_V |e_V(\tau) - \tilde{e}_V(\tau)| \,\mathrm{d}\tau \,,\end{aligned}$$

where

$$\begin{aligned}\alpha(t) = \hat{K}K_v \sum_{J_i\in\mathcal{A}(V)} \left\|v^i - \tilde{v}^i\right\|_{\mathbf{L^1}(0,t)} & + \hat{K}K_0 \sum_{J_i\in\mathcal{A}(V)} \left\|e^{i,0} - \tilde{e}^{i,0}\right\|_{\mathbf{L^1}(a,b)} \\ & + K_{V,0}|e_V(0) - \tilde{e}_V(0)|.\end{aligned}$$

Now we apply Gronwall with $\alpha(t)$ monotone increasing and obtain

$$|e_V(t) - \tilde{e}_V(t)| \leq \alpha(t) e^{\hat{K}K_V t}$$

and due to monotonicity of the right hand side

$$\begin{aligned}\|e_V - \tilde{e}_V\|_{\mathbf{C^0}(0,t)} &\leq \alpha(t) e^{\hat{K} K_V t} \\ &\leq K \left(|e_V(0) - \tilde{e}_V(0)| + \sum_{J_i \in \mathcal{A}(V)} \left\| v^i - \tilde{v}^i \right\|_{\mathbf{L^1}(0,t)} \right. \\ &\quad \left. + \sum_{J_i \in \mathcal{A}(V)} \left\| e^{i,0} - \tilde{e}^{i,0} \right\|_{\mathbf{L^1}(a,b)} \right),\end{aligned}$$

where K does depend on t. By extending the sums on the right hand side to all edges instead of the adjacent ones we get that (3.22) does hold which concludes the proof. □

3.5 The Coupled System

In this part, we deal with the well posedness result for the complete system. The results of both separate parts are put together in order to construct solutions to the coupled problem

$$\begin{aligned}\partial_x \mathbf{v} &= 0 \\ \mathbf{I}_{\mathcal{J}_p} \left(\partial_t \mathbf{v} + \frac{1}{\rho} \Delta \mathbf{p} \right) &= \mathbf{I}_{\mathcal{J}_p} \mathbf{f}(\mathbf{v}) \\ \mathbf{I}_{\mathcal{J}_p} \left(\partial_t \mathbf{e} + \mathbf{V} \partial_x \mathbf{e} \right) &= \mathbf{I}_{\mathcal{J}_p} \mathbf{g}(\mathbf{e}) \\ \mathbf{I}_{\mathcal{J}_c} \mathbf{V} \cdot (\mathbf{e}(0,\cdot) - e_{ret}) &= \tilde{\mathbf{Q}}(\mathbf{v}, \mathbf{e}, t) \\ \mathbf{I}_{\mathcal{J}_s} \mathbf{e}(1,\cdot) &= e_{in} \\ \mathbf{I}_{\mathcal{J}_s} \mathbf{p} &= p^s \\ \mathbf{A}^I \mathbf{v} &= 0,\end{aligned} \tag{3.23}$$

together with the energy coupling

$$\begin{aligned}\dot{e}_V &= \frac{1}{V_V} \left(\sum_{J_i \in \mathcal{I}^t(V)} A^i |v^i(t)| e^i(t, c_i(t)) \quad - \sum_{J_i \in \mathcal{O}^t(V)} A^i \left| v^i(t) \right| e_V(t) \right) \\ e_V(0) &= \frac{1}{\sum_{J_i \in \mathrm{out}^t(V)} A^i |v^i(0)|} \left(\sum_{J_i \in \mathrm{out}^t(V)} A^i \left| v^i(0) \right| e^i(0, c_i(t)) \right) \\ e^i(t, \neg c_i(t)) &= e_V(t), \qquad \forall J_j \in \mathcal{O}(V)\end{aligned} \tag{3.24}$$

for all nodes $V \in \mathcal{V}$.

Definition 3.5.1 *Given $T > 0$, a couple (e, v) is a solution to (3.23)-(3.24) on the time interval $[0, T]$ if the following conditions are satisfied.*

1. $e = (e^{\mathcal{J}}, e_{\mathcal{J}}) = (\left(e^1, \dots, e^N\right), (e_{J_1}, \dots, e_{J_L}))$, *and* $v = \left(v^1, \dots, v^N\right)$.
2. *For the given node energies* $e_{\mathcal{J}}$, v *is a solution as defined in Definition 3.3.1.*
3. *For the given velocity* v, e *is a solution as defined in Definition 3.4.1.*

We finally state the main result of this chapter.

Theorem 3.5.2 *Fix* $T > 0$ *and, for every* $J_k \in \mathcal{J}_C$, *functions* $Q_k \in \mathbf{L^1}\left([0, +\infty); \left[\overline{Q}_{min}, \overline{Q}_{max}\right]\right)$. *Let* $e^{i,0} \in \mathbf{L^1}([a, b]; \mathbb{R})$, $\forall J_i \in \mathcal{J}$. *Furthermore let* f *be continuously differentiable and* g *be Lipschitz continuous. Then system* (3.23)-(3.24) *admits a unique solution, in the sense of Definition 3.5.1.*

Moreover, if $\bar{Q}_k$ *and* $\tilde{Q}_k$ *denote two different power demands for every* $J_k \in \mathcal{J}_C$, *then, denoting respectively by* $(\bar{e}, \bar{v})$ *and* $(\tilde{e}, \tilde{v})$ *the two solutions, the following stability estimate holds: there exists a positive constant* $L > 0$ *such that, for* $t \in [0, T]$,

$$\begin{aligned} &\sum_{J_i \in \mathcal{J}} \left[\left\| \bar{v}^i - \tilde{v}^i \right\|_{\mathbf{C^0}([0,t])} + \left\| \bar{e}^i(t) - \tilde{e}^i(t) \right\|_{\mathbf{L^1}(I_i)} \right] + \sum_{V \in \mathcal{V}} \left\| \bar{e}_V - \tilde{e}_V \right\|_{\mathbf{C^0}(0,t)} \\ &\leq L \sum_{J_k \in \mathcal{J}_C} \left\| \bar{Q}_k - \tilde{Q}_k \right\|_{\mathbf{L^1}(0,t)}. \end{aligned} \tag{3.25}$$

Proof. Define the Banach space

$$X_{e_{\mathcal{V}}} = \mathbf{C^0}\left((0, T); \mathbb{R}^L\right), \tag{3.26}$$

of node energy functions $e_{\mathcal{V}}$ endowed with the norm

$$\left\| e_{\mathcal{V}} \right\|_{X_{e_{\mathcal{V}}}} = \sup_{t \in [0,T], V \in \mathcal{V}} |e_V(t)|. \tag{3.27}$$

Define also the set

$$X_v = \left\{ v \in \mathbf{C^0}\left([0, T]; \mathbb{R}^N\right) : \sum_{J_j \in \mathcal{A}(V)} A^j v^j = 0 \text{ for every } V \in \mathcal{V} \right\}, \tag{3.28}$$

which is a closed subset of the Banach space $\mathbf{C^0}\left([0, T]; \mathbb{R}^N\right)$ endowed with the norm

$$\left\| v \right\|_{X_v} = \sup_{t \in [0,T]} |v(t)|. \tag{3.29}$$

Finally, define the set

$$X = X_{e_{\mathcal{V}}} \times X_v, \tag{3.30}$$

which is a complete metric space endowed with the distance associated to the norm

$$\left\| (e_{\mathcal{V}}, v) \right\|_X = \left\| e_{\mathcal{V}} \right\|_{X_{e_{\mathcal{V}}}} + \left\| v \right\|_{X_v}. \tag{3.31}$$

We consider the operator

$$\begin{array}{rccc} \mathcal{T}: & X & \longrightarrow & X \\ & (e_{\mathcal{V}}, v) & \longmapsto & (f, w)\,, \end{array} \tag{3.32}$$

where $w = \mathcal{T}_v\left(e_{\mathcal{V}}\right)$ is the solution of the hydrodynamics subsystem (see Theorem 3.3.3) and f is defined in the following way. First define $g = \left(g^{\mathcal{J}}, g_{\mathcal{V}}\right) = \mathcal{T}_e\left(v\right)$ as the solution of the energy subsystem (see Theorem 3.4.3). Then denote

$$f = \left(f_{V_1}, \ldots, f_{V_{N_{\mathcal{V}}}}\right) = \left(g_{V_1}, \ldots, g_{V_{N_{\mathcal{V}}}}\right) = g_{\mathcal{V}}$$

as the node energy functions.

$\mathcal{T}$ is well defined. This follows directly from the mentioned theorems. The node energies f lie in $X_{e_{\mathcal{V}}}$ and the velocities w lie in X_v.

$\mathcal{T}$ is a contraction. Fix two elements $(\bar{e}_{\mathcal{V}}, \bar{v}) \in X$, $(\tilde{e}_{\mathcal{V}}, \tilde{v}) \in X$ and denote

$$\left(\bar{f}, \bar{w}\right) = \mathcal{T}\left(\bar{e}_{\mathcal{V}}, \bar{v}\right) \in X, \qquad \left(\tilde{f}, \tilde{w}\right) = \mathcal{T}\left(\tilde{e}_{\mathcal{V}}, \tilde{v}\right) \in X.$$

We have that

$$\left\|\left(\bar{f}, \bar{w}\right) - \left(\tilde{f}, \tilde{w}\right)\right\|_X = \left\|\bar{f} - \tilde{f}\right\|_{X_{e_{\mathcal{V}}}} + \left\|\bar{w} - \tilde{w}\right\|_{X_v}.$$

By Theorem 3.3.3, we deduce that

$$\begin{aligned} \left\|\bar{w} - \tilde{w}\right\|_{X_v} &= \sup_{t\in[0,T]} \left|\bar{w}(t) - \tilde{w}(t)\right| \leq \sum_{J_i \in \mathcal{J}} \left\|\bar{w}^i - \tilde{w}^i\right\|_{\mathbf{C^0}([0,T])} \\ &\leq L \sum_{V\in\mathcal{V}} \left\|\bar{e}_V - \tilde{e}_V\right\|_{\mathbf{L^1}(0,T)} \leq LT \sum_{V\in\mathcal{V}} \left\|\bar{e}_V - \tilde{e}_V\right\|_{\mathbf{C^0}(0,T)} \\ &\leq LTN_{\mathcal{V}} \left\|\bar{e}_{\mathcal{V}} - \tilde{e}_{\mathcal{V}}\right\|_{X_{e_{\mathcal{V}}}}. \end{aligned}$$

By Theorem 3.4.3 we deduce that

$$\begin{aligned} \left\|\bar{f} - \tilde{f}\right\|_{X_{e_{\mathcal{V}}}} &= \sup_{t\in[0,T]} \left|\bar{f}(t) - \tilde{f}(t)\right| \leq \sum_{V\in\mathcal{V}} \left\|\bar{f}_V - \tilde{f}_V\right\|_{\mathbf{C^0}(0,T)} \\ &\leq L \sum_{J_i\in\mathcal{J}} \left\|\bar{v}^i - \tilde{v}^i\right\|_{\mathbf{L^1}(0,T)} \leq LT \sum_{J_i\in\mathcal{J}} \left\|\bar{v}^i - \tilde{v}^i\right\|_{\mathbf{C^0}([0,T])} \\ &\leq LTN_{\mathcal{J}} \left\|\bar{v} - \tilde{v}\right\|_{X_v}. \end{aligned}$$

Therefore

$$\left\|\left(\bar{f}, \bar{w}\right) - \left(\tilde{f}, \tilde{w}\right)\right\|_X \leq LT(N_{\mathcal{J}} + N_{\mathcal{V}}) \left\|\left(\bar{e}_{\mathcal{V}}, \bar{v}\right) - \left(\tilde{e}_{\mathcal{V}}, \tilde{v}\right)\right\|_X,$$

proving that $\mathcal{T}$ is a contraction, provided $T < \frac{1}{L(N_{\mathcal{J}}+N_{\mathcal{V}})}$. Then by Banach's fixed point theorem [5] there exists a unique fixed point $(e^\star_{\mathcal{V}}, v^\star)$ of $\mathcal{T}$ in X, i.e.

$$\mathcal{T}(e^\star_{\mathcal{V}}, v^\star) = (e^\star_{\mathcal{V}}, v^\star).$$

This fixed point is a solution in the sense of Definition 3.5.1 by the definition of $\mathcal{T}$. A sequential consideration similar to the proofs before allows the extension of this result to arbitrary times T.

Stability estimate. Let $\bar{Q}_k$ and $\tilde{Q}_k$ ($J_k \in \mathcal{J}_C$) denote two different power demands. Denote respectively with $(\bar{e}, \bar{p}, \bar{v})$ and $(\tilde{e}, \tilde{p}, \tilde{v})$ the corresponding solutions. By Theorem 3.3.3, we deduce that

$$\sum_{J_i\in\mathcal{J}} \left\|\bar{v}^i - \tilde{v}^i\right\|_{\mathbf{C^0}([0,t])} \leq L \left(\sum_{V\in\mathcal{V}} \|\bar{e}_V - \tilde{e}_V\|_{\mathbf{L^1}(0,t)} + \sum_{J_k\in\mathcal{J}_\mathcal{C}} \left\|\bar{Q}_k - \tilde{Q}_k\right\|_{\mathbf{L^1}(0,t)} \right), \quad (3.33)$$

and by Theorem 3.4.3 we deduce that

$$\sum_{V\in\mathcal{V}} \|\bar{e}_V - \tilde{e}_V\|_{\mathbf{C^0}(0,t)} + \sum_{J_i\in\mathcal{J}} \left\|\bar{e}^i(t) - \tilde{e}^i(t)\right\|_{\mathbf{L^1}(I_i)} \leq L \sum_{J_i\in\mathcal{J}} \left\|\bar{v}^i - \tilde{v}^i\right\|_{\mathbf{L^1}(0,t)}. \quad (3.34)$$

Using (3.21), the spatial $\mathbf{L^1}$-norm can be expressed in terms of the other two $\mathbf{C^0}$-norms and we get

$$\begin{aligned}
&\sum_{J_i\in\mathcal{J}} \left(\left\|\bar{v}^i - \tilde{v}^i\right\|_{\mathbf{C^0}([0,t])} + \left\|\bar{e}^i(t) - \tilde{e}^i(t)\right\|_{\mathbf{L^1}(I_i)} \right) + \sum_{V\in\mathcal{V}} \|\bar{e}_V - \tilde{e}_V\|_{\mathbf{C^0}(0,t)} \\
&\leq (Kt+1) \left(\sum_{J_i\in\mathcal{J}} \left\|\bar{v}^i - \tilde{v}^i\right\|_{\mathbf{C^0}([0,t])} + \sum_{V\in\mathcal{V}} \|\bar{e}_V - \tilde{e}_V\|_{\mathbf{C^0}(0,t)} \right),
\end{aligned}$$

in particular we have the estimates of (3.7) and (3.19) point wise for all t

$$\begin{aligned}
&\sum_{J_i\in\mathcal{J}} \left|\bar{v}^i(t) - \tilde{v}^i(t)\right| + \sum_{V\in\mathcal{V}} |\bar{e}_V(t) - \tilde{e}_V(t)| \\
&\leq L \left(\sum_{V\in\mathcal{V}} \|\bar{e}_V - \tilde{e}_V\|_{\mathbf{L^1}(0,t)} + \sum_{J_i\in\mathcal{J}} \left\|\bar{v}^i - \tilde{v}^i\right\|_{\mathbf{L^1}(0,t)} + \sum_{J_k\in\mathcal{J}_\mathcal{C}} \left\|\bar{Q}_k - \tilde{Q}_k\right\|_{\mathbf{L^1}(0,t)} \right).
\end{aligned}$$

We now can apply Gronwall and get

$$\begin{aligned}
&\sum_{J_i\in\mathcal{J}} \left|\bar{v}^i(t) - \tilde{v}^i(t)\right| + \sum_{V\in\mathcal{V}} |\bar{e}_V(t) - \tilde{e}_V(t)| \\
&\leq e^{Lt} \sum_{J_k\in\mathcal{J}_\mathcal{C}} \left\|\bar{Q}_k - \tilde{Q}_k\right\|_{\mathbf{L^1}(0,t)}.
\end{aligned}$$

Monotonicity of the right hand side immediately implies

$$\begin{aligned}
&\sum_{J_i\in\mathcal{J}} \left(\left\|\bar{v}^i - \tilde{v}^i\right\|_{\mathbf{C^0}([0,t])} + \left\|\bar{e}^i(t) - \tilde{e}^i(t)\right\|_{\mathbf{L^1}(I_i)} \right) + \sum_{V\in\mathcal{V}} \|\bar{e}_V - \tilde{e}_V\|_{\mathbf{C^0}(0,t)} \\
&\leq (1+Kt) \left(\sum_{J_i\in\mathcal{J}} \left\|\bar{v}^i - \tilde{v}^i\right\|_{\mathbf{C^0}(0,t)} + \sum_{V\in\mathcal{V}} \|\bar{e}_V - \tilde{e}_V\|_{\mathbf{C^0}(0,t)} \right) \\
&\leq (1+Kt)e^{Lt} \sum_{J_k\in\mathcal{J}_\mathcal{C}} \left\|\bar{Q}_k - \tilde{Q}_k\right\|_{\mathbf{L^1}(0,t)},
\end{aligned}$$

which concludes the proof. □

3.6 Technical Details

We end this chapter with some technical details concerning Lemma 3.4.2. We give the proof in the end of this section. It relies on some additional properties that are stated and proven first.

First of all, for many estimates in the proof of Lemma 3.4.2 we need an upper bound on the energies involved.

Lemma 3.6.1 ***(Boundedness of energy)***
Fix $T < 0$ and $a, b \in \mathbb{R}$, with $a < b$. Let $v, e_L, e_R \in \mathbf{L^1}([0,T],\mathbb{R})$ and $\bar{e} \in \mathbf{L^1}([a,b],\mathbb{R})$ all with bounded total variation. Further let $g\colon \mathbb{R} \to \mathbb{R}$ be a Lipschitz continiuous function with $|g(y_1) - g(y_2)| \leq G|y_1 - y_2|$.
Then for the solution e to (3.11) we have

$$e(t,x) \leq e_{max}$$

for all $t \leq T$ and almost every x.

Proof. First consider a single pipe and the corresponding transport equation

$$e_t + ve_x = g(e),$$

with $e(0,x) = e_0(x)$ and $e(t,a) = e_L(t), e(t,b) = e_R(t)$, where $e_0, e_L, e_R \in L^1$ are essentially bounded. Denote the essential supremum by

$$e_{max,b} = \max_{f\in\{e_0,e_L,e_R\}} \{\text{ess sup } f\},$$

then we know from basic ODE theory that

$$e(t,x) \leq \Theta(t) e_{max,b} \tag{3.35}$$

for almost every x, where $\Theta(t) = e^{Gt}$. For a node V, the node energy depends on the energies of all connected pipes. Let

$$\bar{e}_{max}(t) = \max_i e^i_{max}(t)$$

be the maximum essential bound of all edges. Then

$$\dot{e}_V = f(t, e_V) \leq \frac{deg(\mathcal{G}) v_{max}}{\tilde{V}} \left(e_{max}(t) + e_V\right)$$

and with Gronwall's inequality

$$|e_V(t) - e_V(0)| \leq C e^{Ct} \int_0^t e_{max}(\tau)\, \mathrm{d}\tau\,, \tag{3.36}$$

where $C = \frac{deg(\mathcal{G}) v_{max}}{\tilde{V}}$. For an arbitrary edge in the network, the left and right boundary values come from some nodes in the network. Inserting (3.36) into (3.35) and applying Gronwall one more time gives

$$|e(t,x) - e(0,x)| \leq \exp(e_{max,0}\Theta(t) C t e^{Ct}).$$

Where $e_{max,0}$ is the essential supremum of the initial datum of the network and the boundary data at the source. The choice $e_{max} = \exp(e_{max,0}\Theta(T) C T e^{CT})$ concludes the proof. □

Furthermore also the total variation of the solution has to be bounded. This result directly transfers from the initial and boundary data.

Lemma 3.6.2 ***(Bound on total variation)***
Assume e_L, e_R and $\bar{e}$ to be functions with bounded total variation and g a globally Lipschitz continuous function with L-constant G. Let e be the solution of

$$\begin{aligned} &\partial_t e + v\partial_x e = g(e) \qquad x \in (a,b),\, t > 0 \\ &e(t,a) = e_L(t) \\ &e(t,b) = e_R(t) \\ &e(0,x) = \bar{e}(x). \end{aligned} \tag{3.37}$$

Then it holds

$$\begin{aligned} \mathrm{TV}\left(e(t)\right) \leq &\exp\left(Gt\right)\left[\mathrm{TV}\left(e_L\right) + |e_L(0+) - \bar{e}(a_L+)| + \mathrm{TV}\left(\bar{e}\right)\right. \\ &\left. + |\bar{e}(b_R-) - e_R(0+)| + \mathrm{TV}\left(e_R\right)\right] \end{aligned} \qquad t \in \mathbb{R}^+ \,.$$

Proof. We use the representation of e with the method of characteristics and denote by

$$X\left(\tau; t, x\right) := x + \int_t^\tau v(s)\, \mathrm{d}s$$

the characteristic curve. Further we define the points

$$\begin{aligned} a_L &= \inf\{x \in (a,b) : X(\tau;t,x) > a \quad \forall \tau \in [0,t]\} \\ b_R &= \sup\{x \in (a,b) : X(\tau;t,x) < b \quad \forall \tau \in [0,t]\}. \end{aligned}$$

For the total variation of e this leads to the following estimate

$$\begin{aligned} \mathrm{TV}(e(t,\cdot)) \leq & \exp(Gt)\,[\mathrm{TV}(e(t,[a,a_L)) + |e(t,a_L-) - e(t,a_L+)| \\ & + \mathrm{TV}(e(t,(a_L,a_R)) + |e(t,b_R-) - e(t,b_B+)| + \mathrm{TV}(e(t,(b_R,b])] \\ \leq & \exp(Gt)\,[\mathrm{TV}(e_L) + |e_L(0+) - \bar{e}(a_L+)| + \mathrm{TV}(\bar{e}) \\ & + |\bar{e}(b_R-) - e_R(0+)| + \mathrm{TV}(e_R)]\ . \end{aligned}$$

□

The last lemma we need for the final proof is the stability of compositions. In many parts of the proof of Lemma 3.4.2 the energy solution e is evaluated along two different paths. Constructing the solution by characteristics, many terms of the form

$$\left\| e(0, \int_0^t v_1(\tau)d\tau) - e(0, \int_0^t v_2(\tau)d\tau) \right\|_{\mathbf{L^1}(0,t)}$$

arise, that we need to estimate in terms of the difference in the velocities. The next lemma provides the necessary properties. It turns out that the inner functions of the composition need to be bilipschitz, that means they have to be Lipschitz continuous with Lipschitz continuous inverse.

Lemma 3.6.3 ***(Stability of compositions)***
*Let $s_1(x), s_2(x) \in$ **Bilip**$([a,b],[a',b'])$ be bilipschitz functions and $f \in L^1([a',b'],\mathbb{R})$ have bounded total variation. Denote the maximum of the bilipschitz constants of s_1 and s_2 by K.*
Then

$$\|f \circ s_1 - f \circ s_2\|_{L^1[a,b]} \leq TV(f) K \|s_1 - s_2\|_{L^\infty([a,b])}.$$

Proof. Define $\epsilon(x) = (s_2(x) - s_1(x))$ and $m(x) = \min\{s_1(x), s_1(x) + \epsilon(x)\}$ and $M(x) = \max\{s_1(x), s_1(x) + \epsilon(x)\}$. Then

$$\max_x (M(x) - m(x)) = \max_x |\epsilon(x)| = \|s_1 - s_2\|_\infty$$

and both $m(x)$ and $M(x)$ are strictly monotone due to the bilipschitz property of s_1, s_2. Also, m and M are bilipschitz with L-constant K, i.e.

$$|x_1 - x_2| \leq K|m(x_1) - m(x_2)|.$$

Furthermore, we define $E(x)$ as the total variation of f on $[a', b']$. Then by using the monotonicity of the total variation and by changing the direction of integration:

$$\begin{aligned}
\|f \circ s_1 - f \circ s_2\|_{L^1[a,b]} &= \int_a^b |f(s_1(x)) - f(s_1(x) + \epsilon(x))| dx \\
&= \int_a^b |f(M(x)) - f(m(x))| dx \\
&\leq \int_a^b E(M(x)) - E(m(x)) dx \\
&= meas\{(x, y) \in [a, b] \times \mathbb{R}; E(m(x)) < y < E(M(x))\} \\
&= \int_{E(m(a))}^{E(M(b))} meas\{x; E(m(x)) < y < E(M(x))\} dy \\
&\leq \int_{E(m(a))}^{E(M(b))} meas\{x; E(m(x)) < y < E(m(x) + \|s_1 - s_2\|_\infty)\} dy \\
&\leq \int_0^{TV(f)} K \|s_1 - s_2\|_\infty \; dy \\
&= TV(f) K \|s_1 - s_2\|_\infty .
\end{aligned}$$

□

Remark:

1. Bilipschitz property of m:
   The minimum of two functions can be written as
   $$\min\{f_1, f_2\} = \frac{f_1 + f_2 - |f_1 - f_2|}{2}.$$
   Furthermore, the absolute value as well as the sum of Lipschitz continuous functions is again Lipschitz continuous.

2. For a velocity $v \in L^1$ with $0 < v_{min} < v < v_{max}$ the integral $s(t) = \int_0^t v(\tau) d\tau$ is bilipschitz with constant $K = \max\{\frac{1}{v_{min}}, v_{max}\}$.

3. If there is no lower bound on v, $s(t)$ is not bilipschitz. Then only a similar estimate on the fluxes holds:
   Let $|v_1(t)| \leq |v_2(t)|$ and $s_1(t), s_2(t)$ be defined as in 2. Then
   $$\|v_1 f \circ s_1 - v_1 f \circ s_2\|_{\mathbf{L^1}[a,b]} \leq TV(f) \|s_1 - s_2\|_\infty$$
   due to $\frac{|v_1(t)|}{|s_i'(t)|} \leq 1$, $i = 1, 2$.

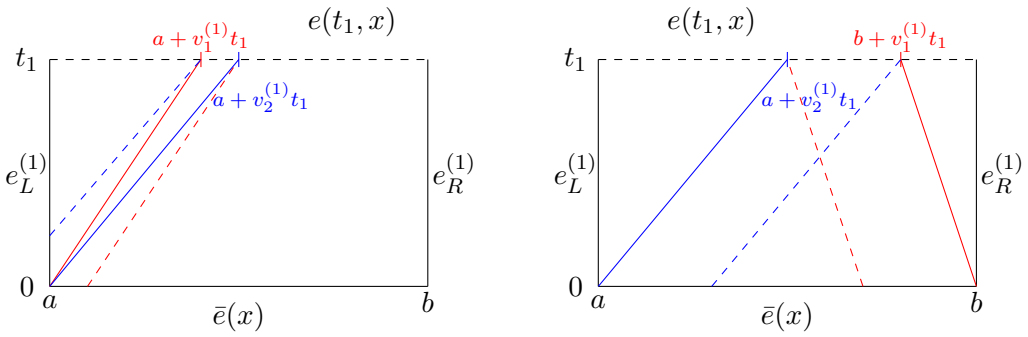

Figure 3.7: Illustration of the characteristics belonging to v_1 (red) and v_2 (blue) for the two cases $0 \leq v_1^{(1)} \leq v_2^{(1)}$ (left) and $v_1^{(1)} \leq 0 \leq v_2^{(1)}$ (right)

3.6.1 Proof of Lemma 3.4.2

Proof. Since $e_L^1, e_L^2, e_R^1, e_R^2, v_1, v_2$ have bounded variation we can approximate them by piecewise constant functions, each with a possibly different partition of $(0, t)$. Let the combination of all these partitions be denoted by $0 < t_1 < t_2 < \cdots < t_N = t$. We call $e_L^{1,(i)}, e_R^{1,(i)}, e_L^{2,(i)}, e_R^{2,(i)}, v_1^{(i)}, v_2^{(i)}$, $i = 1, \ldots, N$, the respective values of the step functions. Further we require from the step functions that $\sum_{i=0}^{N-1} \Delta t_i \left| v_1^{(i)} - v_2^{(i)} \right| \leq \|v_1 - v_2\|_{\mathbf{L^1}(0,t)}$ and $\sum_{i=0}^{N-1} \Delta t_i \left| e_\ell^{1,(i)} - e_\ell^{2,(i)} \right| \leq \left\| e_\ell^1 - e_\ell^2 \right\|_{\mathbf{L^1}(0,t)}$ for $\ell = L, R$. This is possible if e.g. v_1 is approximated from below and v_2 is approximated from above if their sign is positive and the other way around if the sign is negative. Denote the solutions to the corresponding problems $\hat{e}_1$ and $\hat{e}_2$.

We start with (3.14). First we show the estimate for one single time interval and then put them in a successive order. We start with (1), the (i)-th step is almost identical.
Consider the two cases $0 \leq v_1^{(1)} \leq v_2^{(1)}$ and $v_1^{(1)} < 0 \leq v_2^{(1)}$, compare Figure 3.7. All other possible combinations are similar.
We start with $0 \leq v_1^{(1)} \leq v_2^{(1)}$:
As the transport velocity is constant we can construct the solution up to t_1 by the method of characteristics. Regarding the source term we use that for the IVP $\dot{y} = g(y)$, $y(0) = y_0$ the estimate $y(t) \leq y_0 e^{Gt}$ holds and introduce $\theta(t) = 2e^{Gt}$. Using this we obtain

$$\|\hat{e}_1(t_1, \cdot) - \hat{e}_2(t_1, \cdot)\|_{\mathbf{L^1}(a,b)} \leq \underbrace{\int_a^{a + t_1 v_1^{(1)}} \theta(t_1) \left| e_L^{1,(0)} - e_L^{2,(0)} \right| \mathrm{d}x}_{=I_1}$$

$$+ \underbrace{\int_{a+t_1 v_2^{(1)}}^{a+t_1 v_2^{(1)}} \theta(t_1) \left| e_L^{(0)} - \bar{e}_1(x - t_1 v_1^{(1)}) \right| \mathrm{d}x}_{=I_2}$$
$$+ \underbrace{\int_{a+t_1 v_2^{(1)}}^{b} \theta(t_1) \left| \bar{e}_2(x - t_1 v_2^{(1)}) - \bar{e}_1(x - t_1 v_1^{(1)}) \right| \mathrm{d}x}_{=I_3}$$

The individual parts can be estimated by

$$I_1 = \theta(t_1) v_1^{(1)} t_1 \left| e_L^{1,(0)} - e_L^{2,(0)} \right|$$
$$I_2 \leq \theta(t_1) \left(\|\bar{e}_1\|_\infty + \left| e_L^{(0)} \right| \right) t_1 \left| v_2^{(1)} - v_1^{(1)} \right|$$

and

$$\begin{aligned} I_3 &\leq \theta(t_1) \int_{a+t_1 v_2^{(1)}}^{b} \left| \bar{e}_2(x - t_1 v_2^{(1)}) - \bar{e}_2(x - t_1 v_1^{(1)}) \right| + \left| \bar{e}_2(x - t_1 v_1^{(1)}) - \bar{e}_1(x - t_1 v_1^{(1)}) \right| \mathrm{d}x \\ &\leq \theta(t_1) \int_{a}^{b - t_1 v_2^{(1)}} \left| \bar{e}_2(x) - \bar{e}_2(x - t_1 \left(v_2^{(1)} - v_1^{(1)} \right)) \right| \mathrm{d}x + \theta(t_1) \|\bar{e}_1 - \bar{e}_2\|_{\mathbf{L^1}(a,b)} \\ &\leq \theta(t_1) \operatorname{TV}(\bar{e}_2) t_1 \left| v_2^{(1)} - v_1^{(1)} \right| + \theta(t_1) \|\bar{e}_1 - \bar{e}_2\|_{\mathbf{L^1}(a,b)}, \end{aligned}$$

where in the last step we use Lemma 3.6.3. Combining we obtain

$$\begin{aligned} \|\hat{e}_1(t_1, \cdot) - \hat{e}_2(t_1, \cdot)\|_{\mathbf{L^1}(a,b)} \leq & K_1^{(1)} t_1 \left| e_L^{1,(0)} - e_L^{2,(0)} \right| + K_2^{(1)} t_1 \left| v_2^{(1)} - v_1^{(1)} \right| \\ & + \theta(t_1) \|\bar{e}_1 - \bar{e}_2\|_{\mathbf{L^1}(a,b)} \end{aligned}$$

Similarly we proceed in the case $v_1 < 0 \leq v_2$

$$\begin{aligned} \|\hat{e}_1(t_1, \cdot) - \hat{e}_2(t_1, \cdot)\|_{\mathbf{L^1}(a,b)} \leq & \underbrace{\int_{a}^{a+t_1 v_2^{(1)}} \theta(t_1) \left| \bar{e}_1 \left(x - t_1 v_1^{(1)} \right) - e_L^{(0)} \right| \mathrm{d}x}_{=I_4} \\ & + \underbrace{\int_{a+t_1 v_2^{(1)}}^{b+t_1 v_1^{(1)}} \theta(t_1) \left| \bar{e}_1(x - t_1 v_1^{(1)}) - \bar{e}_2(x - t_1 v_2^{(1)}) \right| \mathrm{d}x}_{=I_5} \\ & + \underbrace{\int_{a+t_1 v_2^{(1)}}^{b} \theta(t_1) \left| \bar{e}_R^{(0)} - \bar{e}_2(x - t_1 v_2^{(1)}) \right| \mathrm{d}x}_{=I_6}. \end{aligned}$$

For the parts we have

$$I_5 \leq \theta(t_1) \|\bar{e}_1 - \bar{e}_2\|_{\mathbf{L^1}(a,b)} + \theta(t_1) \operatorname{TV}(\bar{e}_2) t_1 \left| v_2^{(1)} - v_1^{(1)} \right|$$

and

$$I_4 \leq \theta(t_1)\left(\|\bar{e}_1\|_\infty + \left|e_L^{(0)}\right|\right) t_1 \left|v_2^{(1)}\right| \overset{v_1^{(1)}<0<v_2^{(2)}}{\leq} \theta(t_1)\left(\|\bar{e}_1\|_\infty + \left|e_L^{(0)}\right|\right) t_1 \left|v_2^{(1)} - v_1^{(1)}\right|$$

$$I_6 \leq \theta(t_1)\left(\|\bar{e}_2\|_\infty + \left|e_R^{(0)}\right|\right) t_1 \left|v_1^{(1)}\right| \overset{v_1^{(1)}<0<v_2^{(2)}}{\leq} \theta(t_1)\left(\|\bar{e}_2\|_\infty + \left|e_R^{(0)}\right|\right) t_1 \left|v_2^{(1)} - v_1^{(1)}\right|.$$

Altogether this leads to

$$\|\hat{e}_1(t_1,\cdot) - \hat{e}_2(t_1,\cdot)\|_{\mathbf{L^1}(a,b)} \leq K_3^{(1)} t_1 \left|v_1^{(1)} - v_2^{(1)}\right| + \theta(t_1)\,\|\bar{e}_1 - \bar{e}_2\|_{\mathbf{L^1}(a,b)}\,.$$

Note that the $K_2^{(1)}$ and $K_3^{(1)}$ depend on $\mathrm{TV}\,(\bar{e}_1)$, $\mathrm{TV}\,(\bar{e}_2)$, $\|\bar{e}_1\|_\infty$ and $\|\bar{e}_2\|_\infty$. From Lemma 3.6.1 we obtain the bound

$$\|\hat{e}_j(t_{i+1},\cdot)\|_\infty \leq \theta(\Delta t_i)\max\left(\|\hat{e}_j(t_i,\cdot)\|_\infty, \left|e_L^{(i)}\right|, \left|e_R^{(i)}\right|\right)\,,$$

for $j = 1, 2$. For the total variation we do not need an iterative perspective and have from Lemma 3.6.2 the global property

$$\begin{aligned}\mathrm{TV}\,(e_j(t)) \leq &\exp(Gt)\left[\mathrm{TV}\left(e_L^{(j)}\right) + \left|e_L^{(j)}(0+) - \bar{e}_j(a+)\right| + \mathrm{TV}\,(\bar{e}_j)\right.\\ &\left.+\left|\bar{e}_j(b-) - e_R^{(j)}(0+)\right| + \mathrm{TV}\left(e_R^{(j)}\right)\right] =: K_{\mathrm{TV}}\end{aligned} \qquad t \in \mathbb{R}^+,\ j = 1,2\ .$$

Together this leads to the uniform statements

$$K_j^{(i)} \leq 2\theta(t)\max\left(\|\bar{e}_1\|_\infty, \|\bar{e}_2\|_\infty, \|e_L\|_\infty, \|e_R\|_\infty\right) + K_{\mathrm{TV}} \qquad j = 2,3\ .$$

For $K_1^{(i)}$ we have $K_1^{(i)} \leq \max\left(\|v_1\|_\infty, \|v_2\|_\infty\right)$.

Thus, the constants can be chosen independent of i. For a general step, without assumptions on $v_1^{(i)}$ and $v_2^{(i)}$, this reads as

$$\begin{aligned}\|\hat{e}_1(t_{i+1},\cdot) - \hat{e}_2(t_{i+1},\cdot)\|_{\mathbf{L^1}(a,b)} \leq &K_L \Delta t_i \left|e_L^{1,(i)} - e_L^{2,(i)}\right| + K_R \Delta t_i \left|e_R^{1,(i)} - e_R^{2,(i)}\right|\\ &+ K_v \Delta t_i \left|v_1^{(i)} - v_2^{(i)}\right| + \theta\left(\Delta t_i\right)\|\hat{e}_1(t_i,\cdot) - \hat{e}_2(t_i,\cdot)\|_{\mathbf{L^1}(a,b)}\end{aligned}$$

with $\Delta t_i = t_{i+1} - t_i$. Iterative inserting leads to the desired estimate

$$\begin{aligned}\|\hat{e}_1(t,\cdot) - \hat{e}_2(t,\cdot)\|_{\mathbf{L^1}(a,b)} \leq &K_L \sum_{i=0}^{N-1} \theta\left(\Delta t_{i+1}\right)\Delta t_i \left|e_L^{1,(i)} - e_L^{2,(i)}\right|\\ &+ K_R \sum_{i=0}^{N-1} \theta\left(\Delta t_{i+1}\right)\Delta t_i \left|e_R^{1,(i)} - e_R^{2,(i)}\right|\end{aligned}$$

$$
\begin{aligned}
&+ K_v \sum_{i=0}^{N-1} \theta\left(\Delta t_{i+1}\right) \Delta t_i \left| v_1^{(i)} - v_2^{(i)} \right| \\
&+ \sum_{i=0}^{N-1} \left(\prod_{j=0}^{i} \theta\left(\Delta t_j\right) \right) \left\| \bar{e}_1 - \bar{e}_2 \right\|_{\mathbf{L^1}(a,b)} \\
\leq & \tilde{K}_L \left\| e_L^1 - e_L^2 \right\|_{\mathbf{L^1}(0,t)} + \tilde{K}_R \left\| e_R^1 - e_R^2 \right\|_{\mathbf{L^1}(0,t)} \\
&+ \tilde{K}_v \left\| v_1 - v_2 \right\|_{\mathbf{L^1}(0,t)} + \tilde{K}_I \theta(t) \left\| \bar{e}_1 - \bar{e}_2 \right\|_{\mathbf{L^1}(a,b)},
\end{aligned}
$$

with the definition $\Delta t_N = 0$. The last inequality holds, since we have $\prod_{i=0}^{N} \exp\left(G \Delta t_i\right) = \exp\left(Gt\right)$. As this is independent of the discretization of the step functions we obtain (3.14).

In the second part we show (3.16), the proof for (3.15) is symmetric. We remain in the setting with piece wise constant velocities and boundary functions as above. Thus the norm of the trace of the flux at the right boundary can be split accordingly

$$
\left\| v_1^+ e_1(\cdot, b) - v_2^+ e_2(\cdot, b) \right\|_{\mathbf{L^1}(0,t)} = \sum_{i=0}^{N-1} \left\| v_1^{(i),+} e_1(\cdot, b) - v_2^{(i),+} e_2(\cdot, b) \right\|_{\mathbf{L^1}(t_i, t_{i+1})} . \quad (3.38)
$$

Each part can be treated separately. According to the signs of the velocities we have four different cases.
1) $v_1^{(i),+} \leq 0, v_2^{(i),+} \leq 0$ is trivial.
2) $v_1^{(i),+} > 0, v_2^{(i),+} \leq 0$

$$
\left\| v_1^{(i),+} e_1(\cdot, b) - v_2^{(i),+} e_2(\cdot, b) \right\|_{\mathbf{L^1}(t_i, t_{i+1})} = v_1^{(i)} \left\| e_1(\cdot, b) \right\|_{\mathbf{L^1}(t_i, t_{i+1})} \leq \Delta t \left| v_1^{(i)} - v_2^{(i)} \right| e_{max} .
$$

3) $v_1^{(i),+} \leq 0, v_2^{(i),+} > 0$ is similar to 2).
4) $v_1^{(i),+} > 0, v_2^{(i),+} > 0$: We trace back the characteristics

$$
\begin{aligned}
&\left\| v_1^{(i),+} e_1(\cdot, b) - v_2^{(i),+} e_2(\cdot, b) \right\|_{\mathbf{L^1}(I_i)} \\
&= \theta(\Delta t) \int_0^{\Delta t} \left| v_1^{(i)} e_1(t_i, b - v_1^{(i)} \tau) - v_2^{(i)} e_2(t_i, b - v_2^{(i)} \tau) \right| \mathrm{d}\tau \\
&\leq \theta(\Delta t) \int_0^{\Delta t} \left| v_1^{(i)} e_1(t_i, b - v_1^{(i)} \tau) - v_2^{(i)} e_2(t_i, b - v_1^{(i)} \tau) \right| \\
&\qquad + \left| v_2^{(i)} e_2(t_i, b - v_1^{(i)} \tau) - v_2^{(i)} e_2(t_i, b - v_2^{(i)} \tau) \right| \mathrm{d}\tau \\
&\leq \theta(\Delta t) \int_0^{\Delta t_i} \left| v_1^{(i)} e_1(t_i, b - v_1^{(i)} \tau) - v_2^{(i)} e_2(t_i, b - v_1^{(i)} \tau) \right| \mathrm{d}\tau \\
&\quad + \theta(\Delta t) \, \mathrm{TV}\left(e_2\right) \Delta t_i \left| v_2^{(i)} - v_1^{(i)} \right| .
\end{aligned}
$$

From the first term we separate the velocity

$$\begin{aligned}
&\int_0^{\Delta t} \left| v_1^{(i)} e_1(t_i, b - v_1^{(i)}\tau) - v_2^{(i)} e_2(t_i, b - v_1^{(i)}\tau) \right| \mathrm{d}\tau \\
&\leq \int_0^{\Delta t} \left| v_1^{(i)} e_1(t_i, b - v_1^{(i)}\tau) - v_1^{(i)} e_2(t_i, b - v_1^{(i)}\tau) \right| \\
&\qquad + \left| v_1^{(i)} e_2(t_i, b - v_1^{(i)}\tau) - v_2^{(i)} e_2(t_i, b - v_1^{(i)}\tau) \right| \mathrm{d}\tau \\
&\leq \int_{b - v_1^{(i)}\Delta t}^{b} |e_1(t_i, y) - e_2(t_i, y)| \, \mathrm{d}y + 2 e_{max} \Delta t \left| v_2^{(i)} - v_1^{(i)} \right|.
\end{aligned}$$

For the first term we need an estimate, which does not collect too much spatial information, when the number of discretization steps t_i gets large. Therefore we first look how on a small interval $[a_i, b_i]$ the term $\int_{a_i}^{b_i} |e_1(t_i, x) - e_2(t_i, x)| \, \mathrm{d}x$ can be bounded by its shift with v_1. We choose $a_i = b - v_1^{(i)}\Delta t$, $b_i = b$. The following estimates depend on the different velocities $v_1^{(i-1)}, v_2^{(i-1)}$ of the previous time interval I_{i-1}. Again we consider different cases depending on the discrete velocities.

A) $v_1^{(i-1)} \geq 0, v_2^{(i-1)} \geq 0$:

$$\begin{aligned}
&\int_{a_i}^{b_i} |e_1(t_i, x) - e_2(t_i, x)| \, \mathrm{d}x \\
&\leq \theta(\Delta t) \int_{a_i}^{b_i} \left| e_1(t_{i-1}, x - \Delta t v_1^{(i-1)}) - e_2(t_{i-1}, x - \Delta t v_2^{(i-1)}) \right| \mathrm{d}x \\
&\leq \theta(\Delta t) \int_{a_i}^{b_i} \left| e_1(t_{i-1}, x - \Delta t v_1^{(i-1)}) - e_2(t_{i-1}, x - \Delta t v_1^{(i-1)}) \right| \\
&\qquad + \left| e_2(t_{i-1}, x - \Delta t v_1^{(i-1)}) - e_2(t_{i-1}, x - \Delta t v_2^{(i-1)}) \right| \mathrm{d}x \\
&\leq \theta(\Delta t) \int_{a_i - \Delta t v_1^{(i-1)}}^{b_i - \Delta t v_1^{(i-1)}} |e_1(t_{i-1}, x) - e_2(t_{i-1}, x)| \, \mathrm{d}x + \theta(\Delta t) \, \mathrm{TV}\,(e_2) \, \Delta t \left| v_2^{(i-1)} - v_1^{(i-1)} \right|.
\end{aligned}$$

B) $v_1^{(i-1)} \geq 0$, $v_2^{(i-1)} < 0$: Here the characteristic of e_2 can possibly hit the right boundary of the domain. Therefore we introduce $c' = b + \Delta t v_2 < b_i$ and $c = \max(a_i, c')$. Furthermore, define the transformation from space to time corresponding to the velocity v_l by $\tau_l(x) = t_{i-1} - \frac{x - c'}{v_l}$, $l = 1, 2$.
Thus the integral is split into

$$\begin{aligned}
&\int_{a_i}^{b_i} |e_1(t_i, x) - e_2(t_i, x)| \, \mathrm{d}x \\
&\leq \theta(\Delta t) \int_{a_i}^{c} \left| e_1(t_{i-1}, x - \Delta t v_1^{(i-1)}) - e_2(t_{i-1}, x - \Delta t v_2^{(i-1)}) \right| \mathrm{d}x
\end{aligned}$$

$$
\begin{aligned}
&+ \theta(\Delta t) \int_c^{b_i} \left| e_1(t_{i-1}, x - \Delta t v_1^{(i-1)}) - e_{2,R}(\tau_2(x)) \right| \mathrm{d}x \\
&\leq \theta(\Delta t) \int_{a_i - \Delta t v_1^{(i-1)}}^{c - \Delta t v_1^{(i-1)}} |e_1(t_{i-1}, x) - e_2(t_{i-1}, x)| \, \mathrm{d}x \\
&\quad + \theta(\Delta t) \, \mathrm{TV}\,(e_2) \, \Delta t \left| v_2^{(i-1)} - v_1^{(i-1)} \right| + \theta(\Delta t) 2 e_{max} \left| v_2^{(i-1)} - v_1^{(i-1)} \right| \Delta t,
\end{aligned}
$$

where we used $b_i - c \leq |v_2 - v_1| \Delta t$ and a computation similar to case A).

C) $v_2 \geq 0$, $v_1 < 0$: This case is similar to B). Note that it is not completely symmetric in v_1 and v_2, as the interval is always shifted according to v_1, but the computation is not affected, except that $c' = b_i + \Delta t v_1$. As above the boundary term $e_{R,1}$ is not occurring, as they are hidden in e_{max}.

D) $v_2 \leq v_1 < 0$: We define $c_1 = \max(a_i, b_i + \Delta t v_1)$ and $c_2 = \max(a_i, b_i + \Delta t v_2)$.

$$
\begin{aligned}
&\int_{a_i}^{b_i} |e_1(t_i, x) - e_2(t_i, x)| \, \mathrm{d}x \\
&\leq \theta(\Delta t) \int_{a_i}^{c_2} \left| e_1(t_{i-1}, x - \Delta t v_1^{(i-1)}) - e_2(t_{i-1}, x - \Delta t v_2^{(i-1)}) \right| \mathrm{d}x \\
&\quad + \theta(\Delta t) \int_{c_2}^{c_1} \left| e_1(t_{i-1}, x - \Delta t v_1^{(i-1)}) - e_{2,R}(\tau_2(x)) \right| \mathrm{d}x \\
&\quad + \theta(\Delta t) \int_{c_1}^{b_i} |e_{1,R}(\tau_1(x)) - e_{2,R}(\tau_2(x))| \, \mathrm{d}x \\
&\leq \theta(\Delta t) \int_{a_i - \Delta t v_1^{(i-1)}}^{c_2 - \Delta t v_1^{(i-1)}} |e_1(t_{i-1}, x) - e_2(t_{i-1}, x)| \, \mathrm{d}x + \theta(\Delta t) \, \mathrm{TV}\,(e_2) \, \Delta t \left| v_1^{(i-1)} - v_2^{(i-1)} \right| \\
&\quad + \theta(\Delta t) 2 e_{max} \left| v_1^{(i-1)} - v_2^{(i-1)} \right| \Delta t + \theta(\Delta t) \int_{c_1}^{b_i} |e_{1,R}(\tau_1(x)) - e_{2,R}(\tau_2(x))| \, \mathrm{d}x \,,
\end{aligned}
$$

similar to above cases. The last term is estimated by

$$
\begin{aligned}
&\int_{c_1}^{b_i} |e_{1,R}(\tau_1(x)) - e_{2,R}(\tau_2(x))| \, \mathrm{d}x \\
&\leq \int_{c_1}^{b_i} |e_{1,R}(\tau_1(x)) - e_{1,R}(\tau_2(x))| + |e_{1,R}(\tau_2(x)) - e_{2,R}(\tau_2(x))| \, \mathrm{d}x \\
&\leq \int_{c_1}^{b_i} |e_{1,R}(\tau_1(x)) - e_{1,R}(\tau_2(x))| \, \mathrm{d}x + v_{max} \, \| e_{1,R} - e_{2,R} \|_{\mathbf{L^1}(t_{i-1}, t_i)} \\
&= \int_{c_1}^{b_i} \left| e_{1,R}(t_{i-1} - \frac{x - c_1}{v_1^{(i-1)}}) - e_{1,R}(t_{i-1} - \frac{x - c_2}{v_2^{(i-1)}}) \right| \mathrm{d}x + v_{max} \, \| e_{1,R} - e_{2,R} \|_{\mathbf{L^1}(t_{i-1}, t_i)}
\end{aligned}
$$

$$
\begin{aligned}
&= \left|v_1^{(i-1)}\right| \int_{t_{i-1}}^{t_i} \left| e_{1,R}(\tau) - e_{2,R}(\frac{v_1^{(i-1)}}{v_2^{(i-1)}}\tau + (1 - \frac{v_1^{(i-1)}}{v_2^{(i-1)}})t_i) \right| \mathrm{d}\tau \\
&\quad + v_{max} \left\| e_{1,R} - e_{2,R} \right\|_{\mathbf{L^1}(t_{i-1},t_i)} \\
&\leq TV(e_{1,R}) |v_1^{(i-1)}| \max_{\tau} \left| \frac{v_1^{(i-1)}}{v_2^{(i-1)}}\tau + (1 - \frac{v_1^{(i-1)}}{v_2^{(i-1)}})t_i \right| + v_{max} \left\| e_{1,R} - e_{2,R} \right\|_{\mathbf{L^1}(t_{i-1},t_i)} \\
&\leq \Delta t \; TV(e_1) \frac{v_1^{(i-1)}}{v_2^{(i-1)}} |v_1^{(i-1)} - v_2^{(i-1)}| + v_{max} \left\| e_{1,R} - e_{2,R} \right\|_{\mathbf{L^1}(t_{i-1},t_i)} \\
&\overset{|v_1^{(i-1)}|<|v_2^{(i-1)}|}{\leq} TV(e_1) \left\| v_1 - v_2 \right\|_{\mathbf{L^1}(t_{i-1},t_i)} + v_{max} \left\| e_{1,R} - e_{2,R} \right\|_{\mathbf{L^1}(t_{i-1},t_i)}.
\end{aligned}
$$

So the full estimate for this case reads

$$
\begin{aligned}
&\int_{a_i}^{b_i} |e_1(t_i,x) - e_2(t_i,x)| \, \mathrm{d}x \\
&\leq \theta(\Delta t) \left[\int_{a_i - \Delta t v_1^{(i-1)}}^{c_2 - \Delta t v_1^{(i-1)}} |e_1(t_{i-1},x) - e_2(t_{i-1},x)| \, \mathrm{d}x \right. \\
&\quad + (\mathrm{TV}\,(e_2) + \mathrm{TV}\,(e_{1,R}) + 2e_{max}) \, \Delta t \left| v_1^{(i-1)} - v_2^{(i-1)} \right| \\
&\quad \left. + v_{max} \Delta t \left| e_{1,R}^{(i-1)} - e_{2,R}^{(i-1)} \right| \right].
\end{aligned}
$$

E) $v_1 < v_2 < 0$ This case is similar to D).

We now combine all terms for (3.38), starting with I_i and tracing the space integrals back until we reach the initial condition. As shown above, we get

$$
\begin{aligned}
&\left\| v_1^{(i),+} e_1(\cdot,b) - v_2^{(i),+} e_2(\cdot,b) \right\|_{\mathbf{L^1}(I_i)} \\
&\leq \theta(\Delta t) \left[\int_0^{\Delta t_i} \left| v_1^{(i)} e_1(t_i, b - v_1^{(i)}\tau) - v_2^{(i)} e_2(t_i, b - v_1^{(i)}\tau) \right| \mathrm{d}\tau + \mathrm{TV}\,(e_2) \Delta t_i \left| v_2^{(i)} - v_1^{(i)} \right| \right] \\
&\leq \theta(\Delta t) \left[\int_{b - v_1^{(i)} \Delta t}^{b} |e_1(t_i,x) - e_2(t_i,x)| \, \mathrm{d}x + 2e_{max} \Delta t \left| v_2^{(i)} - v_1^{(i)} \right| + \mathrm{TV}\,(e_2) \Delta t_i \left| v_2^{(i)} - v_1^{(i)} \right| \right] \\
&\leq \theta(\Delta t)^2 \left[\int_{a_{i-1}}^{b'_{i-1}} |e_1(t_{i-1},x) - e_2(t_{i-1},x)| \, \mathrm{d}x \right. \\
&\quad + (2e_{max} + \mathrm{TV}\,(e_2)) \, \Delta t \left| v_2^{(i)} - v_1^{(i)} \right| \\
&\quad + (\mathrm{TV}\,(e_2) + \mathrm{TV}\,(e_{1,R}) + 2e_{max}) \, \Delta t \left| v_1^{(i-1)} - v_2^{(i-1)} \right| \\
&\quad \left. + v_{max} \Delta t \left| e_{1,R}^{(i-1)} - e_{2,R}^{(i-1)} \right| \right],
\end{aligned}
$$

where we define

$$\begin{aligned} a_i' &= \min\{b - \Delta t v_1^{(i)}, b\}, \\ a_{i-1} &= \min\{a_i - \Delta t v_1^{(i-1)}, b\}, \\ a_{N-1} &= a_{N-1}' \end{aligned}$$

and

$$b_{i-1}' = \begin{cases} b - \Delta t v_1^{(i-1)}, & \text{if } 0 \le v_1^{(i-1)}, v_2^{(i-1)} \\ c - \Delta t v_1^{(i-1)}, & \text{if } v_2^{(i-1)} < 0 \le v_1^{(i-1)} \\ b, & \text{if } v_1^{(i-1)} < 0 \le v_2^{(i-1)} \\ c_2 - \Delta t v_1^{(i-1)}, & \text{if } v_2^{(i-1)} \le v_1^{(i-1)} < 0 \\ b, & \text{if } v_1^{(i-1)} \le v_2^{(i-1)} < 0, \end{cases}$$

depending on above cases of velocity signs. Adding the two terms for I_N and I_{N-1} leads to

$$\begin{aligned} &\sum_{i=N-1}^{N} \left\| v_1^{(i),+} e_1(\cdot, b) - v_2^{(i),+} e_2(\cdot, b) \right\|_{\mathbf{L^1}(I_i)} \\ &\le \theta(\Delta t)^2 \left[\int_{a_{i-1}}^{b_{i-1}'} |e_1(t_{i-1}, x) - e_2(t_{i-1}, x)| \, \mathrm{d}x \right. \\ &\quad + \int_{a_{i-1}'}^{b} |e_1(t_{i-1}, x) - e_2(t_{i-1}, x)| \, \mathrm{d}x \\ &\quad + (2e_{max} + \mathrm{TV}(e_2)) \, \Delta t \left| v_2^{(i)} - v_1^{(i)} \right| \\ &\quad + (2\,\mathrm{TV}(e_2) + \mathrm{TV}(e_{1,R}) + 4e_{max}) \, \Delta t \left| v_1^{(i-1)} - v_2^{(i-1)} \right| \\ &\quad \left. + v_{max} \Delta t \left| e_{1,R}^{(i-1)} - e_{2,R}^{(i-1)} \right| \right] \\ &\le \theta(\Delta t)^2 \left[\int_{a_{i-1}}^{b} |e_1(t_{i-1}, x) - e_2(t_{i-1}, x)| \, \mathrm{d}x \right. \\ &\quad + (2e_{max} + \mathrm{TV}(e_2)) \, \Delta t \left| v_2^{(i)} - v_1^{(i)} \right| \\ &\quad + (2\,\mathrm{TV}(e_2) + \mathrm{TV}(e_{1,R}) + 4e_{max}) \, \Delta t \left| v_1^{(i-1)} - v_2^{(i-1)} \right| \\ &\quad \left. + v_{max} \Delta t \left| e_{1,R}^{(i-1)} - e_{2,R}^{(i-1)} \right| \right], \end{aligned} \tag{3.39}$$

due to $a_{i-1}' \le b_{i-1}'$ for all velocity sign configurations. This can be seen in Figure 3.8, where several different cases are illustrated. There, the interval $[t_{N-2}, t_{N-1}]$ covers the case, where both velocities are positive. We have $b_{N-2}' = a_{N-2}' = b - \Delta t v_1^{(N-2)}$ and thus the two integrals combine seamlessly. The next interval shows the case where both velocities are negative, with $v_2^{N-3} < v_1^{N-3}$. In this setting, the contribution of

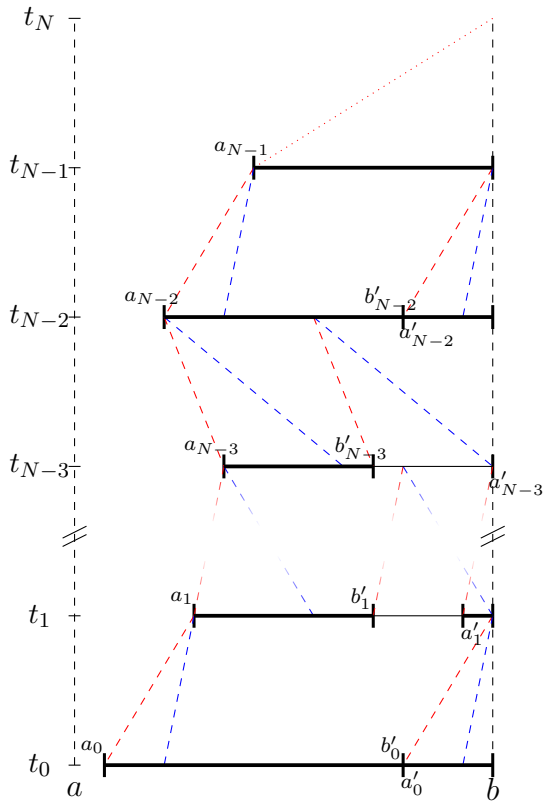

Figure 3.8: Iterative extension of the spatial integrals from (3.39)

this interval itself is zero with $a'_{N-3} = b$. The correct contribution to the estimate then only consists of the (smaller) interval $[a_{N-3}, b'_{N-3}]$ indicated by the bold line. By extending the integral to $[a_{N-3}, b]$ the estimate gets consistent with the first case. We can iteratively add new terms to the desired estimate of (3.38), each giving a contribution in terms controllable by $|v_1^{(i)} - v_2^{(i)}|$ or $|e_{1,R}^{(i)} - e_{2,R}^{(i)}|$ in their respective intervals as well as the space integral $\int_{a'_i}^{b} |e_1(t_i, x) - e_2(t_i, x)| \, \mathrm{d}x$.

For the interval $[t_1, t_2]$ with $v_2^{(1)} < 0 < v_1^{(1)}$ we get two nonempty disjoint segments $[a_1, b'_1]$ and $[a'_1, b]$, where again the integral just gets bigger by extending it to $[a_1, b]$. Finally in the last step to t_0 we have again the same case as in the first step with the integrals perfectly connecting. By assumption we have $a < b - \Delta t \sum_{i=1}^{N} |v_1^{(i)}| \leq a_0$ so at the last step we extend $[a_0, b]$ to $[a, b]$. This iterative procedure finally gives the estimate

$$\left\| v_1^+ e_1(\cdot, b) - v_2^+ e_2(\cdot, b) \right\|_{\mathbf{L^1}(0,t)} = \sum_{i=0}^{N-1} \left\| v_1^{(i),+} e_1(\cdot, b) - v_2^{(i),+} e_2(\cdot, b) \right\|_{\mathbf{L^1}(t_i, t_{i+1})}$$

$$
\begin{aligned}
&\leq \theta(t) \left[\int_a^b |e_1(0,x) - e_2(0,x)| \, \mathrm{d}x \right. \\
&\quad + (2\,\mathrm{TV}\,(e_2) + \mathrm{TV}\,(e_{1,R}) + 4e_{max}) \, \|v_1 - v_2\|_{\mathbf{L^1}(0,t)} \\
&\quad \left. + v_{max} \, \|e_{1,R} - e_{2,R}\|_{\mathbf{L^1}(0,t)} \right] \\
&= K_{I,b} \, \|\bar{e}_1 - \bar{e}_2\|_{\mathbf{L^1}(a,b)} + K_{v,b} \, \|v_1 - v_2\|_{\mathbf{L^1}(0,t)} + K_{R,b} \, \|e_{1,R} - e_{2,R}\|_{\mathbf{L^1}(0,t)} \, .
\end{aligned}
$$

□

Chapter 4

Numerical Methods: Explicit Schemes

In this chapter we will discuss several explicit schemes for the simulation of transport networks.
We first show the basic formulations of finite difference (FD), finite volume (FV) and discontinuous Galerkin (DG) schemes. In order to achieve high order in accuracy, ADER schemes are applied to the network of conservation laws. Due to the pure advection type of the equation, we can further improve the performance of the numerical scheme by applying local time stepping. Similar local time stepping algorithms have already been used in different applications. In this specific context however, the problem structure enables extremely efficient evaluations. Furthermore, we propose a new high order extension for the method on networks.

4.1 Advection Equation

The scalar advection equation is a hyperbolic conservation law of the form

$$u_t + v u_x = 0. \tag{4.1}$$

The energy transport system (2.22) consists of a set of N_p scalar PDEs. We study a single advection equation for the construction of the numerical schemes. The full network setting is achieved by posing suitable boundary conditions to the one dimensional schemes. The flow velocity can be a time dependent function $v = v(t)$. Due to the splitting of the full system in Section 2.3, we assume piece-wise constant velocities in the numerical methods. This means from one time step to the next the velocity can change, within a single step it is assumed to be constant.

For the advection equation, an explicit solution can be given. Defining the characteristics

$$X(t) = x_0 + \int_0^t v(\tau)\mathrm{d}\tau,$$

the time derivative of $u(X(t),t)$ is

$$\begin{aligned}\frac{d}{dt}u(X(t),t) &= u_t(X(t),t) + X'(t)u_x(X(t),t)\\ &= u_t + v(t)u_x\\ &= 0.\end{aligned}$$

Thus along the characteristics, the solution u is constant in time. For a Cauchy problem with initial condition $u(x,0) = u_0(x),\ x \in \mathbb{R}$ we can give the exact solution as just the translated initial datum

$$u(x,t) = u_0\left(x - \int_0^t v(\tau)\mathrm{d}\tau\right).$$

In most applications, the computational domain $\Omega = [a,b]$ is bounded. That means not only initial conditions, but also boundary values have to be prescribed. In those initial boundary value problems (IBVP) it is important to give the 'right' boundary data. For the advection equation with constant velocity, we need to specify the boundary values at the left boundary $x = a$, if $v > 0$ and on the right boundary $x = b$, if $v < 0$. In the case of linear systems, the boundary values must be given according to the eigenvalues of the system. For each positive eigenvalue, the corresponding characteristic variable needs a left boundary condition, while the characteristic variables with negative eigenvalues need one on the right.

4.2 Finite Volume Schemes

Although for the advection equation, explicit solutions can be given, their derivation might not always be easy. Especially in the network case, evaluation of the characteristics for a complicated velocity field is not possible. Therefore we need numerical schemes to approximate the exact solution of the problem [48].
Finite volume schemes are very popular and particularly suited for the solution of hyperbolic conservation laws. They are based on an integral form of the equation and by their construction will conserve the involved quantities.

We consider the one dimensional scalar conservation law

$$u_t + f(u)_x = 0 \tag{4.2}$$

on the space-time domain $\Omega_T = \mathbb{R} \times \mathbb{R}^+$. Space is discretized in equidistant steps x_i, with the grid size $\Delta x = x_{i+1} - x_i$. The time steps are denoted by $\Delta t_n = t_{n+1} - t_n$. We define the finite volume grid cells by $I_i = [x_{i-\frac{1}{2}}, x_{+\frac{1}{2}}] = [x_i - \frac{1}{2}\Delta x, x_i + \frac{1}{2}\Delta x]$. Furthermore the average of u for a given grid cell is denoted by

$$u_i^n = \frac{1}{\Delta x}\int_{x_{1-\frac{1}{2}}}^{x_{1+\frac{1}{2}}} u(x,t_n)\mathrm{d}x.$$

Integration of the conservation law (4.2) over a grid cell and the time interval $[t_n, t_{n+1}]$ leads to

$$\begin{aligned} 0 &= \int_{t_n}^{t_{n+1}} \int_{x_{1-\frac{1}{2}}}^{x+\frac{1}{2}} u_t + f(u)_x \mathrm{d}x\mathrm{d}t \\ &= \int_{x_{1-\frac{1}{2}}}^{x+\frac{1}{2}} u(x, t_{n+1}) - u(x, t_n)\mathrm{d}x + \int_{t_n}^{t_{n+1}} f(u(x_{i+\frac{1}{2}}), t) - f(u(x_{i-\frac{1}{2}}), t)\mathrm{d}t. \end{aligned}$$

After rearranging and division by Δx we obtain

$$u_i^{n+1} = u_i^n - \frac{\Delta t_n}{\Delta x}\left(f_{i+\frac{1}{2}}^n - f_{i-\frac{1}{2}}^n\right), \tag{4.3}$$

with the interface flux

$$f_{i+\frac{1}{2}}^n = \frac{1}{\Delta t_n}\int_{t_n}^{t_{n+1}} f(u(x_{i+\frac{1}{2}}), t)\mathrm{d}t. \tag{4.4}$$

Equation (4.3) gives an exact update formula for the cell values u_i^{n+1}, however the integral (4.4) generally can only be evaluated numerically. A key point of finite volume schemes is to find numerical fluxes $F_{i+\frac{1}{2}}^n$ approximating the analytical ones

$$F_{i+\frac{1}{2}}^n \approx f_{i+\frac{1}{2}}^n$$

and updating the new cell values with those approximations

$$u_i^{n+1} = u_i^n - \frac{\Delta t}{\Delta x}\left(F_{i+\frac{1}{2}}^n - F_{i-\frac{1}{2}}^n\right). \tag{4.5}$$

Schemes of form (4.5) always conserve the initial quantity in absence of boundaries.

4.2.1 Upwind Scheme

The most simple finite volume scheme for the advection equation (4.1) is the upwind scheme. Here, the numerical flux is determined by the cell value in upwind direction, or more precisely

$$F_{i+\frac{1}{2}}^n = \begin{cases} u_i^n, & \text{if } v > 0 \\ u_{i+1}^n, & \text{if } v < 0 \end{cases}.$$

For $v > 0$ the full scheme reads

$$u_i^{n+1} = u_i^n - v\frac{\Delta t}{\Delta x}\left(u_i^n - u_{i-1}^n\right). \tag{4.6}$$

By a von Neumann analysis the stability of the upwind scheme can be shown under the condition

$$c = v\frac{\Delta t}{\Delta x} < 1, \tag{4.7}$$

which is known as the Courant-Friedrichs-Levy (CFL) stability condition. The dimensionless CFL number c measures the number of grid cells that information can travel in one time step. Condition (4.7) means, that the analytical domain of dependence must be contained in the numerical domain of dependence, compare Figure 4.1. All explicit schemes are subject to a stability condition of that or a similar form. In fact all explicit FV schemes presented in this work require $c \leq 1$. The class of discontinuous Galerkin schemes even have a more restrictive stability condition. For a DG method of order k, one requires $c < \frac{1}{2k+1}$, see Section 4.2.7. An analysis of the leading error term

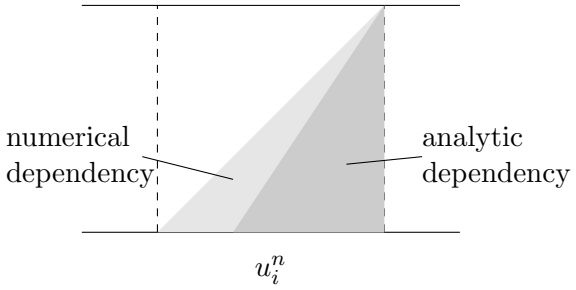

Figure 4.1: Data dependencies

of numerical schemes gives insight in the behavior of the computed solution. We can insert a Taylor expansion up to order two into the discrete scheme and get

$$
\begin{aligned}
& u(x,t+\Delta t) = u(x,t) - c\left(u(x,t) - u(x-\Delta x,t)\right) \\
\Rightarrow\ & u(x,t) + \Delta t u_t(x,t) + \frac{\Delta t^2}{2} u_{tt}(x,t) = u(x,t) - c\left(\Delta x u_x(x,t) - \frac{\Delta x^2}{2} u_{xx}(x,t)\right) + \mathcal{O}(\Delta x^3) \\
\Rightarrow\ & \Delta t u_t(x,t) + c\Delta x u_x(x,t) = \frac{\Delta x^2}{2} u_{xx}(x,t) - \frac{\Delta t^2}{2} u_{tt}(x,t) + \mathcal{O}(\Delta x^3) \\
\Rightarrow\ & u_t(x,t) + v u_x(x,t) = \frac{1}{2}\Delta x v\left(1-c\right) u_{xx}(x,t) + \mathcal{O}(\Delta x^3).
\end{aligned}
$$

The higher order terms in Δt are contained in $\mathcal{O}(\Delta x^3)$ due to (4.7), with $\Delta t = \frac{c}{v}\Delta x$. Hence the upwind scheme, first order accurate for the advection equation, solves the advection-diffusion equation

$$u_t(x,t) + v u_x(x,t) = \alpha u_{xx}(x,t)$$

with diffusion coefficient $\alpha = \frac{1}{2}\Delta x v\left(1-c\right)$ to second order accuracy. Note, that for $c = 1$, the diffusion coefficient is zero and the upwind scheme will in fact reproduce the exact solution to the advection problem. In most cases, especially for nonlinear problems it is impossible to run the computations with a specific given CFL number. However, for our setting we present a local time stepping scheme in Section 4.3, that operates only in time steps with $c = 1$ and takes advantage of the so-produced exact solutions on one dimensional segments.

4.2.2 Other Common Schemes

Now we present some of the most classical finite volume schemes, starting with the Lax-Friedrichs method. It has the form

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

The scheme can be written in conservative form with the numerical flux

$$F_{i+\frac{1}{2}}^{n,LF} = \frac{1}{2}\left(f(u_i^n + f(u_{i+1}^n))\right) - \frac{\Delta x}{2\Delta t}\left(u_{i+1}^n - u_i^n\right).$$

The Lax Friedrichs method is first order accurate, but it is known to be very diffusive and thus strongly smears out sharp gradients.

The last first order scheme we want to present is the Godunov scheme. It is based on solving Riemann problems, i.e. Cauchy problems of the form

$$u_t + f(u)_x = 0,\ u_0(x) = \begin{cases} u_l,\ x < 0 \\ u_r,\ x > 0 \end{cases} \tag{4.8}$$

at every interface. More precisely, the Godunov flux is computed as

$$F_{i+\frac{1}{2}}^n = \int_{t^n}^{t^{n+1}} f(u^{RP}(0,\tau))\mathrm{d}\tau,$$

where u^{RP} is the solution of the Riemann problem with $u_L = u_i^n, u_R = u_{i+1}^n$. For the advection equation, the Godunov scheme is identical to the upwind scheme, as the solution to the Riemann problem is the left or right state, depending on the sign of velocity.

4.2.3 High Order Methods

All the schemes presented before are first order accurate. Their numerical diffusion leads to smeared out solutions and requires a very fine discretization in order to get good numerical results. In general, a high order method will achieve better results than first order ones with less degrees of freedom or coarser grids. There are many different approaches in achieving the higher order. There are two important classes: In discontinuous Galerkin methods, additional degrees of freedom are introduced in each cell, effectively giving a high order (polynomial) representation instead of point values. The other option is to use a polynomial reconstruction on a group of cells, so-called stencils, and continue the computation with those reconstructed functions.

When using a linear high order method, one notices spurious oscillations in the computed solutions near sharp gradients. This is due tho the well-known Gibbs phenomenon: When a function is interpolated by polynomials near a discontinuity, the approximation will overshoot the function. The overshoots produced by those high order schemes lead to a growth in the total variation of the discrete solution and introduce new extrema. Furthermore, refining the grid will leave the magnitude of the overshoots untouched.

Theorem 4.2.1 ***Godunov (1959):***
Linear, monotonicity preserving schemes are at most of first order.

Proof. e.g. in [76]. □

The oscillations can only be avoided by introducing nonlinearities to the scheme [48]. The task of so-called limiters is to modify the numerical fluxes if they would introduce new oscillations to the solution. Usually, the numerical fluxes F^L of a first order method is combined with a higher order flux F^H

$$F_{i+\frac{1}{2}} = F^L_{i+\frac{1}{2}} + \phi_{i+\frac{1}{2}} \left(F^H_{i+\frac{1}{2}} - F^L_{i+\frac{1}{2}} \right). \tag{4.9}$$

The scaling factor ϕ depends on the data around the interface $i + \frac{1}{2}$. For $\phi = 0$, the scheme reduces to the first order method, while $\phi = 1$ represents the high order one. There are many different options how to choose ϕ, e.g.

- Beam-Warming: $\phi(\theta) = \theta$
- Fromm: $\phi(\theta) = \frac{1}{2}(1 + \theta)$
- minmod: $\phi(\theta) = \text{minmod}(1, \theta)$
- superbee: $\phi(\theta) = \max(0, \min(1, 2\theta), \min(2, \theta))$

to name a few, where $\theta_{i+\frac{1}{2}} = \frac{u_i - u_{i-1}}{u_{i+1} - u_i}$ is a comparison of neighboring slopes. In the explicit case, we mainly focus on WENO reconstructions, introduced in the next section. By construction they are nonlinear and reduce the amount of oscillations naturally. More details on limiting for implicit methods are given in Chapter 5.

In the following, we will give a brief insight on some different high order schemes, the ADER schemes, the active flux scheme and the discontinuous Galerkin schemes.

4.2.4 ADER Schemes

A large class of high order schemes for hyperbolic conservation laws are the so-called ADER schemes (Arbitrary DERivative schemes) [69],[70]. They are an extension of the Godunov scheme and can be formulated for any desired order. The basic idea is the use of high order WENO (weighted essential non-oscillatory) reconstruction and the solution of a higher order Riemann problem for the computation of the fluxes.

WENO Reconstruction

The high order spatial representation of the solution by WENO methods was introduced by Shu and Osher [66],[65]. When a reconstruction polynomial of cell u_i for a given

order k has to be computed, there are several options to choose the interpolation stencil. For $r \in \{0, \dots, k-1\}$ we define

$$S_i^r = \{I_{i-r}, \dots, I_{i-r+k-1}\}.$$

For each stencil S_i^r we compute the unique polynomial p_r sharing the same averages in the covered cells as u^n. Then the evaluation $u^r_{i+\frac{1}{2}} = p_r(x_{i+\frac{1}{2}})$ provides an approximation of u at the boundary that is kth order accurate. Furthermore, the evaluation of the derivatives of the polynomial $p_r^{(l)} = \frac{\partial^l p_r}{\partial x^l},\ l = 1, \cdots k-1$ is similar.

Since polynomial interpolation and evaluation are linear operations, we can write this in the form

$$\begin{pmatrix} p_r^{(0)}(x_{i+\frac{1}{2}}) \\ \vdots \\ p_r^{(k-1)}(x_{i+\frac{1}{2}}) \end{pmatrix} = \operatorname{diag}(1, \frac{1}{\Delta x}, \dots, \frac{1}{\Delta x^{k-1}}) \cdot C_{r,k} \cdot \begin{pmatrix} u_{i-r} \\ \vdots \\ u_{i-r+k-1} \end{pmatrix},$$

where $C_{r,k}$ is a coefficient matrix independent of the data.

Applied to smooth data, this approximation produces the expected good result, but will oscillate near sharp gradients. The important step of the so-called ENO (essentially non oscillatory) scheme is the selection of the stencil $S_i^r,\ r = 0, \dots, k-1$, that introduces the least oscillations. In practice, the stencils are shifted such that the discontinuity is excluded from the interpolation data and thus each polynomial remains well behaved.

When selecting one of the k stencils, a lot of the computed reconstructions are discarded after their smoothness indicators were computed. This is why in the weighted ENO (WENO), one uses a linear combination of all the data computed in order to get an even better approximation. The combined stencil over all S_i^r contains $2k-1$ different cells, which is the maximal order of accuracy that can be achieved. There are linear weights fulfilling

$$u_{i+\frac{1}{2}} = \sum_{r=0}^{k-1} d_r u^{(r)}_{i+\frac{1}{2}} = u(x_{i+\frac{1}{2}}) + O(\Delta x^{2k-1}).$$

The WENO scheme uses modified weights $w_r \approx d_r$ for smooth p_r and $w_r \approx 0$ if p_r includes a jump. Those modifications are controlled by the discrete smoothness indicators

$$\beta_r = \sum_{l=1}^{k-1} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \Delta x^{2l-1} \left(\frac{\partial^l p_r(x)}{\partial^l x} \right)^2 \mathrm{d}x \tag{4.10}$$

via

$$w_r = \frac{\alpha_r}{\sum_{l=0}^{k-1} \alpha_l}, \quad \alpha_r = \frac{d_{r,k}}{(\epsilon + \beta_r)^z},$$

with a small parameter ϵ avoiding division by zero. Altogether, the WENO reconstruction can be written as

Algorithm 4.2.2 *WENO reconstruction:*
For $r = 0, \dots, k-1$*:*

$$\begin{pmatrix} p_r^{(0)} \\ \vdots \\ p_r^{(k-1)} \end{pmatrix} = diag(1, \frac{1}{\Delta x}, \dots, \frac{1}{\Delta x^{k-1}}) \cdot C_{r,k} \cdot \begin{pmatrix} u_{i-r} \\ \vdots \\ u_{i-r+k-1} \end{pmatrix},$$

$$\beta_r = (u_{i-r} \quad \cdots \quad u_{i-r+k-1})\, B_{r,k} \begin{pmatrix} u_{i-r} \\ \cdots \\ u_{i-r+k-1} \end{pmatrix}$$

$$\alpha_r = \frac{d_{r,k}}{(\epsilon + \beta_r)^2}$$

Endfor

$$w_r = \frac{\alpha_r}{\sum_{l=0}^{k-1} \alpha_l}, \qquad r = 0, \dots, k-1$$

$$\begin{pmatrix} u_{i+\frac{1}{2}}^{(0)} \\ \vdots \\ u_{i+\frac{1}{2}}^{(k-1)} \end{pmatrix} = \begin{pmatrix} p_0^{(0)} & \cdots & p_{k-1}^{(0)} \\ \vdots & \ddots & \vdots \\ p_0^{(k-1)} & \cdots & p_{k-1}^{(k-1)} \end{pmatrix} \begin{pmatrix} w_0 \\ \vdots \\ w_{k-1} \end{pmatrix}$$

It generates an approximation of the function value and the first $k-1$ derivatives at the interface.

This procedure works really well inside the computational domain, but fails at the boundary. There, only one of the stencils completely lies in the domain. In [68], Shu and Tan introduce a variation of the WENO method for the boundary with stencils of different sizes. The general procedure is the same as in Algorithm 4.2.2, but one uses the matrices corresponding to different order approximations $C_{r,r}, B_{r,r}, d_{r,r}$, for $r = 0, \dots, k-1$. Furthermore, in [40] an improved formulation for the estimate of β_0 was proposed, allowing for scaling invariant results of the reconstruction. The main issue is, that the smoothness indicator β_0 for the constant polynomial is zero, which leads to relatively large contributions in the reconstruction in any case. The choice $\beta_0 = \frac{1}{\Delta x^2}$ leads to correct behaviour for the limit $\Delta x \to 0$. However, the author of [40] shows that this choice leads to reconstructions that are not scaling-invariant. Instead they propose the choice

$$\beta_0 = \beta_1 \left(\frac{\beta_1}{\beta_2}\right)^2, \tag{4.11}$$

where the smoothness parameter for the constant polynomial depends on the quotient $\frac{\beta_1}{\beta_2}$ of first and second order smoothness indicators. This yields the correct behaviour in the limit $\Delta x \to 0$. Furthermore, if there is a discontinuity within the first three cells, usually $\beta_2 >> \beta_1$ and thus β_0 will be very small, dominating the reconstruction. The author shows that the choice (4.11) leads to scaling invariant reconstructions.

Generalized Riemann Problem

We already introduced the Riemann problem in (4.8) for the formulation of the Godunov scheme. Due to the derivative information from the reconstruction, ADER schemes deal with the so called Generalized Riemann problem (GRP):

$$u_t + f(u)_x = 0,\ u_0(x) = \begin{cases} u_L(x),\ x < 0 \\ u_R(x),\ x > 0 \end{cases}, \tag{4.12}$$

where u_L and u_R are polynomials of degree $k-1$. A simple way to appropriately solve (4.12) is to first solve the classical RP with $u_L = u_L(0), u_R = u_R(0)$ and then successively evaluating the higher order classical Riemann problems, linearized at the solution of the first one:

$$(\partial_x^r u)_t + (D_f(u^0)\partial_x^k u)_x = 0,\ \partial_x^k u(x,0) = \begin{cases} u_L^{(r)}(x),\ x < 0 \\ u_R^{(r)}(x),\ x > 0 \end{cases}.$$

Cauchy-Kowalewskaya Procedure

From the solution of the GRP we get a polynomial representation of the solution in space. However,for the computation of the numerical flux a representation as function in time is required. The Cauchy-Kowalewskaya (CK) procedure (sometimes Lax-Wendroff procedure) is an algorithm for rewriting time derivatives in terms of space derivatives or vice versa, by iteratively applying derivatives to the original PDE. For the linear advection this is straight forward and leads to

$$\partial_t^r u = (-v)^r \partial_x^r u.$$

For nonlinear problems, the CK procedure is more involved and leads to increased complexity. Another option that gained more attention recently is the use of a local space-time DG (LSTDG) predictor to completely avoid the CK procedure [27],[79],[29]. In the case of pure advection, the CK procedure is trivial, thus it does not cause any problems.

Now, we have all the tools to formulate the ADER scheme for the advection equation:

Algorithm 4.2.3 *ADER scheme of order k:*

1. *Compute the WENO reconstructions.*
2. *Find the solution of the GRP*

$$u(t+\tau, x_{i+\frac{1}{2}}) = u(t, x_{i+\frac{1}{2}}) + \sum_{n=1}^{k-1} \frac{(\tau)^n}{n!} \partial_x^{(n)} u(t, x_{i+\frac{1}{2}}).$$

3. Insert this solution into (4.4) *by transforming time to space derivatives*

$$\begin{aligned} f_{i+\frac{1}{2}} &= \frac{1}{\Delta t}\int_0^{\Delta t} v\left(u(t,x_{i+\frac{1}{2}}) + \sum_{n=1}^{k-1}\frac{(-v\tau)^n}{n!}\partial_x^{(n)}u(t,x_{i+\frac{1}{2}})\right)d\tau \\ &= vu(t,x_{i+\frac{1}{2}}) + v\sum_{n=1}^{k-1}\frac{(-v\Delta t)^n}{(n+1)!}\partial_x^{(n)}u(t,x_{i+\frac{1}{2}}). \end{aligned}$$

4. Perform the FV update

$$u_i^{n+1} = u_i^n + \frac{\Delta t}{\Delta x}\left(f_{i+\frac{1}{2}} - f_{i-\frac{1}{2}}\right).$$

4.2.5 FV Methods on Advection Networks

The methods so far have been presented in their one dimensional scalar form. In order to create a scheme for the whole advection network, suitable numerical coupling conditions have to be applied. Different options to do so are presented in [11, 53, 12, 45]. We recall the analytical coupling condition (2.17) on the network of Section 2.2.3:

$$\left.\begin{aligned} \sum_{J_i\in\mathcal{I}(V)} A^i|v^i(t)|e^i(c^i(t),t) &= \sum_{J_i\in\mathcal{O}(V)} A^i|v^i(t)|e^V(t) \\ e^j(c_\neg^j(t),t) &= e^V(t) \end{aligned}\right\} \forall V\in\mathcal{V}, \forall J_j\in\mathcal{O}(V). \tag{4.13}$$

On the discrete level, as motivated in Section 2.3, we assume the velocities to be piece wise constant. This means for the formulation of a single step, we can assume the edge orientations to be directed into the node for $\mathcal{I}(V)$ and out of the node for $\mathcal{O}(V)$. The node temperature is computed as

$$e^V = \frac{\sum_{J_i\in\mathcal{I}(V)} A^i v^i(t) e^i(1,t)}{\sum_{J_i\in\mathcal{O}(V)} A^i v^i(t)}, \tag{4.14}$$

and serves as boundary condition for all outgoing edges. Furthermore, for a high order scheme we require the algebraic mixing for all higher temporal derivatives. This is obtained by taking the temporal derivative of (4.13) up to the given order:

$$\begin{aligned} \sum_{J_i\in\mathcal{I}(V)} A^i v^i(t)\partial_t^r e^i(1,t) &= \sum_{J_i\in\mathcal{O}(V)} A^i v^i(t)\partial_t^r e^V(t) \\ \partial_t^r e^j(0,t) &= \partial_t^r e^V(t) \qquad \forall r=0,\dots,k-1,\ \forall V\in\mathcal{V}, \forall J_i\in\mathcal{O}(V). \end{aligned} \tag{4.15}$$

This allows us to maintain the high order of the ADER scheme in the network case.

Due to the linearity of the coupling, we can directly formulate the coupling in terms of the numerical fluxes. For this step, we have to assume constant velocities in time.

Recall that for a scheme of order k, we have a numerical flux fulfilling

$$\begin{aligned} F^n_{i+\frac{1}{2}} &= \frac{1}{\Delta t}\int_{t^n}^{t^{n+1}} f(e(x_{i+\frac{1}{2}},\tau))\mathrm{d}\tau + \mathcal{O}(\Delta x^k) \\ &= \frac{1}{\Delta t}\int_{t^n}^{t^{n+1}} ve(x_{i+\frac{1}{2}},\tau)\mathrm{d}\tau + \mathcal{O}(\Delta x^k). \end{aligned}$$

The numerical flux at the first interface of a pipe, namely at the node, is consequently

$$F^n_0 = \frac{1}{\Delta t}\int_{t^n}^{t^{n+1}} ve^V(\tau)\mathrm{d}\tau + \mathcal{O}(\Delta x^k).$$

Inserting (4.14) leads to

$$\begin{aligned} F^n_0 &= \frac{1}{\Delta t}\int_{t^n}^{t^{n+1}} v\frac{\sum_{J_i\in\mathcal{I}(V)} A^i v^i e^i(1,\tau)}{\sum_{J_j\in\mathcal{O}(V)} A^j v^j}\mathrm{d}\tau + \mathcal{O}(\Delta x^k) \\ &= \frac{1}{\Delta t} v\frac{1}{\sum_{J_j\in\mathcal{O}(V)} A^j v^j}\sum_{J_i\in\mathcal{I}(V)} A^i \Delta t F^n_{1,i} + \mathcal{O}(\Delta x^k). \end{aligned} \tag{4.16}$$

Here, we denote the rightmost flux of an edge J_i incoming node V with $F^n_{1,i}$ and assume the orientation of the edges to be in line with the flow direction. As we can see, the numerical flux at the first interface after the node is a weighted average of all incoming fluxes at the node. This way, we can couple the edges directly via their fluxes at the boundaries.

When lifting the scheme to the full network, one important point to consider is the CFL condition. Applying a global time step Δt_{net} to the full system, all individual stability conditions have to be fulfilled simultaneously. Consequently we have to choose

$$\Delta t^n_{net} = \min_{J_i\in\mathcal{J}}\{\Delta t^n_i\},$$

where Δt_i is the time step of edge J_i. Depending on the concrete discretization setting, this global time step might be much smaller than the local stability condition would require. This can lead to large numerical diffusion, especially for low order methods. If the network flow velocity v is constant in time, we can compensate this effect by choosing non uniform discretization widths Δx_i for $J_i \in \mathcal{J}$ [58]. The total number of degrees of freedom can be distributed to the edges such that

$$\max_i\{\frac{\Delta t_i}{\Delta t_{net}}\} \to \min.$$

However, if the flow velocity is time dependent, a constant mesh is not able to limit the deviations in local CFL numbers.

Another major drawback of the global time step is the increasing computational cost. One single pipeline with short length but fast flow velocity can reduce the global time step of the network by orders of magnitude and increase the computation time of the full system by around the same factor. This is why we investigate a way to decouple the time steps of the single pipes in Section 4.3.

4.2.6 Active Flux Scheme

A scheme that currently gains some attention is the so-called active flux scheme [31]. It is a recent modification of van Leer's scheme V [73]. In contrast to classical finite volume schemes, additional point values at the interfaces are introduced. For this section we therefore indicate the cell averages as $\bar{u}_i$ in order to distinguish them from the point values $u_{i+\frac{1}{2}}$.

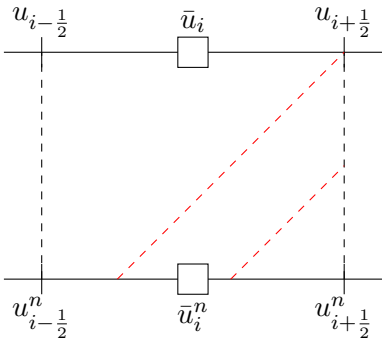

Figure 4.2: Characteristics in the active flux scheme

In contrast to staggered grid ansatzes, the locations of the different discretization parts do not change.
For the update of the point values, a quadratic reconstruction is performed. So we find a polynomial p with

$$p(x_{i-\frac{1}{2}}) = u^n_{i-\frac{1}{2}}, \quad p(x_{i+\frac{1}{2}}) = u^n_{i+\frac{1}{2}}, \quad \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} p(x)\mathrm{d}x = \bar{u}^n_i. \tag{4.17}$$

This is serving as the initial condition of a Cauchy problem. The new interface value is just the solution of the IVP evaluated at $(x_{i+\frac{1}{2}}, t^{n+1})$, comp Figure 4.2. The numerical flux for the cell update is computed with the Simpson rule

$$F_{i+\frac{1}{2}} = \frac{1}{6}\left(u^n_{i+\frac{1}{2}} + 4u^{n+\frac{1}{2}}_{i+\frac{1}{2}} + u^{n+1}_{i+\frac{1}{2}}\right),$$

where the solution at the interface has to be evaluated also at $t^{n+\frac{1}{2}} = t^n + \frac{\Delta t}{2}$. Altogether this results in a third order FV scheme.

A limiting of the active flux scheme could be done in two different ways. One is to use classical flux limiting techniques in the finite volume update, but that would not influence the values at the interfaces. A better option is to modify the reconstruction. There have been several suggestions in [6],[19], where the quadratic polynomial is replaced by a different function fulfilling (4.17).

4.2.7 Discontinuous Galerkin Methods

A different ansatz to solve hyperbolic PDEs than the finite volume context are discontinuous Galerkin (DG) schemes [56]. Instead of an integral formulation, they are based on a weak formulation of the underlying equations. In this work we will mainly focus on FV methods. For the sake of completeness we present the general principles of Discontinuous Galerkin schemes in the following.
Starting with the conservation law

$$\begin{aligned} u_t + f(u)_x &= 0 \\ u(x,0) &= u_0(x), \end{aligned}$$

we multiply with sufficiently piecewise smooth test functions ϕ and integrate over an Interval $I_i = [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}]$. Integration by parts leads to

$$\begin{aligned} &\int_{I_i} u_t(x,t)\phi(x)dx - \int_{I_i} f(u(x,t))\phi_x(x)dx \\ &+ f(u(x_{i+\frac{1}{2}},t))\phi(x_{i+\frac{1}{2}}) - f(u(x_{i+\frac{1}{2}},t))\phi(x_{i+\frac{1}{2}}) = 0 \\ &\int_{I_i} u(x,0)\phi(x)dx = \int_{I_i} u_0(x)\phi(x)dx. \end{aligned}$$

In classical finite element methods, the test functions ϕ are chosen such that they vanish at the boundary, eliminating the flux terms. For hyperbolic conservation laws, we need to resolve possibly discontinuous solutions, thus the flux terms are important. We approximate the solution u and the test functions ϕ by functions from the space

$$P^k(I_i) = \{\phi \in L^1(I_i) | \phi \text{ is a polynomial of degree} \leq k\}.$$

Using the Legendre polynomials ϕ_l as a basis of P^k, we express the approximation of u as

$$u_i(x,t) = \sum_{l=0}^{k} u_i^l \phi_l(x)$$

and get

$$\begin{aligned} \frac{1}{2l+1}\partial_t u_l(x,t) - \frac{1}{\Delta x}\int_{I_i} f(u_i(x,t))\partial_x\phi_l(x)dx& \\ + \frac{1}{\Delta x}\{f_{i+\frac{1}{2}} - (-1)^l f_{i-\frac{1}{2}}\} = 0 \qquad & \forall l = 0,\dots,k \qquad (4.18) \\ u_i^l(0) = \frac{2l+1}{\Delta x}\int_{I_i} u_0(x)\phi_l(x)dx.& \end{aligned}$$

Altogether, we can write (4.18) as an ODE

$$\begin{aligned} \dot{u}_h &= L(u_h) \\ u_h(0) &= u_0, \end{aligned}$$

with a vector u_h containing all $u_j^l,\ j = 1, \dots, n\ \ l = 1, \dots, k$. The ODE can be integrated by e.g. explicit/implicit Runge-Kutta schemes. When explicit time integration schemes are used, the CFL condition for a DG scheme of order k is

$$v\frac{\Delta t}{\Delta x} \leq \frac{1}{2k+1}.$$

Employing implicit time integration, this restriction is avoided. However, also implicit time integration are subject to stability bounds when applied to DG schemes. Altogether, the general behavior of DG schemes is similar to high order finite volume schemes in our context. This is why we focus on formulations in terms of the latter in this work.

4.3 Local Time Stepping (LTS) Scheme

In this section, a new local time stepping (LTS) scheme for transport networks is presented. In the description of the scheme we closely follow the formulations from [10], where we already published the method. The main idea of local time stepping is the choice of individual, optimal time steps for different parts of the computational domain. This has already been done in the context of ADER-DG schemes for elastic waves in [28] and blood flow models in [54]. The special structure of our problem enables us to extremely reduce the computational effort by application of the local time steps. Furthermore, we incorporate high order coupling conditions at the nodes that allow for global high order accuracy on the network setting.

For every edge in the network, we choose the local time step Δt_n such that

$$\Delta t_n v = \Delta x. \tag{4.19}$$

This means the CFL number is exactly equal to 1 for every time step and every edge. Since we deal with linear advection, in such a time step the information travels exactly one cell in flow direction, while its values remain unchanged. Consequently, the use of an upwind discretization will produce exact results on each pipe, as mentioned in section Section 4.2.1.

Important to note here is that an edge can only be updated, if the boundary data is determined. Consider a node V and a set of connected edges starting from a common time level, all with different local time steps Δt_i. Then we know that information can travel in edge J_i by exactly Δx_i during the time step Δt_i. This means we can evaluate the numerical flux over the node only for the intervals, for which the data corresponding to all connected edges is known. For every node, only the edge with the smallest Δt_i can be updated. This condition holds true for all nodes in the network, in particular for starting and end node of a given edge. Therefore, the update condition for an edge $J_i = (V_{i_1}, V_{i_2})$ is

$$\Delta t_i = \min_{J_j \in \mathcal{A}(V_{i_1}) \cup \mathcal{A}(V_{i_2})} \{\Delta t_j\}. \tag{4.20}$$

The different time step sizes inevitably leads to desynchronized time levels inside the network. If there is no common starting time for neighboring edges, it is important that the one with the lowest new time level is updated first. Then the update condition is:

$$t_i + \Delta t_i = \min_{J_j \in \mathcal{A}(V_{i_1}) \cup \mathcal{A}(V_{i_2})} \{t_j + \Delta t_j\}. \tag{4.21}$$

Theorem 4.3.1 *Let $t_i \in \mathbb{R}\ \forall J_i \in \mathcal{J}_P$ and $\Delta t_i \in \mathbb{R}\ \forall J_i \in \mathcal{J}$ be the current time levels and time steps for all edges.*
Then there exists at least one pipe fulfilling (4.21).

Proof. The set of all $t_i + \Delta t_i, i = 1, \ldots, N_{\mathcal{J}_P}$ has a minimum. The corresponding edge obviously also fulfills the local condition. □

Decoupling the time steps leads to numerical fluxes that are composed of multiple terms, each contributing a different time subinterval to the full flux. Assume a 3-junction V with two incoming edges J_1, J_2 and one outgoing edge J_3, all with cross section area $A_i = 1$. Starting from the same time level t, all three edges have different local time steps, with $\Delta t_1 < \Delta t_2 < \Delta t_3 < 2\Delta t_1$. The edges are discretized in finite volumes with $N_j, j \in \{1, 2, 3\}$ grid cells. Each edge computes its solution of the new time step with the upwind scheme (4.6). Due to the special choice of the time steps, this means

$$u_i^{n+1} = u_{i-1}^n$$

and the numerical flux at the boundary $F_{1,j}^{t,\Delta t} = v_j u_{N_j}^n$, which is valid for the time interval $[t, t + \Delta t]$.

As we have seen in (4.16), the flux over the first interface of J_3 can be expressed in terms of all outgoing fluxes of J_1 and J_2. The important difference in local time stepping is that the fluxes on the right hand side do not cover the same common time interval. Instead, coming back to our three way junction example, we can write the flux of J_3 as

$$\begin{aligned} F_{0,3}^{t,\Delta t_3} = \frac{1}{\Delta t_3} \Big(& \Delta t_1 F_{1,1}^{t,\Delta t_1} + (\Delta t_3 - \Delta t_1) F_{1,1}^{t+\Delta t_1, \Delta t_3 - \Delta t_1} \\ & + \Delta t_2 F_{1,2}^{t,\Delta t_2} + (\Delta t_3 - \Delta t_2) F_{1,2}^{t+\Delta t_2, \Delta t_3 - \Delta t_2} \Big). \end{aligned} \tag{4.22}$$

We can see that there are several contributions, each belonging to different time intervals. In order to collect all those terms, we introduce memory variables $\mathcal{MV}_i = 0$ for each edge. Every time an adjacent edge updates, the memory variables get filled with the corresponding fluxes. The first time step in this example will be done by edge J_1, so we update the cell values and increase the memory variable for J_3 by

$$\mathcal{MV}_3 \mathrel{+}= \Delta t_1 F_{1,1}^{t,\Delta t_1}.$$

Next, J_2 updates and again increments the memory variable for the adjacent J_3

$$\mathcal{MV}_3 \mathrel{+}= \Delta t_2 F_{1,2}^{t,\Delta t_2}.$$

Finally, the update criterion for J_3 is fulfilled and only the remaining terms for the flux have to be computed. As we know for J_1 and J_2, in the remaining sub intervals only information from the last cell reaches the boundary. Therefore those terms are computed as

$$\mathcal{MV}_3 \mathrel{+}= (\Delta t_3 - \Delta t_1) v_1 u_{1,N_1}^{t+\Delta t_1} + (\Delta t_3 - \Delta t_2) v_2 u_{2,N_2}^{t+\Delta t_2}.$$

Finally, the FV step can be performed, where for the first cell in J_3 we have

$$\begin{aligned} u_{3,1}^{t+\Delta t_3} &= u_{3,1}^t - \frac{\Delta t_3}{\Delta x_3} \left(v_3 u_{3,1}^t - F_{0,3}^{t,\Delta t_3} \right) \\ &= \frac{\Delta t_3}{\Delta x_3} F_{0,3}^{t,\Delta t_3} \\ &= \frac{1}{\Delta x_3} \mathcal{MV}_3 \end{aligned} \tag{4.23}$$

so it will solely be filled with information from the memory variable. Each other cell value will be shifted exactly one cell to the right and the last cell fills the memory variables of the outgoing edges of the other node in the same way. After the edge update of J_3 we set $\mathcal{MV} = 0$. This procedure is continued for the whole network until all edges reach the final time.

The full computational algorithm can be summarized as

Algorithm 4.3.2 ***LTS scheme***

1. *Initialize all edges $J_i \in \mathcal{J}$ with $v_i, \Delta x_i, t_i$, compute all local time steps Δt_i as in* (4.19)*, initialize $\mathcal{MV}_i = 0, \forall J_i \in \mathcal{J}$*
2. *Compute the set of edges $\mathcal{U} \subset \mathcal{J}$ that fulfills* (4.20)
3. *While $\mathcal{U} \neq \emptyset$:*
   - *Extract $J_i = (V_1, V_2) \in \mathcal{U}$, $\mathcal{U} = \mathcal{U} \setminus \{J_i\}$.*
   - *Compute the missing fluxes over V_1 according to*

$$\mathcal{MV}_i \mathrel{+}= \sum_{J_j \in \mathcal{I}(V_1)} (t_i + \Delta t_i - t_j) v_j u_{j,N_j}^{t_j}.$$

   - *For all $J_j \in \mathcal{O}(V_2)$ increment the memory variables:*

$$\mathcal{MV}_j \mathrel{+}= \Delta t_i v_i u_{i,N_i}^{t_i+\Delta t_i}.$$

   - *Update the cell values as*

$$(u_{i,0}^{t_i+\Delta t_i}, u_{i,1}^{t_i+\Delta t_i}, \ldots, u_{i,N_i}^{t_i+\Delta t_i}) = (\frac{1}{\Delta x_i} \mathcal{MV}_i, u_{i,0}^{t_i}, \ldots, u_{i,N_i-1}^{t_i}).$$

- *Set* $t_i = t_i + \Delta t_i$, $\Delta t_i = \min\{\frac{\Delta x_i}{v_i}, t_{end} - t_i\}$, $\mathcal{MV}_i = 0$.
- *For all* $J_j \in \mathcal{A}(V_1) \cup \mathcal{A}(V_2)$
  - *If* Δt_j *fulfills* (4.20) *and* $t_j < t_{end}$ *set*

$$\mathcal{U} = \mathcal{U} \cup \{J_j\}.$$

End while.

4. *Return solution.*

An exemplary scenario is illustrated in Figure 4.3, compare [10]. There are three edges with corresponding time steps $\Delta t_1 = 1, \Delta t_2 = 0.5, \Delta t_3 = \frac{2}{3}$ as in (A). The vertical axis represents the time elapsed, while for each edge the current time step is drawn in light gray, already performed ones are dark gray. The first edge fulfilling (4.20) is J_2. The memory variables for the corresponding time step are computed and J_2 can be updated, see (B). Next, J_3 fulfills the condition and thus only needs to compute the flux for the remaining interval $[0.5, \frac{2}{3}]$, comp (C). After that, J_1 and J_2 simultaneously can be updated to their respective next time level $t_i = 1$ as in (D). This procedure continues until all edges reach the final time $t_{end} = 2$, (F)-(G).

It is important to note, that in general the final time steps of each edge will not exactly reach t_{end}. Therefore, the last time step has to be performed with a classical upwind step covering the remaining time distance. This step might introduce some slight numerical diffusion on each pipe, but since this happens just in one single step, the effects are marginal. More importantly, on each time step other than the last one, exact solutions inside each pipe are computed. This means no numerical diffusion while the information is transported along the pipes. The only source of numerical errors is in the first order coupling at the nodes, and the very last time step. Altogether this leads to a first order convergent scheme, that nevertheless has total errors much smaller than the ones produces by pure upwind schemes with global time stepping. A comparison of the computation errors are presented in Chapter 7.

Another big advantage of the presented scheme is the efficient computation it enables. The data inside each pipe does is not computed for an update step, but it is just shifted into the direction of flow. The relevant computations all take place at cells adjacent to a node. Therefore, the computational complexity does not increase, if the spatial discretization is refined. To be precise, the number of operations in each time step is constant, independent of the number of grid cells. This leads to a total computational complexity of $\mathcal{O}(\frac{1}{\Delta t})$ instead of $\mathcal{O}(\frac{1}{\Delta t^2})$ for classical FV schemes. Consequently, the LTS scheme outperforms classical schemes in terms of computation time in the asymptotic limit.

Next, we give several extensions to the basic LTS scheme in order to use it in the context of district heating networks and lift it to higher order accuracy.

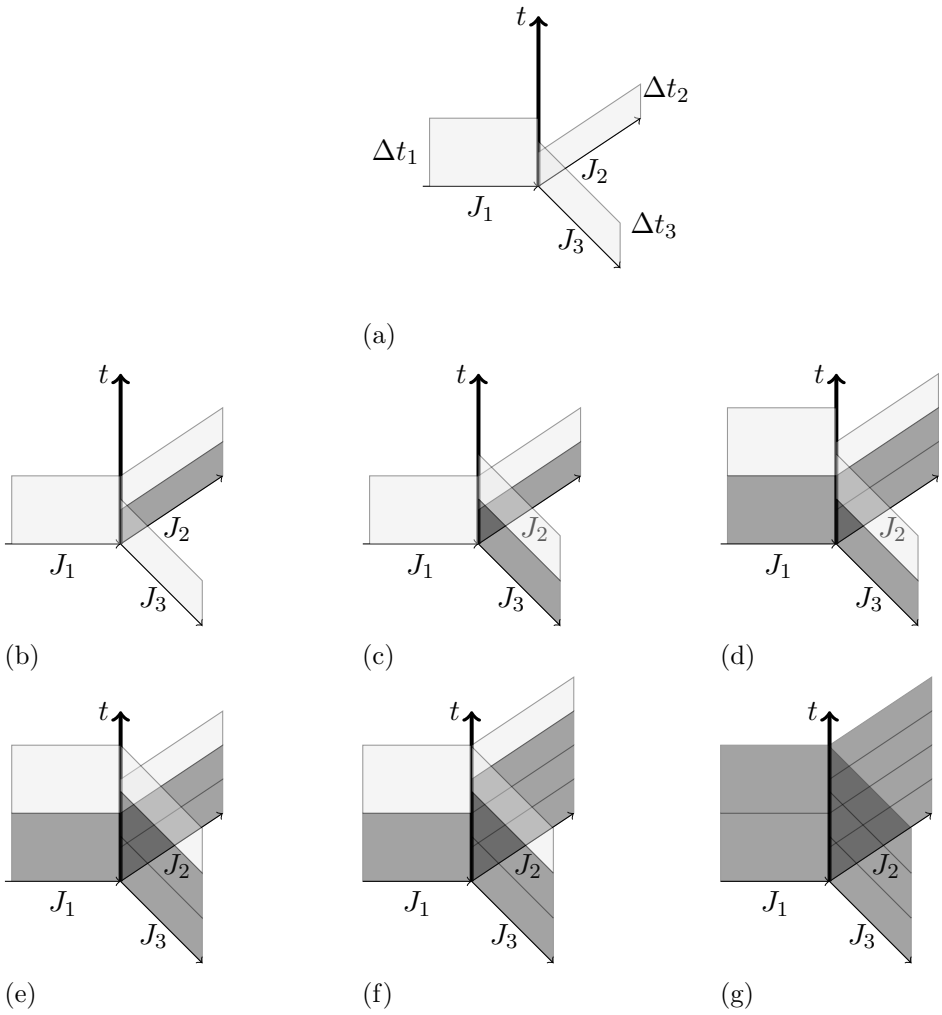

Figure 4.3: Illustration of the local time steps at a single node. The current time step size for each edge is pictured in light gray. Already computed time steps are drawn in dark gray.

4.3.1 High Order LTS Scheme

As mentioned before, the special choice of the time steps leads to exact results in the advective part in the pipelines. By simply using the high order coupling introduced in (4.15), we get global high order of the LTS scheme, under the assumption that also the last sub-steps will be computed with a scheme of the desired order. The only modifications necessary in algorithm 4.3.2 are the following: each time a numerical flux is computed and stored in a memory variable, instead of using the upwind flux, we use

the ADER flux

$$\Delta Q = \int_{t_i}^{t_i+\Delta t_i} v u_{i,N_i}^{t_i}(\tau) d\tau. \tag{4.24}$$

Note that, in contrast to the classical flux definition, we do not divide by Δt in the increments of $\mathcal{MV}$, compare the formulations in (4.22),(4.23). The new formula involves a WENO reconstruction $u_{i,N_i}^{t_i}(\tau)$ at the corresponding boundary and the integration of the resulting polynomial. Since this has to be done just for one interface each pipe, the computational advantage stays as presented in the previous paragraph.

Another option to further reduce the number of WENO reconstructions needed is the storage of derivative information in the cells. Assume for each cell there already exists a reconstruction polynomial. They are stored together with the cell averages in a vector $(u_i, \partial_t u_i, \dots, \partial_t^k u_i)$. Then in the update step of the pipe, we can directly calculate the numerical flux and shift the full vector of derivative information along the flow direction. When computing the new first cell value of a pipe in an update step, it is possible to directly retrieve a high order representation of the data.

From the computation of the numerical flux via the memory variables $\mathcal{MV}$, information of several different sub steps was computed due to the non synchronous updates of the neighboring edges. In case of just a single sub step, the higher order derivatives directly transfer to the new cell. On the other hand, if there are multiple subintervals, each with a high order representation of the solution, they have to be recombined to one single polynomial for the first cell of the edge, compare Figure 4.4. This is a com-

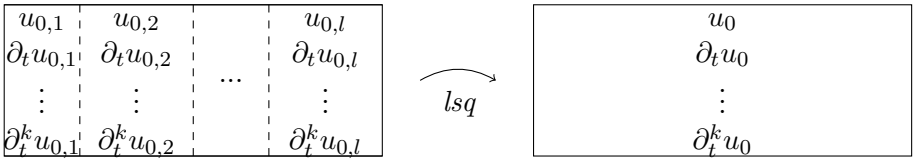

Figure 4.4: Remapping from subcell resolution to cell values at the left (upstream) boundary

mon problem in schemes with subcell resolution, as e.g. in [29]. The new polynomial reconstruction is the solution of a constrained least squares problem, since the arising linear system is overdetermined. The necessary constraint is conservation of the total mass from the subcell data to the reconstruction.
That way, the new cell value already contains all high order information. When it reaches the next node at the end of the pipe, no WENO reconstruction is needed. If the initial data consists of cell means as usual, the WENO step is only performed once for each cell to provide the necessary high order information. The computational effort of a WENO step should be similar to the remapping operator, so in the worst case scenario this does not improve computation speed. However usually the memory variables contain just one or two different segments of high order representation. In that case, the remapping is more efficient.

4.3.2 Source Term

The incorporation of a source term as in (2.22a) is straight forward. Note, that it is linear and not varying in space, thus an exact solution of the evolution along characteristics is possible. In each time step, after the cell values got shifted, they are modified according to the characteristics' ODE. This requires $\mathcal{O}(\frac{1}{\Delta x})$ operations instead of $\mathcal{O}(1)$, as discussed above. As long as the values are only needed at the boundary, it is not necessary to add the influence of the source each step, but only when the data reaches the last cell. Therefore we just advect the values along the pipes, while storing the time delay from entering to exiting the pipe. At the outflow, we apply the source term for the whole time span.

4.3.3 Time Varying Velocities

The general principle of the scheme described above can be applied as well to transport equations with time dependent velocities $v = v(t)$. In this case, we determine the local time steps with

$$\int_{t_n}^{t_n+\Delta t_n} v(\tau)\mathrm{d}\tau = \Delta x. \tag{4.25}$$

Assuming $v(t)$ is known and regular enough, we modify two steps in the algorithm. First, the transformation from a spatial representation to a temporal one in the Cauchy-Kowalewskaya procedure is modified to variable v. The resulting expressions are lenghty and can be done by symbolic calculation algorithms. Furthermore, in the computation of the numerical flux

$$\Delta Q = \int_{t_i}^{t_i+\Delta t_i} v(\tau) u_{i,N_i}^{t_i}(\tau), \tag{4.26}$$

the time dependence is taken into account, in contrast to (4.24).

However, from a computational point of view this ansatz is not applicable in most cases, when the velocity evaluation itself is part of the algorithm. From the used splitting and the numerical solver for the hydraulic equation we get a piece-wise constant approximation of v. We integrate the piece-wise definition of v into the algorithm by giving an additional necessary condition for the choice of pipes ready to be updated. Equation (4.21) changes to

$$t_i + \Delta t_i = \min\{\min_{J_j \in \mathcal{A}(V_{i_1}) \cup \mathcal{A}(V_{i_2})} \{t_j + \Delta t_j\}, t_v + \Delta t_v\}, \tag{4.27}$$

introducing an additional 'local' time t_v and time step Δt_v for the velocity field. Condition (4.27) ensures that for all steps we perform, we have a valid velocity. The hydraulic computations are done in Δt_v intervals, that are not necessarily coupled to the individual steps of the pipes. Whenever (4.27) produces an empty set of updatable pipes, the velocity is recomputed for the next time level. Note, that in this case we have

$$t_v + \Delta t_v \leq t_i + \Delta t_i, \qquad \forall J_i \in \mathcal{J}$$

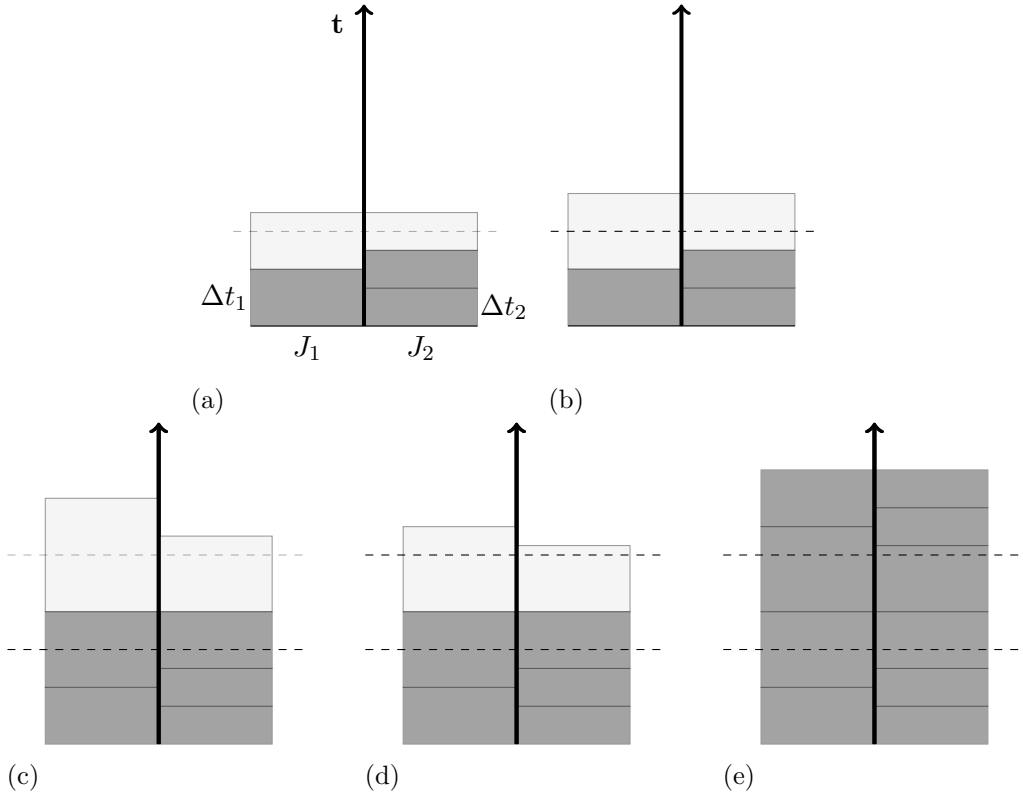

Figure 4.5: Update procedure of the LTS scheme with velocity recomputations for two neighboring edges. The current time step size for each edge is pictured in light gray. Already computed time steps are drawn in dark gray.

and consequently the boundary values at the consumers, necessary for the evaluation of the hydraulic equations, are known for the full interval $[t_v, t_v + \Delta t_v]$. Then the updates of the fluxes reduce to sub intervals $[t_i, t_i + \Delta t_i] = \cup_{j=1}^{n_{t_i}} I_j$ with piecewise constant velocities

$$\Delta Q = \int_{t_i}^{t_i+\Delta t_i} v(\tau) u_{i,N_i}^{t_i}(\tau) = \sum_{j=1}^{n_{t_i}} \int_{I_j} v_j u_{i,N_i}^{t_i}(\tau). \tag{4.28}$$

Usually, this expression will have $n_{t_i} = 1$ or $n_{t_i} = 2$ addends, but if for single pipes the flow velocity is small and consequently the local time step large, several velocity updates are possible in between two successive edge updates. Note, that each time the velocity is recomputed, all local time steps might change and have to be adapted. The corresponding update procedure is illustrated in Figure 4.5. We consider just a single node with two connected edges. Both edges start at $t_0 = 0$ and their time steps are $\Delta t_1 = 1.5$ and $\Delta t_2 = 1$. The time intervals for velocity recomputations is $\Delta t_v = 2.5$. The beginning is similar to the previous example. The initial velocity is valid until

$t = 2.5$ and each edge can update one and two times respectively. Then we reach the situation of the first picture, Figure 4.5a, where both next time levels would exceed the validity interval of the velocity. At that point, we recalculate v and assume for this example, that all velocities get multiplied by 0.5. This also affects the local time steps of both edges: The new time step of edge J_1 has to fulfill

$$\begin{aligned}(\Delta t_1 - 1)v_2 + 1 \cdot v_1 = \Delta x_1 = 1.5 \cdot v_1 \\ \Rightarrow \Delta t_1 = 2,\end{aligned}$$

since it applies the old v_1 for $t \in [1.5, 2.5]$ and the new v_2 from $t = 2.5$ onward. In the same way we get $\Delta t_2 = 1.5$, so both edges now have a different time step, compare Figure 4.5b. Furthermore, after those steps illustrated in Figure 4.5c, both edges apply velocity v_2 for the whole next interval, leading to $\Delta t_1 = 3, \Delta t_2 = 2$. Again, they would exceed the validity interval of v such that at $t = 5$ the velocity is recomputed again. We assume that for the new velocity we get $v_3 = v_1 = 2 \cdot v_2$ the initial values again. This change of velocity again influences the time steps of the edges. This procedure continues until the final time is reached. For this example, we get the intermediate time levels $\{0, 1.5, 3.5, 5.75, 7.25\}$ with time steps $\{1.5, 2, 2.25, 1.5\}$ for edge one and $\{0, 1, 2, 3.5, 5.25, 6.25, 7.25\}$ with time steps $\{1, 1, 1.5, 1.75, 1, 1\}$ for edge two. Note, that by the change of velocities, an interval can never shrink below $t_v + \Delta t_v$, since the new values only apply from that time on.

Chapter 5

Numerical Methods: Implicit Schemes

Explicit schemes, as presented in the previous chapter, can generally be written as an update of the form

$$u^{n+1} = \mathcal{F}(u^n), \tag{5.1}$$

where the solution at the new time level only depends on data from the old one. In general, explicit methods are easier to implement and each evaluation of (5.1) can be computed faster than the implicit counterpart

$$u^{n+1} = \mathcal{F}(u^n, u^{n+1}).$$

Implicit schemes involve the solution of a (possibly nonlinear) system. However, there is one big advantage of implicit methods: Some of them are not bound to a stability restriction concerning the possible time steps we can perform. Furthermore, for some special problems, implicit schemes prove to produce more accurate results, most notably in the case of stiff ODEs.

In this chapter we will present several implicit numerical schemes for the solution of transport problems. We start with the classical implicit upwind scheme and a similar first order characteristic scheme. Then, using a generic reconstruction approach a class of candidate finite volume schemes is constructed and investigated with respect to their stability properties. With the focus on purely upwind directed stencils, we find two high order schemes of fourth and third order, that show stability for arbitrary large time steps. In a second part, we review three different possibilities for limiters for implicit FV schemes. The special upwind structure of our problem enables the efficient implementation of an a posteriori limiting technique. Some of the sections in this chapter have already been submitted in similar form or in parts in Advances in Computational Mathematics (2021), a preprint version can be found in [30].

5.1 Implicit Upwind Scheme

We start by formulating the implicit upwind scheme. While for $v > 0$ the explicit upwind flux is $F_{i+\frac{1}{2}} = vu_i^n$, for implicit upwind it is

$$F_{i+\frac{1}{2}} = vu_i^{n+1},$$

leading to a full scheme of the form

$$u_i^{n+1} = u_i^n - \frac{\Delta t}{\Delta x}\left(vu_i^{n+1} - vu_{i-1}^{n+1}\right). \tag{5.2}$$

Instead of a direct evaluation of the solution at the new time level, a linear system has to be solved. It has the form

$$\begin{bmatrix} 1-c & c & & \\ & \ddots & \ddots & \\ & & 1-c & c \\ c & & & 1-c \end{bmatrix} u^{n+1} = u^n$$

for periodic boundary conditions, where $c = \frac{v\Delta t}{\Delta x}$ is the CFL number. Similarly to the explicit case, we can here aswell investigate the modified equation and find that the implicit upwind solves

$$u_t + vu_x = \frac{v\Delta x}{2}(1+c)u_{xx}$$

to second order accuracy and thus is stable for arbitrary CFL numbers $c > 0$. This is the big advantage compared to the explicit scheme, where for stability $c \leq 1$ is required. On the other hand, the diffusion coefficient is larger than for the explicit case, so in a time step regime, where $c < 1$ we can expect the explicit upwind scheme to be more accurate than the implicit one.

5.2 Characteristic Scheme

In [51], a slightly modified numerical method for the simulation of district heating networks was proposed. The characteristic scheme is a finite difference method based on tracing the characteristics leading to (x, t^{n+1}) to their origin point (x_0, t_0) which can be either at the boundary or at the previous time level, see Figure 5.1. If $c \leq 1$, the characteristics of x_i reach the previous time level, where a linear interpolation is used to approximate the value at the desired origin point. This means we calculate

$$u(x_i, t^{n+1}) = u(x_i - \Delta t v, t^n) = u(x_i, t^n) - c\left(u(x_i, t^n) - u(x_{i-1}, t^n)\right),$$

which is exactly the explicit upwind scheme.
On the other hand if $c > 1$, we get

$$u(x_i, t^{n+1}) = \frac{1}{c}u(x_{i-1}, t^n) + (1 - \frac{1}{c})u(x_{i-1}, t^{n+1}), \tag{5.3}$$

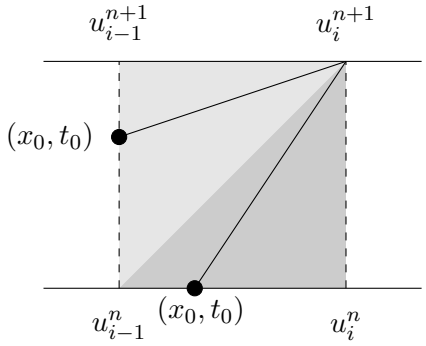

Figure 5.1: Characteristic scheme: Origin points of the characteristics in the explicit (dark gray) and implicit (light gray) case

which seems very similar to the implicit upwind scheme. Inserting a Taylor approximation, we get the modified equation of the characteristic scheme as

$$u_t + vu_x = \frac{v\Delta x}{2}(c-1)u_{xx}.$$

We can see, that it has slightly smaller numerical diffusion than implicit upwind, at the cost of the stability condition $c \geq 1$. Furthermore, scheme (5.3) does not have a conservative form as FV schemes do. However, in the case of periodic boundary conditions we can show that the total mass is conserved

$$\begin{aligned}\sum_{i=0}^{k} u_i^{n+1} &= \frac{1}{c}\sum_{i=-1}^{k-1} u_i^n + (1-\frac{1}{c})\sum_{i=-1}^{k-1} u_i^{n+1} = \frac{1}{c}\sum_{i=0}^{k} u_i^n + (1-\frac{1}{c})\sum_{i=0}^{k} u_i^{n+1} \\ &= \sum_{j=0}^{\infty}\left[(1-\frac{1}{c})^j \frac{1}{c}\sum_{i=0}^{k} u_i^n\right] = \left(\sum_{i=0}^{k} u_i^n\right)\frac{1}{c}\left(\frac{1}{1-(1-\frac{1}{c})}\right) \\ &= \sum_{i=0}^{k} u_i^n.\end{aligned}$$

Due to the conditional formulation of the general characteristic scheme, we can improve the explicit case by a Lagrangian ansatz on the pipes. Consider a pipe of length $L = 1$ and a discretization with n cells of width $\Delta x = \frac{1}{n}$. Tracing the characteristics of cell i, they reach the initial condition inside the whole pipe if $v\frac{\Delta t}{dx} < i$. We compute the linear interpolation of the two cell values neighboring the origin point of the characteristics

$$u(x_i, t^{n+1}) = u(x_i - \Delta t v, t^n) = u(x_j, t^n) - c\left(u(x_j, t^n) - u(x_{j-1}, t^n)\right),$$

for

$$j - 1 \leq v\frac{\Delta t}{dx} \leq j, \qquad j \in \{1, \ldots, i\}.$$

In those cases, the numerical solution will be much more accurate than a classical upwind discretization. In fact, it will behave as an exact transport for $j-1$ cells and an upwind step just for the remaining fraction of Δx. On the other hand, multiple cells will retrieve their data from the same linear interpolation of the boundary, effectively reducing the information they store. All cell values $u_i^{n+1}, i = 0, \dots, j-1$ take their values from the evaluation of the same linear function. Consequently, while the Lagrangian ansatz might be beneficial in general, it is not in this case even for large CFL numbers. Most of the pipe data will be linear and therefore the upwind scheme of the explicit part, as first order scheme, recovers the exact solution.

5.3 Implicit Active Flux Scheme

The active flux method as presented in Section 4.2.6 is an explicit scheme and thus subject to the CFL condition. In its construction, the idea of the characteristic scheme is very similar. We want to briefly present some ideas to increase the order of the characteristic scheme and an implicit version of the active flux method. Unfortunately, both those ansatzes did not lead to stable schemes.

5.3.1 Extending Characteristic Scheme to Higher Order

We now want to extend the idea of the characteristic scheme to second order. Therefore, we use an additional degree of freedom in every cell, representing the slope $m_i = u'(x_i)$.

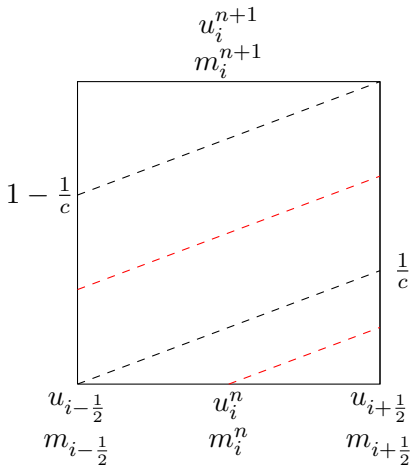

Figure 5.2: Characteristic scheme with linear data representation

As above, we can transport this information with velocity a. For the interval $[0, \frac{1}{c}]$ it is just the space-time transformation of the data in cell i:

$$(u_{i+\frac{1}{2}}, m_{i+\frac{1}{2}})(t) = (u_i^n, -\frac{1}{c} m_i^n), \text{ for } t < \frac{1}{c}.$$

Note, that due to the transformation, derivatives get scaled by $-\frac{1}{c}$. For the upper part of the interval, we have to use the information coming from $u_{i-\frac{1}{2}}$. Since only the part $u_{i-\frac{1}{2}}|_{0,1-\frac{1}{c}}$ is transported to $u_{i+\frac{1}{2}}$, the cell mean has to be corrected. The desired cell mean is the value at $t = \frac{1}{2} - \frac{1}{2c}$. So altogether we have

$$u_{i+\frac{1}{2}}(t) = \begin{cases} (u_i^n, -\frac{1}{c}m_i^n), \text{ for } t \leq \frac{1}{c} \\ (u_{i-\frac{1}{2}} - \frac{1}{2c}m_{i-\frac{1}{2}}, m_{i-\frac{1}{2}}), \text{ for } t > \frac{1}{c}. \end{cases}$$

Those 4 degrees of freedom have to be reduced to a representation of the form $(u_{i+\frac{1}{2}}, m_{i+\frac{1}{2}})$. The mean value should be preserved, so we have

$$u_{i+\frac{1}{2}} = \frac{1}{c}u_i^n + (1 - \frac{1}{c}) \left[u_{i-\frac{1}{2}} + m_{i-\frac{1}{2}}(-\frac{1}{2c}) \right].$$

For the determination of the derivative, we use moment matching. The first moment of the new linear function should contain the sum of the first moments of the two parts, i.e.

$$m_{i+\frac{1}{2}} = \frac{3}{2} \left[\int_{-\frac{1}{2c}}^{\frac{1}{2c}} \tau \left(u_i^n - \frac{1}{c}\tau m_i^n \right) \mathrm{d}\tau + \int_{-\frac{1-\frac{1}{c}}{2}}^{\frac{1-\frac{1}{c}}{2}} \tau \left(u_{i-\frac{1}{2}} - \frac{1}{2c}m_{i-\frac{1}{2}} + m_{i-\frac{1}{2}}\tau \right) \mathrm{d}\tau \right]$$

which results in the form

$$m_{i+\frac{1}{2}} = \frac{3}{2} \left((\frac{1}{c})^3 m_i^n + (1 - \frac{1}{c})^3 m_i^{n+1} \right).$$

In order to get the value of cell u_{i+1}^{n+1}, the interface information is transported along the characteristics. Then the cell mean consists of the segment $[1 - \frac{1}{c}, 1]$ transformed from time to space. Altogether we have

$$(u_{i+1}^{n+1}, m_{i+1}^{n+1}) = (u_{i+\frac{1}{2}} + (\frac{1 - \frac{1}{c}}{2})m_{i+\frac{1}{2}}, -cm_{i+\frac{1}{2}}). \tag{5.4}$$

Numerical test showed however, that scheme (5.4) is in fact only first order accurate. It is just slightly less diffusive than the first order version and the implicit upwind scheme for periodic boundary conditions. Unfortunately, in the case of Dirichlet boundary conditions it was completely unstable for implicit time steps, such that it is not applicable for our setting. The influence of boundary conditions on scheme stability is explored in more detail in Section 5.4.

5.3.2 Implicit Active Flux Scheme

Similar to above considerations, we investigated an implicit version of the active flux scheme. When the CFL number is larger than two, both characteristics that are considered in Figure 4.2 hit the left boundary of the space-time cell, compare Figure 5.3. It

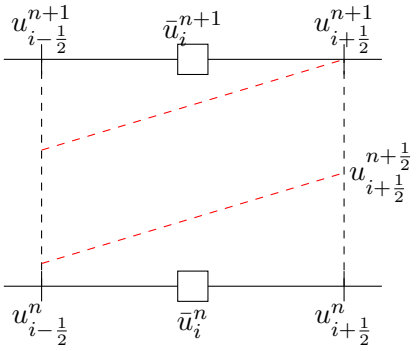

Figure 5.3: Characteristics in the active flux scheme (implicit case)

is still possible to formulate the scheme in this case. Then the quadratic interpolation at the interface $x_{i+\frac{1}{2}}$ depends on the one from the previous interface with

$$p_{i+\frac{1}{2}}(t^n + \frac{\Delta t}{2}) = p_{i-\frac{1}{2}}(t^n + \Delta t(\frac{1}{2} - \frac{1}{c}))$$
$$p_{i+\frac{1}{2}}(t^{n+1}) = p_{i-\frac{1}{2}}(t^{n+1} - \frac{\Delta t}{c})$$

The main problem here is, that the numerical flux no longer depends on the cell averages, but only on the point values on the interfaces. This leads to an evolution of the point values, that is completely decoupled from the averages. They consequently might drift apart in the course of the computation. The updates of $u_{i+\frac{1}{2}}$ just follow a collocation method of second order.

Another option is the use of integral interpolation instead of a pointwise consideration. Assuming $c \geq 2$ we define the quadratic polynomial in time $p_{i+\frac{1}{2}}$ at the interface $x_{i+\frac{1}{2}}$ by

$$\int_{t^n}^{t^n+\frac{\Delta t}{2}} p_{i+\frac{1}{2}}(\tau)d\tau = \frac{1}{c}\bar{u}^n_i + \int_{t^n}^{t^n+\Delta t(\frac{1}{2}-\frac{1}{c})} p_{i-\frac{1}{2}}(\tau)d\tau$$
$$\int_{t^n+\frac{\Delta t}{2}}^{t^n+\Delta t} p_{i+\frac{1}{2}}(\tau)d\tau = \int_{t^n+\Delta t(\frac{1}{2}-\frac{1}{c})}^{t^n+\Delta t(1-\frac{1}{c})} p_{i-\frac{1}{2}}(\tau)d\tau$$
$$p_{i+\frac{1}{2}}(t^n) = u^n_{i+\frac{1}{2}}.$$

Then the finite volume update can be performed in the usual way with the numerical flux

$$F^n_{i+\frac{1}{2}} = \int_{t^n}^{t^{n+1}} vp(\tau)d\tau.$$

Unfortunately, the resulting scheme proved to be unstable.
In the following a similar approach is presented, which can construct stable finite volume schemes of higher order.

5.4 Implicit High Order Schemes

The finite volume formulation itself describes a generic approach for the construction of numerical schemes, as already described in Section 4.2. The goal is to find an approximation $F_{i+\frac{1}{2}}$ to the analytical flux $f_{i+\frac{1}{2}}$ of the desired order. In the case of the ADER schemes, this is done by calculating a spatial WENO reconstruction.

In this section we want to investigate the reconstruction using a combination of explicit and implicit cell values. We analyze the stability of those schemes numerically in order to get candidates worth further investigation. We have submitted parts of the following sections as a separate paper, a preprint version is available at [30].

5.4.1 Reconstruction at Interface

Consider the interface $x_{i+\frac{1}{2}}$ and all cells inside a 2-neighborhood. We trace the cell values along their characteristics to the interface, see Figure 5.4. For each cell, this leads to an interval on the time axis with the corresponding cell value as average. Then we construct a polynomial having those average values on the respective intervals.

Let c be the CFL number, then the cell value of u_i^n will be transported to the interval $[t^n, t^n + \frac{1}{c}\Delta t]$, compare Section 5.4.1 and the corresponding integral condition of the interpolation polynomial is

$$\int_{t^n}^{t^n+\frac{1}{c}\Delta t} p(\tau)d\tau = \frac{1}{c}u_i^n.$$

Similarly, e.g. for the implicit cell value u_i^{n+1}, we get the condition

$$\int_{t^{n+1}}^{t^{n+1}+\frac{1}{c}\Delta t} p(\tau)d\tau = \frac{1}{c}u_i^{n+1},$$

and analogously for all the other cell values. Altogether, that leads to 8 possible conditions the reconstruction polynomial can fulfill, giving us an approximation of seventh order. When not using all the cell values for the reconstruction, different polynomials of orders one to six can be built. There are a total of $k = 2^8 - 1$ different combinations, which cells to use, leading to 255 different possible schemes. The finite volume method is constructed from the polynomial p via

$$F_{i+\frac{1}{2}} = \int_{t^n}^{t^{n+1}} vp(\tau)d\tau.$$

Obviously, many of those methods are not stable. This is why we numerically investigate the stability properties of the different possible choices.

5.4.2 Stability Analysis

For the investigation of the stability of numerical schemes, the most common technique is the so-called von Neumann analysis. It is based on the Fourier transform of the

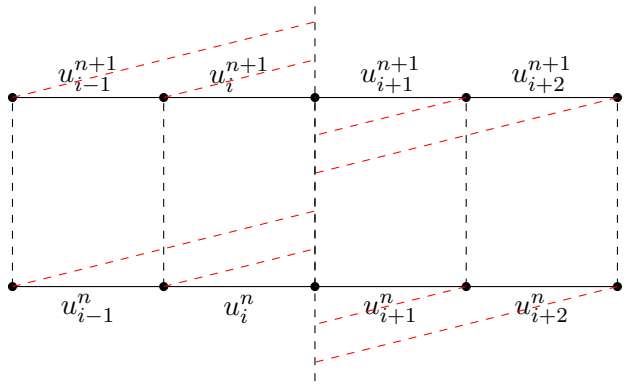

Figure 5.4: Tracing the characteristics to $x_{i+\frac{1}{2}}$

solution u. Since the Fourier transform preserves the $\mathbf{L^2}$-norm of a function, the stability analysis is investigated on the Fourier frequencies. One assumes the solution u to be a wave of the form

$$u_j^n = G^n e^{i\omega j}.$$

Note that we use the discretization index j instead of i in this section to distinguish it from the imaginary unit $i^2 = -1$. Inserting this ansatz into the numerical scheme leads to an equation determining G. If the absolute value $|G| \leq 1$, the numerical scheme is considered von Neumann stable. Applied to the implicit upwind scheme (5.2) for example, we get

$$\begin{aligned} Ge^{i\omega j} &= e^{i\omega j} - c(Ge^{i\omega j} - Ge^{i\omega(j+1)}) \\ \iff G &= \left[1 + c - c(\cos(\omega) - i\sin(\omega))\right]^{-1}, \end{aligned}$$

and for the absolute value we get

$$|G|^2 = \left[1 + 2c(1 - \cos(\omega)) + 2c^2(1 - \cos(\omega))\right]^{-1} \leq 1, \quad \forall c, \omega$$

confirming the unconditional stability of the implicit upwind scheme. For schemes involving many cells, the analytical solution for the magnitude of G is no longer that obvious. This is why we numerically evaluate the absolute value of G for all the schemes and check whether it is larger or smaller than one. Since the involved integrations and interpolations are linear operations, we can write the resulting scheme in the form

$$A_k u^{n+1} = B_k u^n,$$

where $k \in \{1, \dots, 255\}$ indicates the choice of the stencil and

$$A_k = \begin{pmatrix} & \ddots & & & & & \\ \dots & a_{-2}^k & a_{-1}^k & a_0^k & a_1^k & a_2^k & \dots \\ & & & & \ddots & & \end{pmatrix}, B_k = \begin{pmatrix} & \ddots & & & & & \\ \dots & b_{-2}^k & b_{-1}^k & b_0^k & b_1^k & b_2^k & \dots \\ & & & & \ddots & & \end{pmatrix}. \tag{5.5}$$

The update in the frequency domain is

$$G_k(c,\omega) = \frac{b_{-2}^k e^{i\omega(j-2)} + b_{-1}^k e^{i\omega(j-1)} + b_0^k e^{i\omega j} + b_1^k e^{i\omega(j+1)} + b_2^k e^{i\omega(j+2)}}{a_{-2}^k e^{i\omega(j-2)} + a_{-1}^k e^{i\omega(j-1)} + a_0^k e^{i\omega j} + a_1^k e^{i\omega(j+1)} + a_2^k e^{i\omega(j+2)}}.$$

A numerical evaluation of $G_k(c,\omega)$ for different $c > 1$ and $\omega \in [0, 2\pi]$ gives the following statistics:
From the 255 schemes

- 31 are unconditionally stable
- 98 are stable for $c \leq 1$
- 15 are stable for $c \geq 1$

For the application to the DH network, the choice of the stencil has significant influences on the formulations of the coupling conditions on the numerical level. As already pointed out in the previous chapter, a pure upwind formulation is most beneficial, as the coupling can be formulated directly on the numerical fluxes. Therefore we have a closer look at the schemes using just the upwind directed data.

We want to emphasize, that the kind of boundary conditions implied can influence the stability of a scheme. Above considerations are only valid for periodic boundary conditions. When applying boundary conditions of Dirichlet type, we need a different criterion. This can be motivated by the following observations:
The operator norm of the linear operator $A^{-1}B$ is given by its largest singular value and describes the maximum growth from one time step u^n to the next u^{n+1}. Matrices of the form (5.5) are circulant and therefore also normal, i.e.

$$A^T A = A A^T.$$

For normal matrices, the singular values are the absolute values of their eigenvalues. The connection between circulant matrices and the Fourier transform gives an explicit formula for their eigenvalues λ_l:

$$\lambda_l(A) = \sum_{j=0}^{N-1} a_j \omega^{lj} \ ,$$

where N is the dimension of the matrix and $\omega = e^{\frac{i2\pi}{N-1}}$ the Fourier frequencies. This leads to the stability condition

$$\frac{|\lambda_{B,l}|}{|\lambda_{A,l}|} \leq 1 \qquad \forall l = 0, \ldots, N-1 \tag{5.6}$$

for the scheme $u^{n+1} = (A^{-1}B)u^n$. In the case of Dirichlet boundary conditions, the system matrices A and B are no longer normal, that means we need to compute the singular values σ of $(A^{-1}B)$ in order to get a sufficient stability condition

$$\sigma_l \leq 1 \qquad \forall l = 0, \ldots, N-1. \tag{5.7}$$

We applied this analysis on all possible upwind biased combinations of stencils up to order 4. The results are summarized in Table 5.1. Out of the 16 purely upwind based

Nr.	u_i^{n+1}	u_i^n	u_{i-1}^{n+1}	u_{i-1}^n	order	stable expl.	stable impl.
1	x	x	x	x	4	✓	✓ ✓
2	x	x	x		3		✓ ✓
3	x	x			2	✓ ✓	✓ ✓
4	x		x		2	✓ ✓	✓ ✓
5	x			x	2	✓ ✓	
6		x	x		2	✓ ✓	
7		x		x	2	✓ ✓	
8	x				1	✓ ✓	✓ ✓
9		x			1	✓ ✓	
10			x		1		✓

Table 5.1: Stability of different stencils for periodic (✓) and Dirichlet boundary conditions (✓). Unstable combinations are not listed.

schemes, 10 show stable behavior in at least one case. We recognize the explicit upwind scheme in Nr. 9 and the implicit upwind scheme in Nr. 8. We want to focus on high order schemes in this work, that is why we will have a closer look on the fourth and third order methods.
For the fourth order scheme the matrix $A \in \mathbb{R}^{N\times N}$ is given by

$$A = \begin{pmatrix} a_0 & & & a_{-2} & a_{-1} \\ a_{-1} & a_0 & & & a_{-2} \\ a_{-2} & a_{-1} & a_0 & & \\ & \ddots & \ddots & \ddots & \\ & & a_{-2} & a_{-1} & a_0 \end{pmatrix}, \tag{5.8}$$

with $a_0 = \frac{1}{6} + \frac{c}{4} + \frac{c^2}{12}$, $a_{-1} = \frac{2}{3} - \frac{c^2}{6}$, $a_{-2} = \frac{1}{6} - \frac{c}{4} + \frac{c^2}{12}$. B has the same form with the coefficients $b_0 = \frac{1}{6} - \frac{c}{4} + \frac{c^2}{12}$, $b_{-1} = \frac{2}{3} - \frac{c^2}{6}$, $b_{-2} = \frac{1}{6} + \frac{c}{4} + \frac{c^2}{12}$.

For the six point stencil of the fourth order scheme we obtain for A

$$|\lambda_{A,k}|^2 = a_0^2 + a_{-1}^2 + a_{-2}^2 + 2\cos(\frac{2\pi k}{N})(a_0 a_{-1} + a_{-1}a_{-2}) + \left(4\cos^2(\frac{2\pi k}{N}) - 2\right) a_0 a_{-2} \ .$$

Since $a_0 = b_{-2}$, $a_{-1} = b_{-1}$ and $a_{-2} = b_0$ it follows directly, that

$$\frac{|\lambda_{B,k}|^2}{|\lambda_{A,k}|^2} = 1 \qquad \forall k = 0, \ldots, N-1 \ .$$

Thus all eigenvalues of the solution operator $A^{-1}B$ have modulus one and therefore the fourth order scheme is unconditionally stable.

For the scheme of order 3, evaluation of the stability condition is more involved, as the symmetry in A and B is missing. Therefore we evaluate (5.6) numerically to indicate the stability region of the scheme. In Figure 5.5, the modulus of the eigenvalues is shown for different CFL numbers and $N = 50$. We observe, that for $c < 1$ the modulus

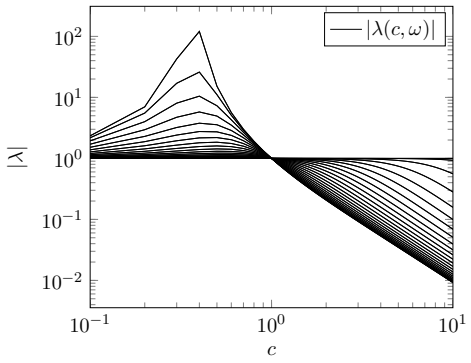

Figure 5.5: Eigenvalues of $A^{-1}B$ for third order scheme for different frequencies $\omega = e^{\frac{i2\pi k}{N}}, k \in (0, \ldots, N-1)$

is larger than 1 and the scheme is unstable. For $c > 1$ all eigenvalues are smaller or equal than 1, which indicates stability.

Next, we discuss the stability of the two schemes in case of Dirichlet boundary conditions. The layout of the matrices slightly changes, i.e. compared to (5.8) the upper right block is eliminated and the matrix $\tilde{A}$ has the form

$$\tilde{A} = \begin{pmatrix} a_0 & & & & \\ a_{-1} & a_0 & & & \\ a_{-2} & a_{-1} & a_0 & & \\ & \ddots & \ddots & \ddots & \\ & & a_{-2} & a_{-1} & a_0 \end{pmatrix} \ .$$

The new form of $\tilde{B}$ is analogue. These matrices are no longer normal, which implies that the error amplification factor is measured by the largest singular value of $\tilde{A}^{-1}\tilde{B}$, which is not related to its eigenvalues.

As we did not find any analytical expressions, we rely on a numerical evaluation of the stability regions. In Figure 5.6 the largest singular value is shown for different matrix sizes N and different CFL numbers.

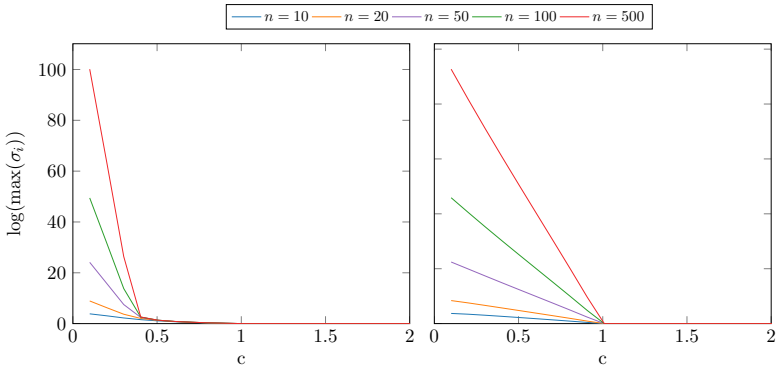

Figure 5.6: Largest singular values of $\tilde{A}^{-1}\tilde{B}$ for the third (left) and fourth (right) order scheme for different n and c

We observe that both schemes are unstable for $c < 1$, but stable for $c > 1$. Note that for the scheme of order 4 this differs from the case of periodic boundary conditions. Since for $c < 1$ the size of the largest singular value increases with N we do expect this instability to hold for any choice of discretization. Thus both schemes are unstable for $c < 1$. This is just a mild drawback, as in this case a classical explicit method can be used.

To sum up, we found two Finite Volume schemes of third and fourth order respectively, that use a pure upwind stencil and are stable for all time steps with $c > 1$. The numerical flux functions corresponding those methods are

$$F^3_{i+\frac{1}{2}} = v\left(\frac{c^2+3c-4}{6c}u_i^{n+1} - \frac{c^2-c}{6c}u_{i-1}^{n+1} + \frac{c+2}{3c}u_i^n\right) \tag{5.9}$$

and

$$\begin{aligned} F_{i+\frac{1}{2}} = v\Bigg[&\left(\frac{c}{12}+\frac{1}{4}+\frac{1}{6c}\right)u_{i-1}^n + \left(-\frac{c}{12}+\frac{1}{4}-\frac{1}{6c}\right)u_{i-1}^{n+1} \\ &+\left(-\frac{c}{12}+\frac{1}{4}+\frac{5}{6c}\right)u_i^n + \left(\frac{c}{12}+\frac{1}{4}-\frac{5}{6c}\right)u_i^{n+1}\Bigg] \ . \end{aligned} \tag{5.10}$$

5.5 Implicit Limiting

As already discussed for the explicit case, high order linear schemes tend to introduce oscillations near discontinuous solutions or sharp gradients. They can be reduced or

avoided by introducing nonlinearities to the system. Especially for implicit numerical schemes, the step from linear systems to nonlinear ones results in a large raise of computational complexity. We present several approaches in limiting high order implicit schemes.

5.5.1 Flux Corrected Transport

In the recent years, flux corrected transport (FCT) schemes gained more attention in simulation of fluid dynamic problems. The basic idea of FCT is very similar to the classical flux limiting technique [47]. Let F^L be the numerical flux corresponding to a monotone scheme and F^H the numerical flux of a high order method. The numerical solution is then first transported with the diffusive but monotone first order method

$$\tilde{u}_i^{n+1} = u_i^n - \frac{\Delta t}{\Delta x}(F^L_{i+\frac{1}{2}} - F^L_{i-\frac{1}{2}}).$$

Then the so-called "antidiffusive fluxes" $F^{AD} = F^H - F^L$ are computed. A direct FV update with those fluxes would retrieve the oscillatory high order solution. Instead those antidiffusive fluxes are limited to avoid nonphysical values

$$F^C = \alpha F^{AD}, \qquad 0 \leq \alpha \leq 1$$

and finally the new solution is computed as

$$u_i^{n+1} = \tilde{u}_i^{n+1} - \frac{\Delta t}{\Delta x}(F^C_{i+\frac{1}{2}} - F^C_{i-\frac{1}{2}}).$$

The concept of FCT has already been successfully implemented for simulation of compressible Euler equations in [52]. There the authors apply the method to an implicit formulation with finite element discretizations.
In the work of Steinle and Morrow [67], an implicit flux correction algorithm for one dimensional advection problems was presented. While they seemed to achieve good results with their approach, we would like to emphasize some difficulties in applying it to our framework.
The main ingredient of the limiting process [78] is avoiding the creation of new extrema when applying the antidiffusive fluxes. The problem is that in our case with large time steps, the low order solution introduces very large diffusion. Then the limited fluxes cannot make up for the decreased magnitude of the peaks of the true solution. This comes from the fact, that the amount antidiffusion is limited by a discrete maximum principle. If the low order solution is highly diffusive, the high order one violates this maximum principle everywhere and just a mild antidiffusion can be applied.

One big advantage of the FCT formulation is that the numerical fluxes can be computed by arbitrary methods. For instance it is possible to perform 5 steps with an explicit upwind scheme and $c = 0.8$, adding the numerical fluxes of each individual step to form F^L instead of doing a single implicit upwind step with $c = 4$. In [67] exactly this procedure is proposed and could improve the transport stage of the FCT algorithm.

Then the computational effort would be similar to the LTS scheme from section 4.3, so it would be more efficient to directly use the exact transport algorithm of that method without the need of antidiffusion afterwards.

5.5.2 Implicit WENO

Another possibility to avoid oscillations in high order implicit methods is the use of implicit WENO schemes (iWENO). There are several works describing iWENO methods for different applications ([17],[16],[80]). The reconstructions used are identical or at least similar to the ones presented in 4.2.4. For implicit time stepping, there are several options. In [4] and [35], implicit Runge-Kutta methods are used as time integrators, while in the second article additionally a semi-Lagrangian approach is used. They get the expected excellent results on one dimensional segments but as already discussed above, Lagrangian schemes are very hard to implement on networks, especially for large time steps. In [3] the authors use Simpsons rule to approximate the integral at the interface using the WENO reconstructions from the three different time levels $t^i, t^{i+\frac{1}{2}}, t^{i+1}$. The numerical results look very promising, but for the application on large networks or large CFL numbers the nonlinear systems become hard to solve and the method becomes very slow.

5.5.3 A Posteriori Limiting

Originally designed for the solution of multidimensional systems of conservation laws, the Multidimensional Optimal Order Detection (MOOD) paradigm [21],[20] can be adapted to our requirements while eliminating most disadvantages of the options above. In contrast to FCT, the key idea of a posteriori limiting is to begin with the computation of a high order solution. Only if that solution appears to be unsuitable, the limiting is activated and a lower order solution is computed. This has the big advantage, that no comparison between high and low order solution is needed. Especially for large time steps the difference between the two would clearly get large resulting in the overlimiting of FCT ansatzes.

We now give a brief introduction on the basic a posteriori procedure. Furthermore we present the high order cascade we use in this work and propose an additional convex combination of fluxes of different orders to smooth out the order switching process. The paper describing the full scheme has been submitted to Advances in Computational Mathematics, a preprint is available in [30].

Whether a solution is suitable can be checked with some (possibly problem specific) criteria. The basic procedure is the following:

1. Compute a high order solution for the next time level
2. Check, whether this solution can be accepted (i.e. it does not produce spurious oscillations or negative mass, etc.)

3. If high order solution must be discarded, recompute with a lower order
4. Iterate (2.-3.) until a monotone fallback solution is reached or the solution passes the check.

In the determination, whether a computed value is suitable, Diot *et al.* propose the following six criteria [25]:

Physical Admissibility Detection (PAD):
Most importantly, the solution has to fulfill some basic admissibility criteria in order to ensure that the numerical code does not crash and the variables stay in a meaningful regime.
A classical physical admissibility condition is positivity of the solution. For our application on district heating networks, we incorporate checks for lower and upper bounds of the energy density. The energy must not fall below the return temperature in order to have a well defined consumer model for example. Possible additional conditions are the check for NaN or INF, indicating that something with the solution went wrong.

While the PAD can detect oscillations overshooting the total range, it does not find the ones near steep gradients with moderate values. Those have to be found by numerically analyzing the computed solution. Using a neighborhood of the current cell and evaluating curvature data, it is possible to distinguish local oscillations from the smooth solution as follows.

Extrema Detector (ED):
If

$$\min(u_{i-1}, u_{i+1}) \leq u_i \leq \max(u_{i-1}, u_{i+1})$$

does not hold, cell u_i is a local extremum. This can be either a true extremum of the solution, a local oscillation or some artifact due to round off errors. If no extremum is found, the computed value is considered valid. In presence of an extremum in order to identify the right case, the local curvatures next to the cell have to be taken into account.
We define

$$C_i(u) = u_i^{''}$$

a discrete approximation to the local curvature (eg. by central difference). The local curvature indicators are defined as

$$\chi_{i,m} = min(C_{i-1}, C_i, C_{i+1}), \qquad \chi_{i,M} = max(C_{i-1}, C_i, C_{i+1}).$$

Depending on these indicators, there are three cases to be distinguished:

Plateau Detector (PD): If the total curvatures are close to zero, we have found a local plateau, where the extremum arised from floating point errors.

$$max(|\chi_{i,m}, \chi_{i,M}|) < \epsilon_{PD},$$

This means we found a valid solution.

Local Oscillation Detector (LOD):
If the signs of the curvature change, i.e.

$$\chi_{i,m}\chi_{i,M} < 0,$$

there is a local oscillation in cell u_i. In order to avoid rounding errors, a relaxed version is used

$$\chi_{i,m}\chi_{i,M} < \epsilon_{LOD}.$$

Smoothness Detector (SD):
There is a smooth extremum, if the minimal and maximal curvature are close, i.e.

$$1 \geq \frac{min(|\chi_{i,m}, \chi_{i,M}|)}{max(|\chi_{i,m}, \chi_{i,M}|)} \geq 1 - \epsilon_{SD}.$$

The check can be schematically drawn as in Figure 5.7.

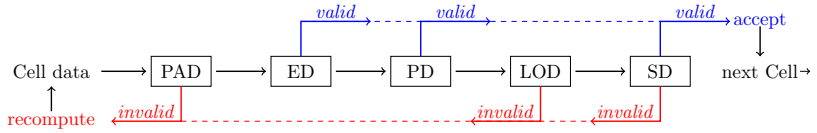

Figure 5.7: Admissibility check procedure

Whenever the check finds 'invalid' cells, the order of the corresponding flux is reduced. Then the check is performed again with the new cell data. In this work, we use a cascade of three schemes with the orders $4 \rightarrow 3 \rightarrow 1$, where the high order schemes are as described above and the first order scheme is the implicit upwind scheme, which always produces valid results.

Discrete Maximum Principle (DMP):
An important criterion for most numerical schemes is the satisfaction of a discrete maximum principle. This is a very powerful tool for the detection of spurious oscillations, as the high order overshoots would violate the DMP and can therefore be eliminated. However, for the application of implicit schemes on transport equations we do not have a discrete maximum principle. The long travel distance of cell values makes it impossible to state bounds for the solution. This is why we have to drop the DMP check from our procedure.

Extrapolation and computation of downwind data for MOOD check

The MOOD check is a function $y = MOOD(u_{i-l}^{n+1}, \ldots, u_{i+k}^{n+1}) \in \{TRUE, FALSE\}$. For the evaluation of the ED step, which is the most crucial one in order to detect local

oscillations, that means at least the cell data $(u_{i-1}^{n+1}, u_i^{n+1}, u_{i+1}^{n+1})$ has to be provided. This means in addition to the candidate solution u_i^{n+1}, we have to provide the cell value u_{i+1}^{n+1} as well. While inside the domain, we have access to u_{i+1}^n and the computation of the new cell value at $i+1$ is possible. Note, that this new data has to be computed with the same order as the value to be checked.
While at the boundary, i.e. at the last cell of any edge in the network we do not have the data required to compute another cell value. In this case, we have to extrapolate u_{i+1}^{n+1} using only values inside the domain. Also here we require an accuracy of the same order than used before. The numerical flux functions we use are:

$$F_c^4 = v\left(\frac{-c^2+3c+10}{24}u_{i-1}^{n+1} + \frac{c^3+19c+18}{12c}u_i^{n+1} + \frac{-c^3-2c^2+13c+62}{24(c-1)}u_{i+1}^{n+1} + \frac{c^2+5c+6}{4(c-1)c}u_i^n\right)$$

for the fourth order and

$$F_c^3 = v\left(-\frac{c^2+3c+4}{6c}u_i^{n+1} + \frac{c^2-13}{6(c-1)}u_{i+1}^{n+1} + \frac{c^2+3c+2}{3(c-1)c}u_i^n\right)$$

for the third order.
These fluxes are only used for filling ghost cells that are needed in the MOOD check. They do not directly contribute to the solution itself.

5.5.4 Convex Combination of High Order Fluxes

The classical MOOD strategy determines the order of the spatial reconstruction polynomial for a given cell. Therefore, there is a hard switch of the corresponding fluxes, when the order is decreased. Since we are using the strategy directly on the fluxes, a continuous transition between the different orders is possible. Whenever the order for a given cell has to be reduced from order 4 to order 3 and the third order solution is suitable, we might as well use a flux of the form

$$F = \alpha F^4 + (1-\alpha)F^3.$$

Then the parameter α can be chosen in a way, such that

$$\alpha^\star = max\{\alpha | u_i^{n+1} \text{satisfies the MOOD check}\}.$$

Unfortunately, this optimization problem is discontinuous, as the MOOD check will only return TRUE or FALSE. The parameter α here is determined by a bisection method. The incorporation of sufficient criteria for the check might accelerate the process.

Altogether, we have constructed a hybrid scheme taking advantage of the monotonicity of the first order upwind and the good approximation quality of the high order ones.

We finally end up with several different options of how to apply the limiter. The most simple one would be the direct switch from fourth to first order whenever the MOOD-check fails. We denote this variant by $H1$. A second option is to incorporate a larger cascade of schemes. In the method $H2$ we introduce an intermediate step with the third order scheme and deploy the chain $4 \to 3 \to 1$ of scheme orders. Furthermore, we discussed the option to convex combine the fluxes of two different orders, whenever one of them fails the MOOD check and the other one passes it. This gives rise to two more versions, where $H3$ denotes the addition of this feature to $H1$ and $H4$ is the corresponding modification of $H2$. All four options will be investigated in Chapter 7.

5.5.5 Limitations of the Scheme

We finally want to discuss some limitations of the constructed scheme as in [30]. For explicit methods the MOOD approach guarantees certain properties by design. However with the implicit method, there are cases where the solution with the limiter applied still produces overshoots. This is due to the fact, that the flux orders are chosen sequentially throughout the domain. It is possible, that a choice at some point in the domain may locally fulfill the smoothness test but produces problems a few cells further downstream. The limited window of available data for the check makes it impossible to predict these cases. A possible way out would be an iteration process with backtracking elements to eliminate these cases. However, in practical applications this might imply a huge growth in computation times. For our application on district heating networks, it is not essential to avoid all small overshoots inside the range of admissible temperature values. When the total temperature however gets below a minimal value, the system is not solvable anymore and in these cases, such a mechanism has to be implemented. In realistic settings this does not happen, as near such points the resulting velocities and pressures would tent to infinity. In real applications, there always is a large margin between true solution and the set of solutions that are not admissible.

Chapter 6

Numerical Schemes for the Hydraulic Part

In the last chapter before coming to the results, we briefly want to explain the computation of the velocity as the solution of the hydraulic DAE. In Section 2.3 we introduced the splitting of hydraulic and transport part. This has the big advantage, that for each system we can use numerical methods specifically designed to its inherent properties. For the simulation of the transport part we presented several different schemes in the previous chapters. Here we give an algorithm for solving the hydraulic DAE.

Furthermore we briefly want to discuss schemes for solving the full set of equations in the case of low Mach compressible flow.

6.1 Solving Transformed Hydraulic DAE

The big advantage when solving (2.19):

$$\begin{aligned} \partial_x \mathbf{v} &= 0 \\ \mathbf{I}_{\mathcal{J}_p}\left(\partial_t \mathbf{v} + \frac{1}{\rho}(\mathbf{A}^I)^T \mathbf{p}\right) &= \mathbf{I}_{\mathcal{J}_p}\left(\frac{1}{2}\mathbf{\Lambda}\mathbf{D}^{-1}\mathbf{V}|\mathbf{v}| + g(\mathbf{A}^I)^T \mathbf{b}\right) \\ \mathbf{A}^I \mathbf{v} &= 0 \\ \mathbf{I}_{\mathcal{J}_c}\mathbf{v} &= \tilde{\mathbf{Q}}(\mathbf{e}, t) \\ \mathbf{I}_{\mathcal{J}_s}\mathbf{p} &= p^s \end{aligned} \tag{6.1}$$

is that we can use explicit knowledge of the algebraic constraints to reduce the complexity of the system. As it is in its general form, (6.1) is a nonlinear DEA with a solution in $\mathbb{R}^{n_{\mathcal{J}}+n_{\mathcal{V}}}$. In the analysis section we have already seen that there is an equivalent formulation to (6.1) of the form

$$v_1 = \left(\mathbf{A}^I \mathbf{A}_t^T\right)^{-1} \left(\mathbf{A}^I \mathbf{I}_{\mathcal{J}_C}\right) \mathbf{Q}(v_0, \mathbf{e}, t) \tag{6.2}$$

$$\dot{v}_2 = \left(\mathbf{A}_{PC}\mathbf{A}_{PC}^T\right)^{-1} \mathbf{A}_{PC}\left(r + \mathbf{f}\left(\mathbf{A}_t^T v_1 + \mathbf{A}_{PC}^T v_2 - \mathbf{A}_t^T \dot{v}_1\right)\right) \tag{6.3}$$

$$\mathbf{p} = \rho\left(\mathbf{A}_t(\mathbf{A}_r)^T\right)^{-1} \mathbf{A}_t\left(r + \mathbf{f}\left(\mathbf{A}_t^T v_1 + \mathbf{A}_{PC}^T v_2\right) - \left(\mathbf{A}_t^T \dot{v}_1 + \mathbf{A}_{PC}^T \dot{v}_2\right)\right), \tag{6.4}$$

which is exactly what we will solve in the numerical method.
Note, that in the numerics we stay with the original algebraic consumer formulation. In a first step, for given node energy values at the consumers, we solve the linear problem

$$\left(\mathbf{A}^I \mathbf{A}_t^T\right) v_1 = \left(\mathbf{A}^I \mathbf{I}_{\mathcal{J}_C}\right) \mathbf{Q}(v_0, \mathbf{e}, t),$$

for v_1, which is a linear system of size $n_{\mathcal{V}} - 1 \times n_{\mathcal{V}-1}$. After that, we solve the nonlinear problem (6.3) for v_2, which is of dimension $v_2 \in \mathbb{R}^{n_C}$. This means e.g. for tree shaped networks, the hydraulic DEA degenerates to a pure algebraic part, which can be solved directly.

Altogether, for solving the full system we perform a first order splitting. We initialize with $e(0,x) = e_0(x), v(0) = v_0, p(0) = p_0$. Then we iterate solving energy transport and hydraulic system, where for the coupling terms we insert the respective values of the previous time step. The numerical solution operators of the two respective systems are denoted by $\mathcal{T}_e^N(\cdot, \Delta t)$ and $\mathcal{T}_v^N(\cdot, \Delta t)$ similar to the analytic ones for given time step Δt. Then we calculate

Algorithm 6.1.1 *Solve Network*

- *Initialize* e^0, v^0,*set* $t = 0, i = 0$ *and specify* $\Delta t, t_{end}$.
- *While* $t < t_{end}$*:*
  1. $i = i + 1$
  2. $v^i = \mathcal{T}_v^N(e^{i-1}, \Delta t)$
  3. $e^i = \mathcal{T}_e^N(v^i, \Delta t)$
  4. $t = t + \Delta t, \Delta t = \min\{\Delta t, t_{end-t}\}$
- *return* e^i, v^i

This procedure obviously produces first order accurate solutions in relation to the grid size. However, the main source of errors is the numerical diffusion in the advection step, so it is reasonable to apply the additional effort of high order schemes exclusively in that step. Furthermore we want to emphasize that due to the much higher number of degrees of freedom in the transport step, also most of the computational time contributes there.

6.2 Methods for the Full System

There also is the option to solve the whole Euler system with the same numerical method. Generally such a scheme cannot account for the specific properties of different parts of the equation as the splitting scheme we use. On the other hand, there are several reasons for still doing so. A formulation of the district heating system in a port Hamiltonian framework is subject to current research. A first step was done in [34], where a port-Hamiltonian embedding of the compressible instationary thermodynamic

Euler system was presented. This kind of formulation enables straight forward coupling to other port-Hamiltonian systems where the overall structure of the coupled system retains the structural elements of its parts. Doing so, a coupling of e.g. district heating networks with gas networks or the power grid is possible. The asymptotic transition from the compressible to the incompressible case is non-trivial and subject to current research. In the numerical simulation of port-Hamiltonian systems Galerkin-type projections are most common, as they conserve the structure in the discrete setting.

For the simulation of the compressible equations in this context it is essential to use so called asymptotic preserving (AP) schemes. They avoid severe time step restrictions coupled to the propagation speed of pressure waves. When transforming the compressible Euler equations to dimensionless form, one ends up with the system

$$\begin{aligned} \partial_t \rho + \nabla \cdot (\rho \vec{v}) &= 0 \\ \partial_t (\rho \vec{v}) + \nabla \cdot (\rho \vec{v} \otimes \vec{v}) + \frac{1}{\epsilon^2} \nabla p &= 0 \\ \partial_t E + \nabla \left((E + p) \vec{v} \right) &= 0. \end{aligned} \tag{6.5}$$

This type of equation is well studied in the context of gas dynamics, where the parameter ϵ is directly linked to the Mach number, the ratio of flow velocity and speed of sound in the medium. When the Mach number is very small, the medium behaves nearly incompressible. It has been shown, that in the limit $\epsilon \to 0$ the solution of (6.5) converges to its incompressible counterpart [42],[43]. There exist multiple numerical methods for the simulation of the compressible Euler equations that fulfill the AP property. That means they give a consistent discretization of the system independent of the values of ϵ. The general idea is to treat the fast pressure waves implicitly, while the material transport is discretized in an explicit manner. In [24] a first order accurate all speed scheme is presented. By an IMEX (Implicit-EXplicit) time integration, several second order methods have been developed [63],[13]. However for the simulation of district heating networks, a pure incompressible consideration seems most efficient.

Chapter 7

Results

In the Chapters 4 and 5 several different numerical methods were presented. We evaluate the schemes in different test cases.

Altogether, we show the results for four different tests. The first one is Shu's test on one dimensional segments. We gradually increase the complexity by going from this case via smaller networks to real district heating scenarios.

7.1 Numerical Experiments for Explicit Schemes

The behavior of the classical FV schemes and ADER schemes on one dimensional segments are well-known. Due to its construction, the LTS scheme is exact on those domains and we therefore omit Shu's test case for the explicit schemes as it would not give any new insights.

7.1.1 Split Network

The first test for the explicit schemes will be on a small artificial split and join network shown in Figure 7.1 The network consists of 6 pipes and 4 junctions. All pipes have the

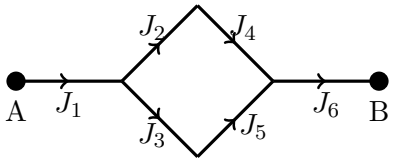

Figure 7.1: Graph of split-and-join network

same diameter and the lengths $L_1 = L_6 = 1$ and $L_2 = L_3 = L_4 = L_5 = \frac{1}{2}$. The velocities are constant in time but have different values $v_1 = v_6 = 1$, $v_2 = v_4 = \frac{1}{3}$,$v_3 = v_5 = \frac{2}{3}$.

The initial energy is zero in all pipes, the inflow profile at node A is given by

$$e_A(t) = \begin{cases} \sin^4(\pi x), & t < 1 \\ 0, & t \geq 1 \ . \end{cases} \tag{7.1}$$

The input function e_A is three times continuously differentiable and thus suitable for testing the schemes' general behavior and their convergence order. In all tests we are generally interested in the energy signals in selected nodes of the network, most commonly at the consumer nodes. For the split network, we compare the energy $e_B(t)$ at the node B. This is the major quantity of interest when operating district heating systems. It collects all the energy losses due to wall cooling and it is the only quantity coupling back to the velocity computation in real applications.

For the current test case the velocity is set constant. That is necessary for testing the convergence order of the transport step, since a coupled computation will always degenerate to first order convergence.

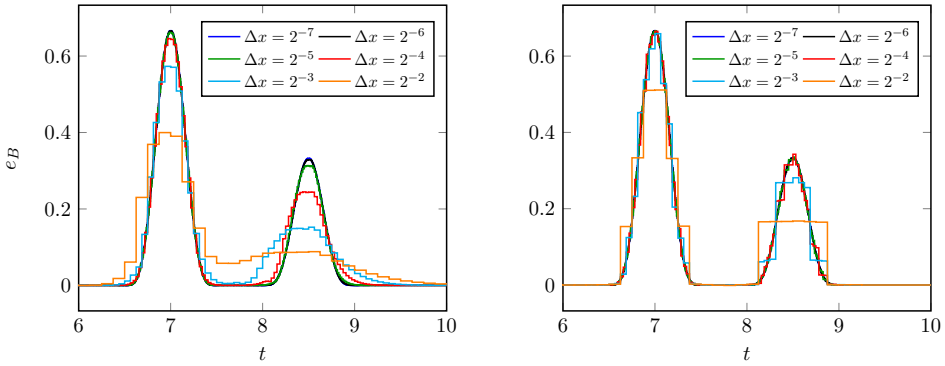

Figure 7.2: Energy e_B for different spatial grid resolutions computed with the third order ADER scheme (left) and the third order local time stepping scheme (right)

In Figure 7.2 the values of e_B in the time interval $[6, 10]$ are plotted. The picture on the left hand side shows the solution obtained with the third order ADER scheme for different spatial resolutions in all edges. For a coarse grid with only two or four cells per pipe the strong numerical diffusion lowers the peaks of the temperature significantly. As the spatial, and with it also the temporal, resolution is increased, the two signals separate and become more pronounced. The solutions seem to converge to a given profile, but only for the two finest resolutions the curves in the picture can not be distinguished any more.

The values obtained with the local time stepping scheme are shown on the right hand side. Even for the coarsest resolution the two peaks are separated. Their heights are comparable with those of the ADER scheme on a grid with twice as many cells. Already for a resolution of $\Delta x = 2^{-4}$ the final shape is reached.
In Figure 7.3 the numerical solutions with a fixed spatial grid with $\Delta x = 2^{-5}$ are shown. The curves in red are the ADER schemes of order one up to order 5, the blue curves display the solutions of the local time stepping scheme for an increasing order of accuracy. The reference solution in green has been computed using 5th order ADER scheme on a grid size of $\Delta x = 2^{-}9$. For both approaches the accuracy increases with the order of the method. The ADER schemes can capture the height of the first peak

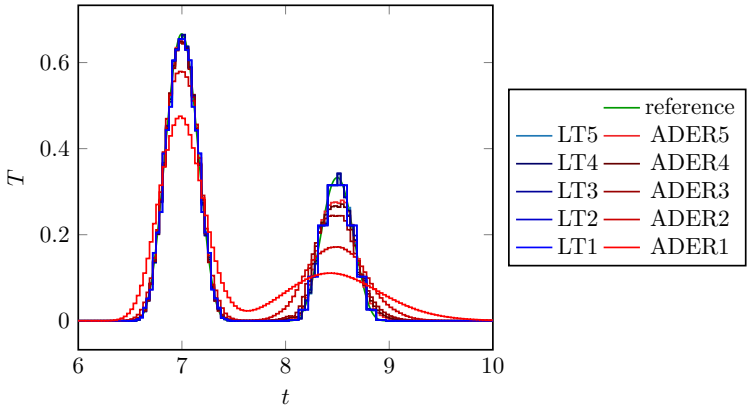

Figure 7.3: Schemes of different order on the same grid with $\Delta x = 2^{-5}$

for order three or more, but for the second peak the full height is never reached. In contrast to this, the local time stepping scheme provides more accurate solutions. For all orders both peaks have the correct height, but the values of the second peak seem to be on a coarser grid. This is related to the coarse resolution and the varying velocities in the network. The signal T_A send into the e_1 is split at the first node into one peak traveling along e_2, e_4 and another one along e_3, e_5. Since the flow in edge two and four is significantly slower, the arriving peak is compressed into very few grid cells. This loss of resolution can not be recuperated when the flows merge again, nor excluded by the present high order approach. Thus the coarser signal is transferred to the faster flow in e_6.

The Tables 7.1 and 7.2 show the L^1-errors, rates of convergence and computational times for the ADER and the local time stepping schemes. The L^1 errors are computed in comparison with a numerical reference solution obtained with the fifth order ADER scheme on a grid with $\Delta x = 2^{-10}$. Note that the fifth order ADER scheme can not be used on the coarsest grid, as there are not sufficient cells for the polynomial reconstruction available.

For all schemes the solution converge to the reference solution when the spatial grid is refined. With the ADER methods this is paid by quadratically increasing computational costs. For the local time stepping schemes the increase is only linear. Here the numerical effort is almost independent of the number of cells and just scales with the increasing number of time steps.

The orders of convergence are visualized in Figure 7.4. For fine enough grids all schemes achieve the predicted order of convergence. Except for the fifth order schemes, the local time stepping schemes are about one order of magnitude more accurate compared to the ADER methods.

In Figure 7.5 the efficiency of the schemes is studied. The local time stepping scheme needs significantly less time to obtain accurate numerical results, e.g. the fifth order

	ADER1			ADER3		
Δx	L^1 error	order	time	L^1 error	order	time
1.25e-1	3.477e-1	-	0.1	1.713e-1	-	0.8
6.25e-2	2.840e-1	0.29	0.2	8.495e-2	1.01	1.0
3.13e-2	2.104e-1	0.43	0.6	2.791e-2	1.61	3.8
1.56e-2	1.451e-1	0.54	1.6	3.577e-3	2.96	14.5
7.81e-3	9.276e-2	0.64	5.4	3.818e-4	3.23	55.9
3.91e-3	5.446e-2	0.77	19.7	4.226e-5	3.18	221
1.95e-3	2.890e-2	0.91	74.6	4.460e-6	3.24	880

	ADER5		
Δx	L^1 error	order	time
1.25e-1	-	-	0.9
6.25e-2	2.413e-2	-	2.4
3.13e-2	7.892e-4	4.93	6.0
1.56e-2	2.728e-5	4.85	22.8
7.81e-3	8.280e-7	5.04	90.0
3.91e-3	2.580e-8	5.00	355
1.95e-3	9.715e-10	4.73	1411

Table 7.1: L^1-errors, rates of convergence and computational times for the ADER schemes

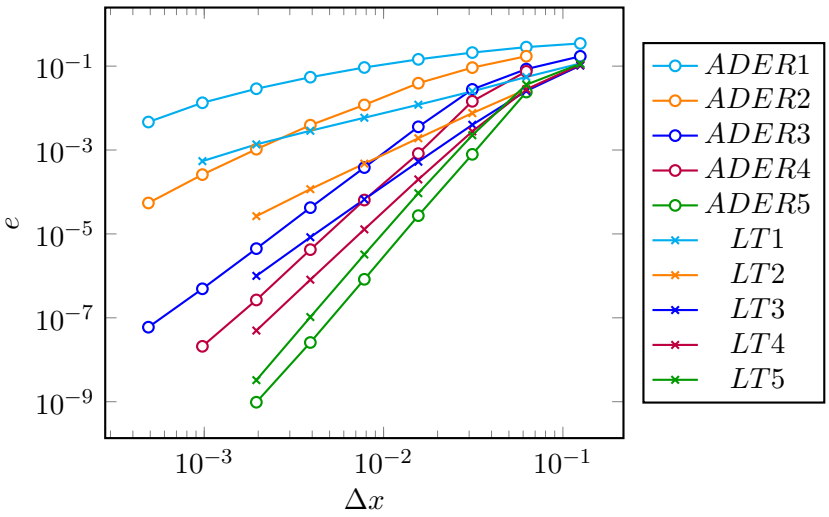

Figure 7.4: Convergence plots for the different schemes

	LT1			LT3		
Δx	L^1 error	order	time	L^1 error	order	time
1.25e-1	1.157e-1	-	0.1	1.035e-1	-	0.1
6.25e-2	5.508e-2	1.07	0.2	2.556e-2	2.02	0.2
3.13e-2	2.507e-2	1.14	0.4	3.996e-3	2.68	0.4
1.56e-2	1.202e-2	1.06	0.8	5.269e-4	2.92	0.7
7.81e-3	5.877e-3	1.04	1.4	6.669e-5	2.98	1.5
3.91e-3	2.876e-3	1.08	2.7	8.328e-6	3.00	3.4
1.95e-3	1.363e-3	1.33	5.9	9.988e-7	3.06	6.7

	LT5		
Δx	L^1 error	order	time
1.25e-1	1.133e-1	-	0.1
6.25e-2	3.612e-2	1.65	0.2
3.13e-2	2.260e-3	4.00	0.4
1.56e-2	9.352e-5	4.59	0.8
7.81e-3	3.227e-6	4.86	1.6
3.91e-3	1.042e-7	4.95	3.2
1.95e-3	3.266e-9	5.00	7.0

Table 7.2: L^1-errors, rates of convergence and computational times for the local time stepping schemes

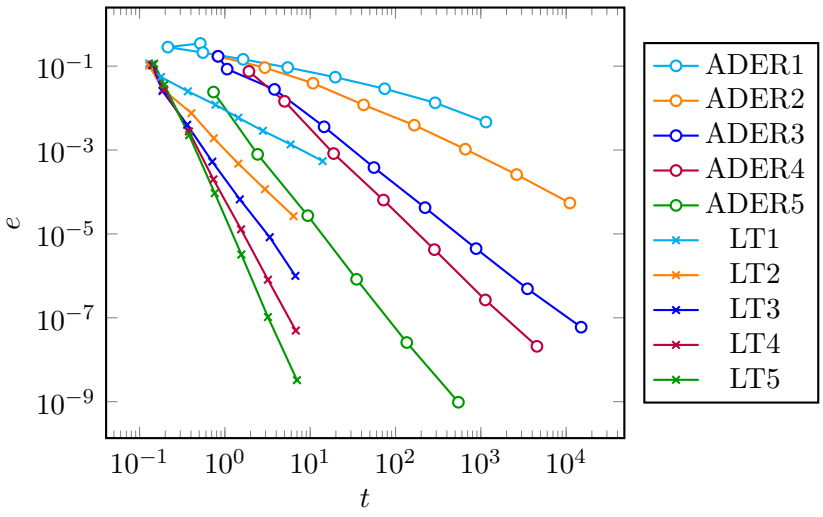

Figure 7.5: Efficiency plots for the different schemes

local time stepping scheme on the finest grid with $\Delta x = 2^{-9}$ takes as long as the first order upwind scheme with $\Delta x = 2^{-5}$, but the error is 3.266e-9 in contrast to 2.104e-1.

7.1.2 One-to-One Coupling with Varying Velocities

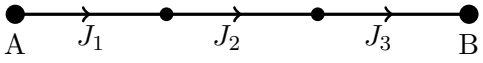

Figure 7.6: Three pipes in a row with slightly varying velocities

In this simple test case we want to illustrate the influence of the high order in the LTS scheme. We consider three pipes in a straight line, as depicted in Figure 7.6. The edges are all of the same length 1, but there is a variation in the width of the pipes. The diameters are $d_1 = d_3 = 10$ and $d_2 = 10\sqrt{5}$. Due to mass conservation the resulting velocities are given by $v_1 = v_3 = 1$ and $v_2 = \frac{1}{5}$, i.e. the variation in the width directly transfers to the velocities. The energy of the inflow is again given by (7.1). In Figure 7.7

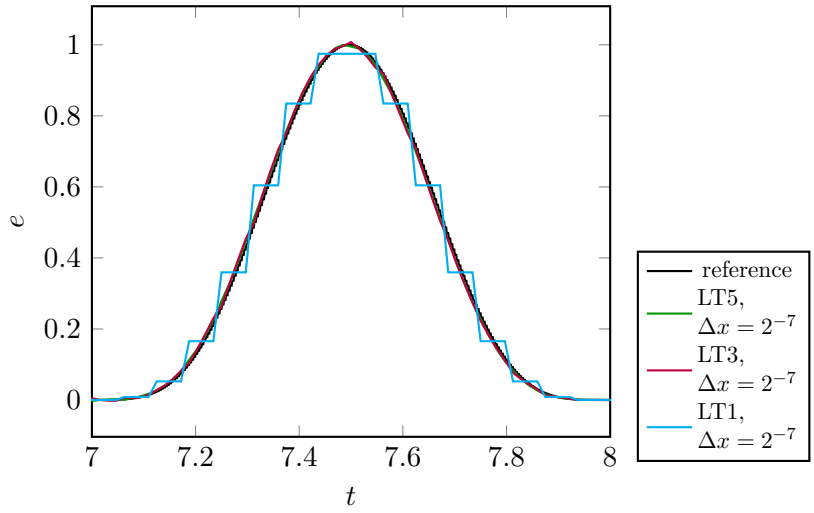

Figure 7.7: Energy in node B computed with third order schemes for different grid sizes

the numerical results for the temperature at the outflow of the third pipe is shown. At the second node the signal traveling through the pipe gets compressed. The much lower velocity in the second edge results in five cell values from the previous one filling one cell of the second edge. Thus the initial signal gets condensed in $\frac{1}{5}$th of the width. After the second edge, this signal stretches out again. The first order method just extends each cell into five identical new cell values. We can see in the figure, that this results in a discretization that is more course than the grid would allow. The third and fifth order methods however are able to recover the initial signal quite good. For the higher

order solutions, no big difference to the exact solution can be seen. This shows that apart from the better convergence in very fine grids, the high order schemes can also be beneficial for rather course grids. That effect gets larger, the larger the compression-decompression factors between the pipes are. If the CFL numbers are similar for all pipes, the different orders of the LTS will also behave more similar.

7.1.3 A District Heating Network

In the final test case we study the numerical algorithms on a more realistic setting for a district heating network. The layout of the system is shown in Figure 7.8. The supply branch (red) and the return branch (blue) consist of nine edges each. Each pipe has a diameter of 0.1 and the lengths are $L_1 = L_{18} = L_6 = L_{13} = 500$, $L_2 = L_{17} = L_5 = L_{14} = L_7 = L_{12} = L_9 = L_{10} = 282$, $L_3 = L_{16} = L_8 = L_{11} = 400$ and $L_4 = L15 = 424$.

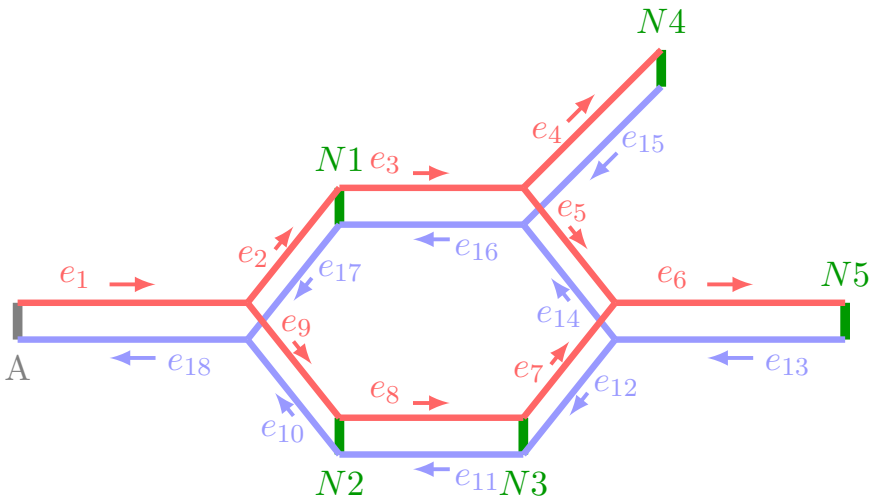

Figure 7.8: Graph of a district heating network

The five green vertical lines indicate the locations of the consumers. Each of them has a given energy demand $Q_1(t) = Q_2(t) = Q_3(t) = 3000$, $Q_4(t) = 6000$ and $Q_5(t) = 9000$, which enters into the mathematical model via equation (2.15). The thermal energy is provided by the boundary conditions for e_1 at the node A by

$$T_A(t) = \begin{cases} 120 & t < 3600 \\ 80 & t > 3600 \end{cases}.$$

At the same location the pressure of the supply branch is $p_1(t) = 3$ and $p_{18}(t) = 2$. In the houses thermal energy is consumed such that the temperature is cooled down to $T_{out} = 60$. The specific heat capacity is $c_p = 4160$ and the external cooling parameters are $k = 0.1$, $\lambda = 0.006$ with $T_{inf} = 20$.

In the numerical integrations spatial discretizations from $\Delta x = 40$ up to $\Delta x = 2.5$ are considered. The cell along the edges are equally spaced, i.e. Δx is slightly decreased if the length of the pipe is not a multiple of the given value. This leads to 8 up to 13 cells for the coarsest grids and 118 up to 200 cells on the fines level. The simulation is run up to the time $t_{end} = 36000$.

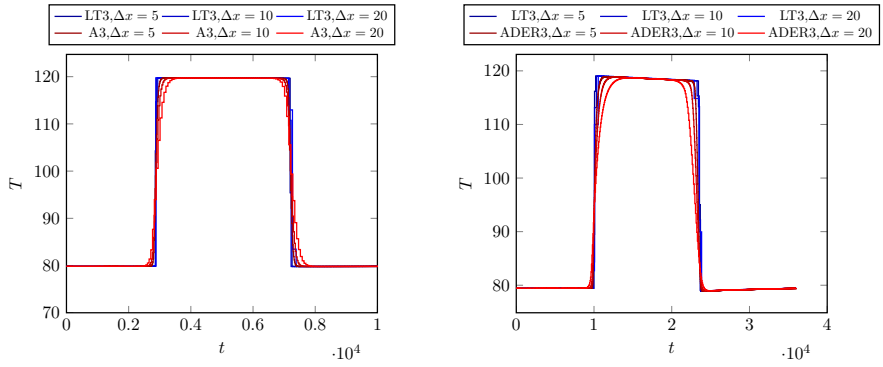

Figure 7.9: Temperature at the node $N1$ (left) and $N4$ (right)

In Figure 7.9 the evolution of the temperature in the node $N1$ and $N4$ are shown. As no flows mix, a part of the input signal passes almost unchanged. Just the slight cooling due to the outer temperature can be observed in $N4$. All numerical schemes give accurate results.

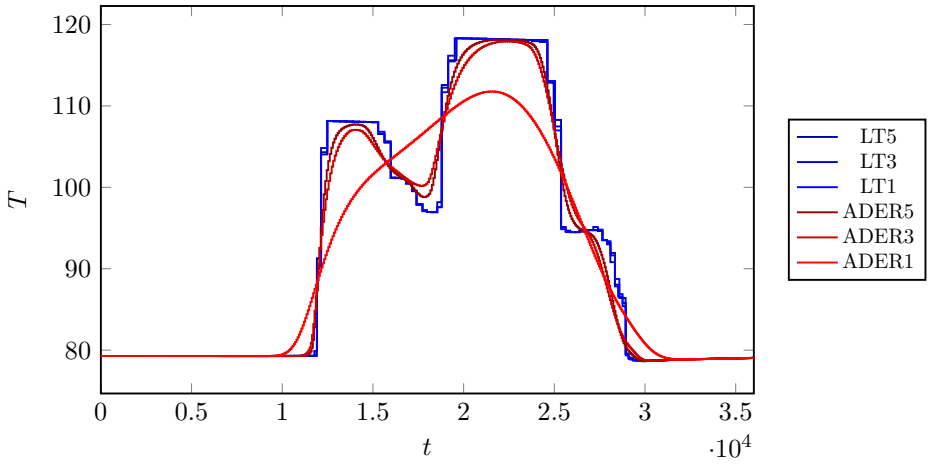

Figure 7.10: Temperature in $N5$ computed with schemes of different orders with $\Delta x = 20$

More complex is the behaviour of the temperature at node $N5$. In Figure 7.10 the results for the local time stepping scheme are compared to those of the ADER methods on a grid with $\Delta x = 20$. The temperature varies over time, since it is a mixture of waves passing the upper branch e_2, e_3 and e_5 with those of the lower branch e_9, e_8 and

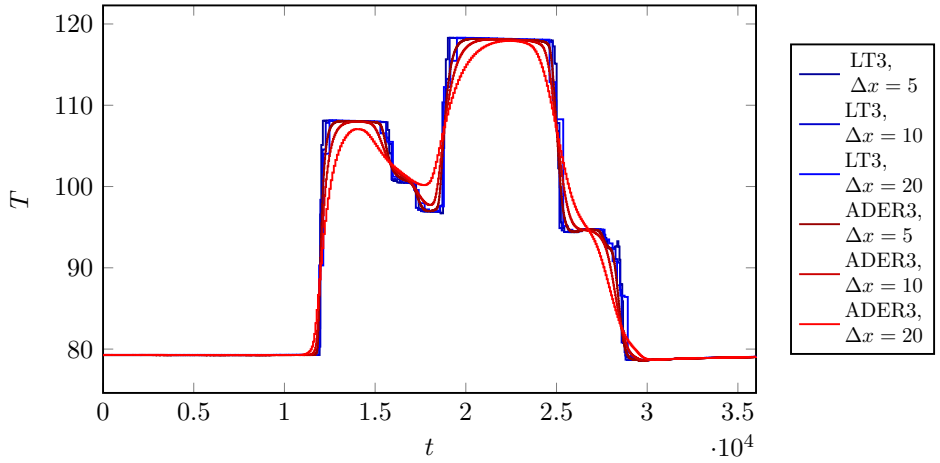

Figure 7.11: Temperature in $N5$ computed with schemes of order 3 for different spatial resolutions

e_7. The first order ADER scheme is not able to resolve these features. For the local time stepping method even the first order is more accurate than the fifth order ADER scheme. Figure 7.11 shows that all methods converge to the same solution as the grid is refined.

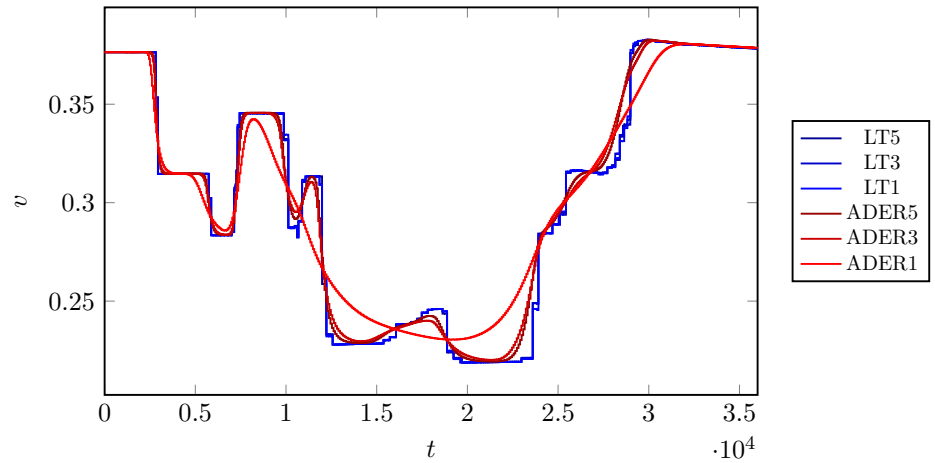

Figure 7.12: The velocity on edge e_1 for $\Delta x = 20$

Note that with differing values in T also influence the flow solver, i.e. also the velocity profile changes. In Figure 7.12 the velocity on edge e_1 is plotted. The values fluctuate over time as the inflow has to adjust according to the needs of all consumers simultaneously. Similar to the temperature, the first order ADER scheme can not capture all features of the solution, while the local time stepping methods provide very accurate results.

	ADER1		ADER3		ADER5	
Δx	L^1 error	time	L^1 error	time	L^1 error	time
40	4.76	4	2.80	6	-	-
20	4.01	9	1.75	16	1.28	20
10	3.26	19	0.98	45	0.66	57
5	2.54	44	0.52	126	0.32	185
2.5	1.88	109	0.26	446	0.14	664

Table 7.3: L^1-errors and computational times for the ADER schemes

	LT1		LT3		LT5	
Δx	L^1 error	time	L^1 error	time	L^1 error	time
40	1.17	7	1.15	7	1.16	8
20	0.61	13	0.65	13	0.63	14
10	0.30	24	0.30	25	0.28	28
5	0.19	47	0.18	48	0.17	54
2.5	0.11	96	0.10	97	0.09	118

Table 7.4: L^1-errors and computational times for the LT schemes

In the Tables 7.3 and 7.4 the L^1-errors and the computation times for the ADER and the local time stepping methods are shown. The reference solution is computed with the local time stepping scheme of order five on a grid with $\Delta x = 0.625$. The errors are calculated after transforming the time interval to $[0, 1]$.

For all numerical schemes the errors decrease and the computation times increase when the grid is refined. The errors of the local time stepping schemes are below those of the fifth order ADER method. Further the results are obtained in much shorter time. Note that the gain in terms of computation time is not as big as in the previous examples. This is due to the flow solver, which is identical for all methods. It occupies about 30 seconds on the finest grid. These costs could be reduced by exploiting the structure of the network or considering stationary version of (2.14) as in [44].

7.2 Numerical Experiments for Implicit Schemes

We now present the numerical results for the implicit schemes introduced in Chapter 5.

7.2.1 Shu's Linear Test

We start the numerical investigations with Shu's linear test case [37], a classical test in the community of hyperbolic conservation laws. It consists of scalar linear advection of a complicated initial condition. The exact setting is the following: We solve the linear

equation

$$u_t + u_x = 0, \qquad -1 < x < 1$$

with periodic boundary conditions and

$$u(0,x) = u_0(x) = \begin{cases} \frac{1}{6}\left(G(x,\beta,z-\delta) + G(x,\beta,z+\delta) + 4G(x,\beta,z)\right), & -0.8 \le x \le -0.6 \\ 1, & -0.4 \le x \le -0.2 \\ 1 - |10(x-0.1)|, & 0 \le x \le 0.2 \\ \frac{1}{6}\left(F(x,\alpha,a-\delta) + F(x,\alpha,a+\delta) + 4F(x,\alpha,a)\right), & 0.4 \le x \le 0.6 \\ 0, & \text{otherwise,} \end{cases}$$

where

$$G(x,\beta,z) = e^{-\beta(x-z)^2}$$
$$F(x,\alpha,a) = \sqrt{\max\left(1-\alpha^2(x-a)^2, 0\right)}.$$

The parameters are chosen as $a = 0.5,\ z = -0.z,\ \delta = 0.005,\ \alpha = 10,\ \beta = log(2)/36\delta$. The initial condition consists of a Gaussian, a square wave, a triangle wave and an ellipse wave, each relatively narrow with a width of 0.2. This test case is especially suited for investigating the ability of high order schemes to capture the sharp gradients involved and thus evolved to a classical benchmark for such problems.

Implicit Euler and Characteristic Scheme

First of all we want to demonstrate the behaviour of two first order schemes, the implicit upwind scheme and the characteristic scheme.

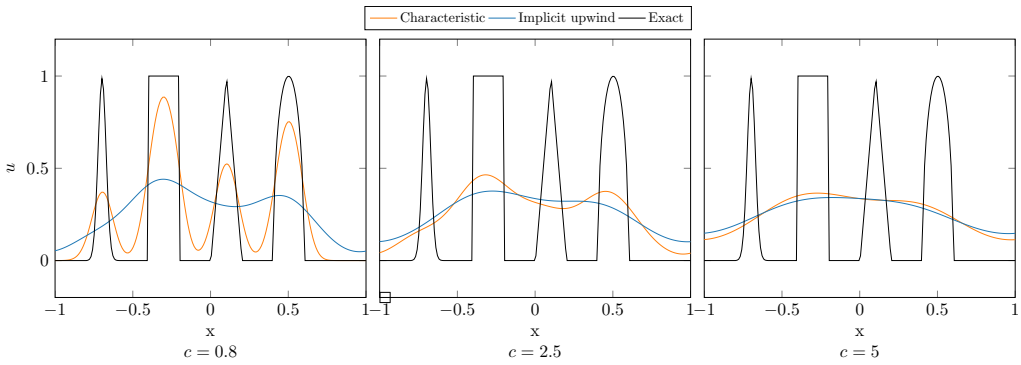

Figure 7.13: Shu's linear test for $n = 200$ with CFL numbers 0.8, 2.5 and 5 for the characteristic scheme (orange) and the implicit upwind scheme (blue). The exact reference solution is drawn in black.

In Figure 7.13 the results for the two schemes are compared to the exact solution. The piecewise definition of the characteristic scheme for CFL numbers smaller than 1 leads to considerably better results in the case $c = 0.8$. The result is not that surprising, since in the explicit regime the explicit upwind has lower numerical diffusion than its implicit

counterpart. When comparing both schemes in the implicit time stepping regime we can see, that they produce similar results. The characteristic scheme shows slightly better results for moderate time steps of $c = 2.5$. As we have shown in Section 5.2 the diffusion coefficient is proportional to $c - 1$ versus $c + 1$ in the implicit upwind case. This effect becomes smaller, the larger c gets, such that already for $c = 5$ the difference between the two becomes marginal. Most importantly we can see that for large CFL numbers the features of the exact solution get completely lost in both numerical approximations.

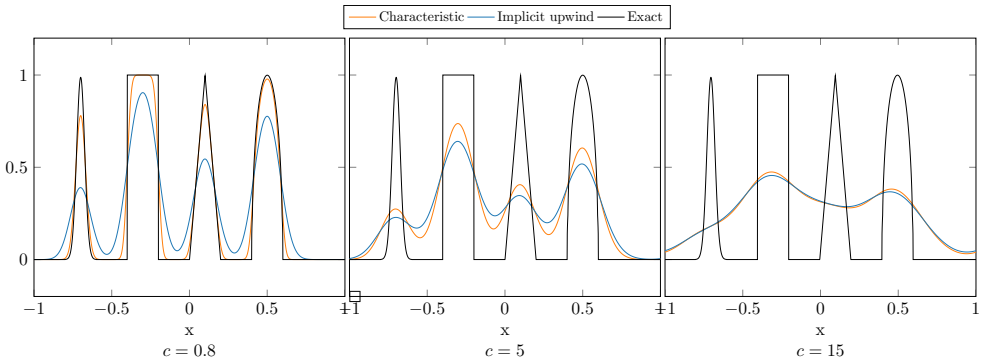

Figure 7.14: Shu's linear test for $n = 2000$ with CFL numbers 0.8, 5 and 15 for the characteristic scheme (orange) and the implicit upwind scheme (blue). The exact reference solution is drawn in black.

The tenfold refinement to $n = 2000$ does improve both schemes in the moderate CFL regime up to $c = 5$, but for very large time steps with $c = 15$ we again see the large numerical diffusion as well as the similar behaviour of the two schemes, see Figure 7.14. These results show the necessity of high order schemes when large time steps are applied.

High Order Schemes and Limiting

In Figure 7.15 the results for the unlimited schemes of different orders are shown. Due to their upwind formulation and the marching computation we cannot use periodic boundary conditions for these schemes. This is why the domain is extended in space and the initial datum is transported from $[-1, 1]$ to $[1, 3]$. As reference the exact solution is drawn in black and the implicit upwind scheme is shown in blue. The third order FV scheme used (green) is the one introduced in (5.9), the fourth order scheme (orange) is defined in (5.10). We can clearly see that by using the higher order schemes the solution gets much better in the sense that the four different features of the solution can be well distinguished. The third order scheme in green shows relatively smooth behaviour. It overshoots the exact height of the square wave and undershoots all areas between the features. At the two narrow kinks it does not reach the correct height. The fourth order scheme in contrast performs better in those extremely sharp parts and resolves those peaks very good. However, the fourth order scheme shows highly oscillatory behaviour, especially between the features. This gives rise to the need of limiters when applying

these schemes. In particular, the positivity of solutions is strongly violated by the high order methods. In the introduction of the a posteriori limiting in Section 5.5.3, we ended

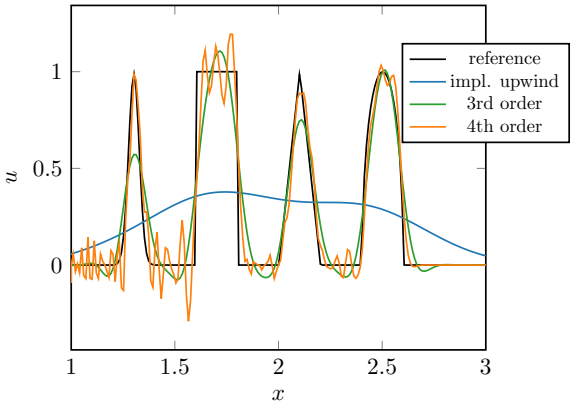

Figure 7.15: Shu's linear test with $n = 200, c = 2.5$ for unlimited schemes of different orders

up with several options, how the limiter could be applied. We now want to compare these four options of limiting in the same test cases as above.

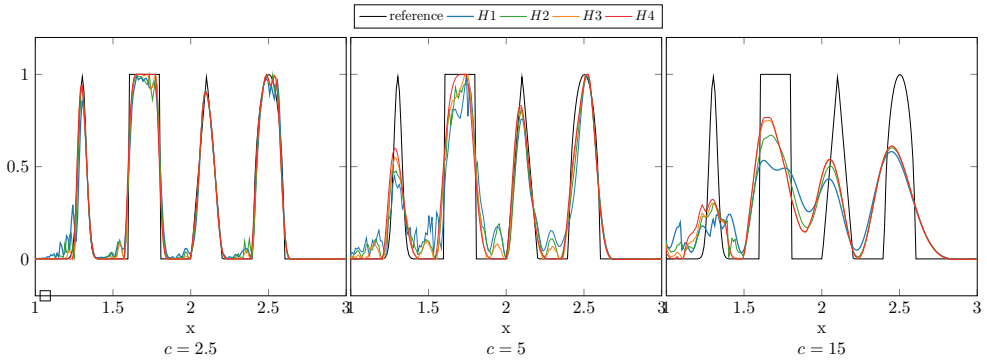

Figure 7.16: Shu's linear test for $n = 200$ with CFL numbers 2.5 and 5 for four different limiting variants

All of the four limited variants shown in Figure 7.16 perform better than the unlimited high order methods. In the left picture for $c = 2.5$, all four accurately recover the height of the features and clearly separates them. For both CFL numbers $c = 2.5$ and $c = 5$ we can observe some differences between the schemes. The methods $H1$ and $H2$ have relatively sharp spikes, especially in between the peaks of the exact solution. Furthermore, we observe that $H2$ seems slightly more accurate. The options $H3$ and $H4$ are a lot smoother and slightly more accurate with $H4$ providing the best results. All the schemes are able to completely avoid overshoots and stay within the min-max bounds of the initial condition. The test shows, that both optional additions to the limited scheme improve its accuracy. When adding the third order scheme to the cascade, the

result gets more accurate as it can be seen in the difference between $H1$ and $H2$ and in between $H3$ and $H4$. The addition of convex combinated fluxes ($H1 \to H3$ and $H2 \to H4$) smooths out the solution and also increases the accuracy. To conclude, the numerical method $H4$ incorporating the full cascade and the convex combination performs best. In the following numerical test cases we focus on that method and call it the *hybrid scheme*.

7.2.2 Convergence Analysis

In a similar test case, instead of Shu's initial condition we initialize with the smooth data

$$u_0(x) = \begin{cases} \sin^4(4\pi x), & \text{if } x \in [0, 0.5] \\ 0, & \text{else} \end{cases}$$

and transport it with $v = 1$ and $c = 5$ until $t_{end} = 0.5$. We test the convergence rate of three different schemes: the schemes $H2$ and $H4$ from the previous section and furthermore another variant $\tilde{H}$ where we do not use the fourth order FV scheme at all, but just the cascade $3 \to 1$. The results are shown in Figure 7.17.

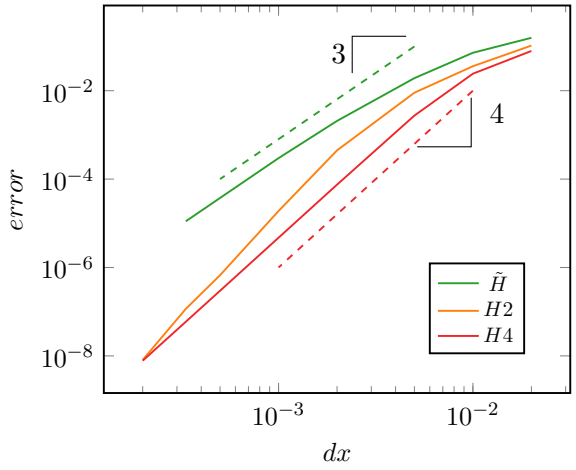

Figure 7.17: Error convergence plot

For smooth data and small enough discretization, the MOOD step does not trigger an order reduction in the computation. This means the scheme fully uses the high order fluxes for all calculations. We can see that all three schemes perform at their respective theoretical order of convergence.

7.2.3 Triangle Network

After the test cases on one dimensional segments, we apply the scheme on networks of advection equations. For applications with time varying velocities on networks, there is in general no explicit solution to verify the computations. In those cases a reference

solution with a very fine grid resolution is needed. With the example network from Section 3.1.1 we have access to operation conditions of the network, where an analytic solution can be constructed. This is why we first demonstrate the improvement of our hybrid scheme over the implicit upwind method in that setting. The topology and initial conditions and the velocity function have already been specified in Chapter 3.

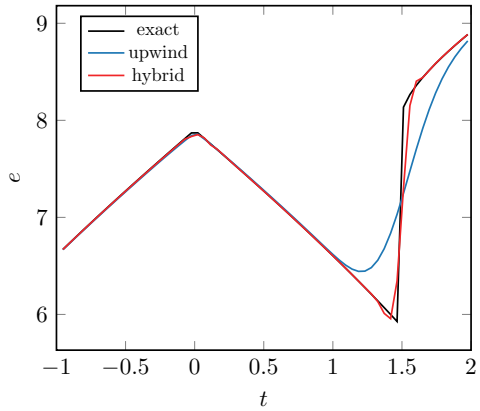

Figure 7.18: Temperature signal in node 2

The result of the hybrid scheme is compared to the implicit upwind scheme in Figure 7.18. Position and height of the discontinuity are well recovered by the hybrid scheme, while the upwind method is rather diffusive. Especially the values of the minimum at $t \approx 1.5$ are much worse for the latter.

7.2.4 Street Network

The street network is a part of a real district heating network. It contains 162 pipes and 32 consumers. The total pipe length is 1672m with the largest pipe measuring 80m and the shortest having 0.06m. Even though not being obvious from the topology plot Figure 7.19, it does contain one cycle near the label C_2. In the figure, we can see a temperature front traveling through the network. For the pipes, red color means hotter water, blue is cooler water. The dots represent the consumers with the color coding the mass flow from high (red) to low (blue). The solution shown was solved by an implicit upwind scheme, to demonstrate the smearing of the sharp front inside the network. The initial condition in the network was $T_0 = 80°C$ and the boundary condition at the source is set to

$$T_B = \begin{cases} 100°C, \text{if } t < 1000 \\ 80°C, \text{otherwise.} \end{cases}$$

We want to focus especially on the two consumers marked by red circles, C_1 and C_2.

Figure 7.20, displays the temperature signal in time at the rightmost node (C_1) of the network. The resolution for all schemes is $\Delta x = 0.5$m and $\Delta t = 5$s. Since we do not have an analytic description of the solution, the reference is computed by the upwind

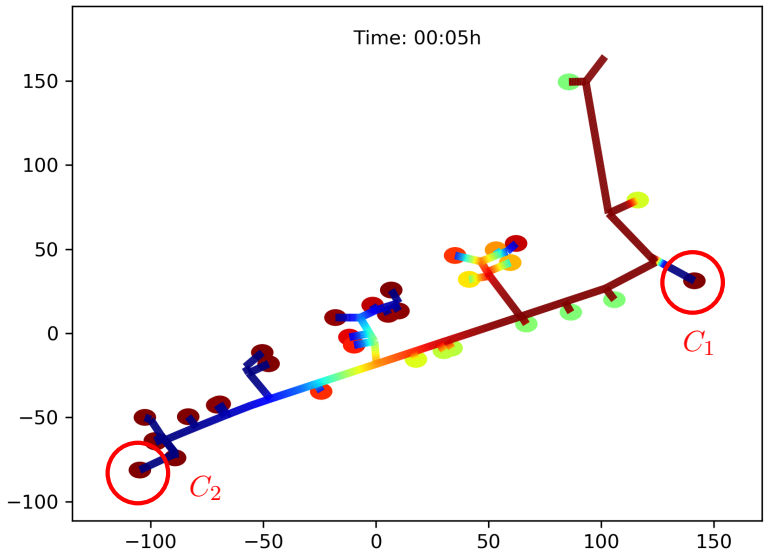

Figure 7.19: Street network topology

scheme with 50 times finer resolution ($\Delta x = 0.01$m and $\Delta t = 0.1$s). The plain fourth order scheme does not produce stable results due to the extremely high oscillations in front of the shock. The upwind scheme heavily smears out the discontinuity. The hybrid scheme accurately resolves position and height of the traveling front.

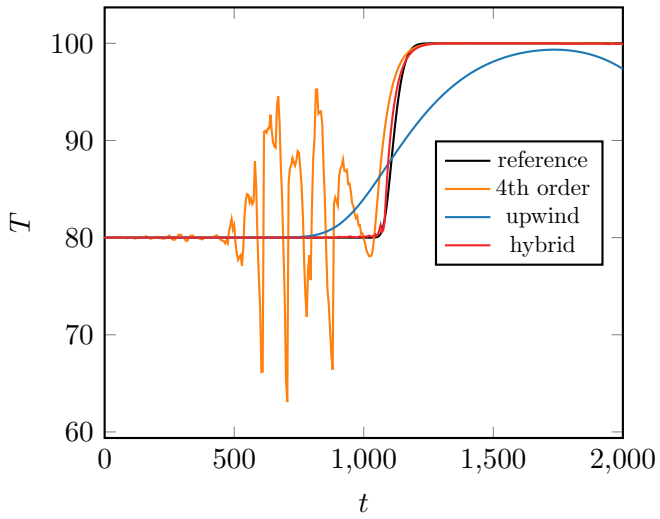

Figure 7.20: Temperature signal for consumer C_1

The advantage of the hybrid scheme becomes even more evident when looking at the solution at the consumer furthest away from the source (C_2). The more accurate we can compute the temperatures at the consumers, the better the resulting velocities for the next time steps. Therefore, large errors in the temperature may lead to completely different flow fields and consequently different looking solutions. This is even more important when there are circles in the network, that can possibly change their flow direction. Such a case can be seen in Figure 7.21. Several flow changes and different mixing ratios lead to a rather complicated temperature signal entering the consumer C_2. Compared to the reference solution, the upwind scheme loses most of the dynamics. The solution of the plain fourth order scheme is omitted, since it too oscillatory to give any meaningful contribution. The hybrid scheme accurately resolves all the kinks and waves we observe in the reference solution. For a precise simulation of such networks, it seems evident that high order approaches like the hybrid scheme are much more powerful than a straight forward first order approach.

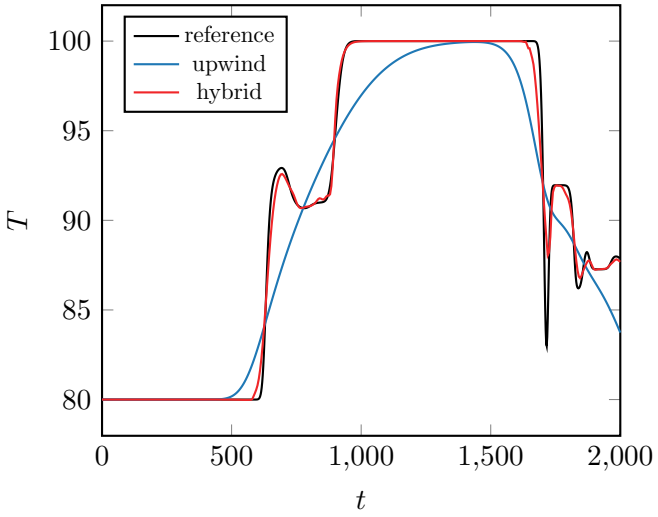

Figure 7.21: Temperature signal for consumer C_2

Chapter 8

Conclusion and Outlook

The scope of this work addressed two main topics. The first one is providing a suitable mathematical model describing the dynamics in district heating networks. The second one is the construction of efficient high order methods for its simulation.

A model has been set up by incompressible Euler equations on the edges of a graph. The individual equations are coupled by stating conservation of the involved quantities in all nodes. For the coupling of the flow velocities, a straight forward algebraic coupling is possible. Concerning the energy part, a small node volume had to be introduced, resulting in an ODE formulation of the energy conservation. That was necessary for the BV stability of solutions, where a counter example in the case of algebraic coupling was given. A splitting of the whole system of equations into a hydrodynamic part and an energy transport part was motivated. The advantage is, that specific properties of the two parts can be exploited, both in analytic considerations and on a numerical level.

The wellposedness of the mathematical model has been shown in three parts. The first two parts were dealing with the wellposedness of the hydrodynamics and the energy transport respectively. For the first one we could mainly rely on existing result applied to water supply networks. The second part is a new contribution, that yields powerful stability estimates under relatively weak regularity assumptions. Finally in the last step, the two separate systems are brought back together. We showed the unique existence of solutions to the coupled problem with Lipschitz continuous dependence on important system parameters. A main step in the proof of the analytic results was showing the stability of advection problems for just integrable velocity fields, without any assumptions on their sign. The result might also be of interest for other transport problems.

Many different numerical methods for the simulation of district heating networks were investigated. Hereby, the main focus was on high order methods. As a classical all purpose method for conservation laws, ADER schemes were presented. A coupling of the numerical fluxes on the nodes leads to schemes of arbitrary order on the network. The computational efficiency of explicit methods has been drastically increased by the construction of a local time stepping algorithm. The time steps were chosen in such a way, that the CFL condition is sharply satisfied and thus the upwind method produces

exact results along a pipe. The decoupling of the time steps led to a more complicated coupling algorithm, but a higher order reconstruction similar to the ADER scheme led to arbitrary order of the LTS scheme. It has been shown that efficient data structures resulted in a numerical method where computation time only scales linearly with the grid size, compared to the quadratic dependence for classical schemes. Thus, enormous speed ups were possible.

The still rather severe explicit CFL condition motivates the use of implicit schemes. Many aspects are more involved in the implicit case. We investigated two similar first order schemes, the implicit upwind and an implicit characteristic scheme. However, a second version of the characteristic scheme and an implicit formulation of the active flux method proved to be unstable. A generic construction of implicit high order schemes was described and their stability was assessed numerically. Hereby, the focus lied on methods using a pure upwind directed stencil. The big advantage of such schemes is the possibility to compute the solution by marching through the domain instead of solving an implicit system. Another difficult task for implicit methods is limiting. Many classical approaches result in implicit systems, that are hard to solve, or still rely on a small time step due to the high diffusion of the low order method. Instead we apply an *a posteriori* limiter to our implicit method. Several detection criteria are able to distinguish a smooth high order solution from artificial oscillations. A cascade of schemes with the orders four, three and one together with a convex combination of the fluxes of different orders leads to an accurate hybrid scheme. The theoretical convergence order and the behavior in different test cases was investigated. The hybrid scheme achieved good results in resolving the complicated solution structure of real district heating networks.

There are many possibilities in further researching this topic.
The introduction of an ODE formulation for the energy coupling was necessary for stability of solutions in our setting. This comes from the fact that infinite switches of flow directions can occur for general continuous velocities. For tree networks however, the solution to the hydraulic equations will never have a change in flow direction, if all consumer demands are positive. This raises the question whether there are other (more general) classes of networks, or specific choices of consumer behavior, for which one can have an a priori bound on the number of such switches.

Furthermore, the application of the obtained stability results to optimization problems is subject to further research.

The next step with the numerical simulations is their coupling to an optimizer. Since this work mainly focused on the numerical properties, a runtime optimized implementation in C++ needs to be done. When using automatic differentiation tools, the necessary sensitivities required by the optimizer can directly be evaluated. Alternatively, current research is directed towards an adjoint optimization approach. Due to the symmetry of system and adjoint for the advection equation, the same numerical method can be used to solve both parts.

Furthermore, due to the finite travel speed of information in the network, and the graph topology itself parallel computing can be used. Each time the flow splits into several parts at a node, all those edges can be computed simultaneously. Depending on the network structure, this can lead to significant speedup of the computation.

Bibliography

[1] Lorenzo Alliévi and Robert Dubs. *Allgemeine Theorie über die veränderliche Bewegung des Wassers in Leitungen*. Springer Berlin Heidelberg, 1909.

[2] Technische Werke Ludwigshafen am Rhein AG. Technische Anschlussbedingungen für die Versorgung mit Fernwärme aus den Versorgungsnetzen der Technischen Werke Ludwigshafen am Rhein AG. 2019.

[3] Todd Arbogast, Chieh-Sen Huang, and Xikai Zhao. Von neumann stable, implicit, high order, finite volume weno schemes. SPE Reservoir Simulation Conference, 04 2019. D011S002R004.

[4] Todd Arbogast, Chieh-Sen Huang, Xikai Zhao, and Danielle N. King. A third order, implicit, finite volume, adaptive runge–kutta weno scheme for advection–diffusion equations. *Computer Methods in Applied Mechanics and Engineering*, 368:113155, 2020.

[5] Stefan Banach. Sur les opérations dans les ensembles abstraits et leur application aux équations intégrales. *Fundamenta Mathematicae*, 3:133–181, 1922.

[6] Wasilij Barsukow. The active flux scheme for nonlinear problems. *Journal of Scientific Computing*, 86(1), December 2020.

[7] Bundesverband der Energie- und Wasserwirtschaft e.V. BDEW. *Leitfaden zur Abwicklung von Standardlastprofilen Gas*, 2016.

[8] A Benonysson. Dynamic modelling and operational optimization of district heating systems. Sep 1991.

[9] R. Borsche, M. Eimer, M. Garavello, and E. Rossi. Analysis of district heating networks. *(submitted 2021)*.

[10] R. Borsche, M. Eimer, and N. Siedow. A local time stepping method for thermal energy transport in district heating networks. *Applied Mathematics and Computation*, 353:215 – 229, 2019.

[11] Raul Borsche and Jochen Kall. ADER schemes and high order coupling on networks of hyperbolic conservation laws. *J. Comput. Phys.*, 273:658–670, 2014.

[12] Raul Borsche and Jochen Kall. High order numerical methods for networks of hyperbolic conservation laws coupled with ODEs and lumped parameter models. *J. Comput. Phys.*, 327:678–699, 2016.

[13] Walter Boscheri, Giacomo Dimarco, Raphaël Loubère, Maurizio Tavelli, and Marie-Hélène Vignal. A second order all mach number imex finite volume solver for the three dimensional euler equations. *Journal of Computational Physics*, 415:109486, 2020.

[14] Franck Boyer and Pierre Fabrie. *Mathematical Tools for the Study of the Incompressible Navier-Stokes Equations and Related Models*, volume 183. 11 2012.

[15] Enrique Cabrera, Jorge Garcia-Serra, and Pedro L. Iglesias. *Modelling Water Distribution Networks: From Steady Flow to Water Hammer*, pages 3–32. Springer Netherlands, Dordrecht, 1995.

[16] A. Cadiou and C. Tenaud. Implicit weno shock capturing scheme for unsteady flows. application to one-dimensional euler equations. *International Journal for Numerical Methods in Fluids*, 45(2):197–229, 2004.

[17] Yih-Nan Chen, Shih-Chang Yang, and Jaw-Yen Yang. Implicit weighted essentially non-oscillatory schemes for the incompressible navier–stokes equations. *International Journal for Numerical Methods in Fluids*, 31(4):747–765, 1999.

[18] Alexandre J. Chorin and Jerrold E. Marsden. *A Mathematical Introduction to Fluid Mechanics*. Springer New York, 1993.

[19] Erik Chudzik, Christiane Helzel, and David Kerkmann. The cartesian grid active flux method: Linear stability and bound preserving limiting. *Applied Mathematics and Computation*, 393:125501, 2021.

[20] Stéphane Clain, Steven Diot, and Raphaël Loubère. A high-order finite volume method for hyperbolic systems: Multi-dimensional Optimal Order Detection (MOOD). *Journal of Computational Physics*, February 2011.

[21] Stéphane Clain, Raphaël Loubère, and Gaspar J. Machado. a posteriori stabilized sixth-order finite volume scheme for one-dimensional steady-state hyperbolic equations. *Advances in Computational Mathematics*, 44(2):571–607, Apr 2018.

[22] A. Coddington and N. Levinson. *Theory of Ordinary Differential Equations*. International series in pure and applied mathematics. McGraw-Hill Companies, 1987.

[23] Rinaldo M. Colombo and Elena Rossi. Ibvps for scalar conservation laws with time discontinuous fluxes. *Mathematical Methods in the Applied Sciences*, 41(4):1463–1479, 2018.

[24] Pierre Degond and Min Tang. All speed scheme for the low mach number limit of the isentropic euler equations. *Communications in Computational Physics*, 10, 08 2009.

[25] S. Diot, S. Clain, and R. Loubère. Improved detection criteria for the multi-dimensional optimal order detection (mood) on unstructured meshes with very high-order polynomials. *Computers & Fluids*, 64:43 – 63, 2012.

[26] R. J. DiPerna and P. L. Lions. Ordinary differential equations, transport theory and sobolev spaces. *Inventiones Mathematicae*, 98(3):511–547, October 1989.

[27] Michael Dumbser, Cedric Enaux, and Eleuterio F. Toro. Finite volume schemes of very high order of accuracy for stiff hyperbolic balance laws. *Journal of Computational Physics*, 227(8):3971–4001, 2008.

[28] Michael Dumbser, Martin Käser, and Eleuterio F. Toro. An arbitrary high-order discontinuous Galerkin method for elastic waves on unstructured meshes – V. Local time stepping and p-adaptivity. *Geophysical Journal International*, 171(2):695–717, 2007.

[29] Michael Dumbser, Olindo Zanotti, Raphaël Loubère, and Steven Diot. A posteriori subcell limiting of the discontinuous galerkin finite element method for hyperbolic conservation laws. *Journal of Computational Physics*, 278:47–75, 2014.

[30] Matthias Eimer, Raul Borsche, and Norbert Siedow. Implicit finite volume method with a posteriori limiting for transport networks. *Advances in Computational Mathematics*, 48:21, 2022.

[31] Timothy Eymann and Philip Roe. Multidimensional active flux schemes. *21st AIAA Computational Fluid Dynamics Conference*, 06 2013.

[32] Fischer-Uhrig, Ingenieurbüro. STANET Netzberechnung. `http://stafu.de/de/home.html`, 2021. Accessed: 2021-05-10.

[33] Martin Gugat and Stefan Ulbrich. The isothermal euler equations for ideal gas with source term: Product solutions, flow reversal and no blow up. *Journal of Mathematical Analysis and Applications*, 454(1):439–452, 2017.

[34] Sarah-Alexa Hauschild, Nicole Marheineke, Volker Mehrmann, Jan Mohring, Arbi Moses Badlyan, Markus Rein, and Martin Schmidt. Port-hamiltonian modeling of district heating networks. In Timo Reis, Sara Grundel, and Sebastian Schöps, editors, *Progress in Differential-Algebraic Equations II*, pages 333–355, Cham, 2020. Springer International Publishing.

[35] Chieh-Sen Huang and Todd Arbogast. An implicit eulerian–lagrangian WENO3 scheme for nonlinear conservation laws. *Journal of Scientific Computing*, 77(2):1084–1114, May 2018.

[36] Lennart Jansen and Jonas Pade. Global unique solvability for a quasi-stationary water network model. *Preprint series: Institut für Mathematik, Humboldt-Universität zu Berlin*, 2013-11, 2013.

[37] Guang-Shan Jiang and Chi-Wang Shu. Efficient implementation of weighted ENO schemes. *Journal of Computational Physics*, 126(1):202–228, June 1996.

[38] Shou jun Zhou, Mao cheng Tian, You en Zhao, and Min Guo. Dynamic modeling of thermal conditions for hot-water district-heating networks. *Journal of Hydrodynamics*, 26(4):531–537, August 2014.

[39] Dieter Jungnickel. *Graphs, Networks and Algorithms*. Springer Berlin Heidelberg, 2013.

[40] Jochen Kall. *ADER Schemes for Systems of Conservation Laws on Networks*. PhD thesis, Technische Universität Kaiserslautern, 2016. Verlag Dr. Hut.

[41] Grant Keady. Colebrook-white formula for pipe flows. *Journal of Hydraulic Engineering*, 124(1):96–97, January 1998.

[42] Sergiu Klainerman and Andrew Majda. Singular limits of quasilinear hyperbolic systems with large parameters and the incompressible limit of compressible fluids. *Communications on Pure and Applied Mathematics*, 34(4):481–524, 1981.

[43] Sergiu Klainerman and Andrew Majda. Compressible and incompressible fluids. *Communications on Pure and Applied Mathematics*, 35(5):629–651, 1982.

[44] Ralf Köcher. *Beitrag zur Berechnung und Auslegung von Fernwärmenetzen*. PHD Thesis, Technische Universität Berlin, 2000.

[45] T. Köppl, B. Wohlmuth, and R. Helmig. Reduced one-dimensional modelling and numerical simulation for mass transport in fluids. *Internat. J. Numer. Methods Fluids*, 72(2):135–156, 2013.

[46] E. Krause, H. Schlichting, H.J. Oertel, and K. Gersten. *Grenzschicht-Theorie*. Springer, 2006.

[47] Dmitri Kuzmin, Rainald Löhner, and Stefan Turek, editors. *Flux-Corrected Transport*. Springer Netherlands, 2012.

[48] Randall J LeVeque. *Finite-Volume Methods for Hyperbolic Problems*, volume 31. Cambridge university press, 2002.

[49] P.L. Lions. *Mathematical Topics in Fluid Mechanics: Volume 1: Incompressible Models*. Mathematical Topics in Fluid Mechanics. Oxford University Press, Incorporated, 1996.

[50] Axel Klar Mapundi K. Banda, Michael Herty. Coupling conditions for gas networks governed by the isothermal euler equations. *Networks and Heterogeneous Media*, 1(2):295–314, 2006.

[51] Jan Mohring, Dominik Linn, Matthias Eimer, Markus Rein, and Norbert Siedow. *District Heating Networks – Dynamic Simulation and Optimal Operation*, pages 303–325. Springer International Publishing, Cham, 2021.

[52] Matthias Möller, Dmitri Kuzmin, and Stefan Turek. Implicit flux-corrected transport algorithm for finite element simulation of the compressible euler equations. In *Scientific Computation*, pages 325–354. Springer Berlin Heidelberg, 2004.

[53] Lucas O Müller and Pablo J Blanco. A high order approximation of hyperbolic conservation laws in networks: Application to one-dimensional blood flow. *Journal of Computational Physics*, 300:423–437, 2015.

[54] Lucas O. Müller, Pablo J. Blanco, Sansuke M. Watanabe, and Raúl A. Feijóo. A high-order local time stepping finite volume solver for one-dimensional blood flow simulations: application to the ADAN model. *International Journal for Numerical Methods in Biomedical Engineering*, 32(10):e02761, January 2016.

[55] National Institute of Standards and Technology. *Thermophysical Properties of Fluid Systems*, 2016.

[56] Daniele Antonio Di Pietro and Alexandre Ern. *Mathematical Aspects of Discontinuous Galerkin Methods*. Springer Berlin Heidelberg, 2012.

[57] PSI Software AG. PSIcontrol Netzberechnung, website, 2021, Accessed: 2021-05-10 https://www.psienergy.de/de/loesungen/netzleittechnik/netzberechnungen/gas-fernwaerme-wassernetze/.

[58] Markus Rein. *Order reduction for nonlinear dynamic models of district heating networks*. PhD thesis, TU Kaiserslautern, 2020.

[59] Markus Rein, Jan Mohring, Tobias Damm, and Axel Klar. Parametric model order reduction for district heating networks. *PAMM*, 18(1), December 2018.

[60] Markus Rein, Jan Mohring, Tobias Damm, and Axel Klar. Optimal control of district heating networks using a reduced order model. *Optimal Control Applications and Methods*, 41(4):1352–1370, 2020.

[61] Markus Rein, Jan Mohring, Tobias Damm, and Axel Klar. Model order reduction of hyperbolic systems focusing on district heating networks. *Journal of the Franklin Institute*, May 2021.

[62] Arne Roggensack. *Low Mach number equations with heat source on networks*. PhD thesis, Universität Hamburg, 2014.

[63] Leonardo Scandurra. *Numerical Methods for All Mach Number flows for Gas Dynamics*. PhD thesis, 02 2017.

[64] Hermann Schlichting and Klaus Gersten. *Grenzschicht-Theorie*. Springer-Verlag, 2006.

[65] Chi-Wang Shu. *Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws*, pages 325–432. Springer Berlin Heidelberg, Berlin, Heidelberg, 1998.

[66] Chi-Wang Shu and Stanley Osher. Efficient implementation of essentially non-oscillatory shock-capturing schemes. *Journal of Computational Physics*, 77(2):439–471, 1988.

[67] P Steinle and R Morrow. An implicit flux-corrected transport algorithm. *Journal of Computational Physics*, 80(1):61–71, 1989.

[68] Sirui Tan and Chi-Wang Shu. Inverse lax-wendroff procedure for numerical boundary conditions of conservation laws. *Journal of Computational Physics*, 229(21):8144–8166, 2010.

[69] E. F. Toro, R. C. Millington, and L. A. M. Nejad. *Towards Very High Order Godunov Schemes*, pages 907–940. Springer US, Boston, MA, 2001.

[70] Eleuterio F Toro. *Riemann Solvers and Numerical Methods for Fluid Dynamics: A Practical Introduction*. Springer, 2009.

[71] B. van der Heijde, A. Aertgeerts, and L. Helsen. Modelling steady-state thermal behaviour of double thermal network pipes. *International Journal of Thermal Sciences*, 117:316 – 327, 2017.

[72] B. van der Heijde, M. Fuchs, C. Ribas Tugores, G. Schweiger, K. Sartor, D. Basciotti, D. Müller, C. Nytsch-Geusen, M. Wetter, and L. Helsen. Dynamic equation-based thermo-hydraulic pipe model for district heating and cooling systems. *Energy Conversion and Management*, 151:158–169, November 2017.

[73] Bram van Leer. Towards the ultimate conservative difference scheme. v. a second-order sequel to godunov's method. *Journal of Computational Physics*, 32(1):101–136, 1979.

[74] Lisa Wagner. *Second-Order Implicit Methods for Conservation Laws with Applications in Water Supply Networks*. PhD thesis, Technische Universität, Darmstadt, 2018.

[75] W. Wagner and A. Pruß. The IAPWS formulation 1995 for the thermodynamic properties of ordinary water substance for general and scientific use. *Journal of Physical and Chemical Reference Data*, 31(2):387–535, June 2002.

[76] Pieter Wesseling. *Principles of Computational Fluid Dynamics*. Springer Berlin Heidelberg, 2001.

[77] Martin Stoll Peter Benner Yue Qiu, Sara Grundel. Efficient numerical methods for gas network modeling and simulation. *Networks & Heterogeneous Media*, 15(4):653–679, 2020.

[78] Steven T Zalesak. Fully multidimensional flux-corrected transport algorithms for fluids. *Journal of Computational Physics*, 31(3):335–362, 1979.

[79] Olindo Zanotti, Francesco Fambri, Michael Dumbser, and Arturo Hidalgo. Space–time adaptive ader discontinuous galerkin finite element schemes with a posteriori sub-cell finite volume limiting. *Computers & Fluids*, 118:204–224, 2015.

[80] Nelida Črnjarić Žic and Bojan Crnković. High order accurate semi-implicit weno schemes for hyperbolic balance laws. *Applied Mathematics and Computation*, 217(21):8611–8629, 2011.

Curriculum vitae

Matthias Eimer

Personal information

Place of birth	Kaiserslautern, Germany
Nationality	German

Education

10/2017 - 10/2021	PhD studies TU Kaiserslautern Fraunhofer Institute for Industrial Mathematics
10/2015 - 09/2017	Master of Science in Mathematics TU Kaiserslautern
10/2012 - 09/2015	Bachelor of Science in Mathematics TU Kaiserslautern
10/2011 - 09/2012	Correspondance studies Mathematics & Physics
03/2012	A-levels, Burggymnasium Kaiserslautern

Lebenslauf

Matthias Eimer

Persönliche Daten

Geburtsort	Kaiserslautern
Nationalität	deutsch

Education

10/2017 - 10/2021	Doktorand TU Kaiserslautern Fraunhofer-Institut für Techno- und Wirtschaftsmathematik ITWM
10/2015 - 09/2017	Master of Science Mathematik TU Kaiserslautern
10/2012 - 09/2015	Bachelor of Science Mathematik TU Kaiserslautern
10/2011 - 09/2012	Fernstudium Mathematik & Physik
03/2012	Abitur, Burggymnasium Kaiserslautern

This thesis tackles the modeling and efficient simulation of district heating networks. The dynamical behavior of water in the pipes is described by the incompressible Euler equations. Conservation of the involved quantities determines the coupling at each junction in the network, leading to a system of partial differential algebraic equations. For that system, the unique existence of a solution is shown.

Furthermore, a stability estimate for the dependence on important parameters is derived. A new local time stepping algorithm is presented that is perfectly suited for the solution of the transport problem involved. In comparison to generic high order ADER schemes its high efficiency outperforms the classical approach significantly. In order to enable computation of very large time steps, implicit methods are investigated. A high order finite volume method is equipped with an a posteriori limiter. The superior behavior of the constructed hybrid scheme is shown in different numerical tests and applications. The obtained results build an important foundation for the upcoming optimization problems.

ISBN 978-3-8396-1853-0

Fraunhofer VERLAG

9 783839 618530