Optimization and Control of Traffic Flow Networks

Vom Fachbereich Mathematik
der Technischen Universität Kaiserslautern
zur Erlangung des Grades eines
Doktors der Naturwissenschaften
(Dr. rer. nat.)
genehmigte

Dissertation

von

M. Sc. Anita Kumari Singh

aus Indien

Referent :	Prof. Dr. A. Klar
Koreferent :	Prof. Dr. P. Spellucci
Tag der mündlichen Prüfung :	18. September 2006

Kaiserslautern 2006
D 386

Dedicated to
my beloved
Parents

Acknowledgements

I would like to express my deep sense of gratitude and indebtedness to my thesis advisor **Prof. Dr. Axel Klar** for his continual encouragement and patient guidance throughout the course of this work.

Words are inadequate to express my thanks to **Dr. Michael Herty** for his enthusiastic guidance, constructive comments and valuable suggestions for the successful completion of this work.

My sincere thanks and special reference to **Dr. Mohammed Seaid** for his intangible support and ready cooperation at each and every step of this work. His invaluable advises and encouragement throughout the work were extremely helpful.

I am thankful to the authorities of University of Technology Darmstadt and University of Technology Kaiserslautern for providing me all infrastructure facilities throughout my course of study. Without the financial and other related support of **GK (GradieurtenKolleg TU Darmstadt)** and **DFG TU Kaiserslautern**, i would not have got a wonderful opportunity to work for my PhD. I thank one and all for making my dream possible.

I take this opportunity to sincerely express my gratitude to my beloved brothers and husband (Dr. Prabhat Kumar) without them my emotional existence would not have been possible during my stay at Darmstadt and Kaiserslautern in Germany.

I also thank the members, past and present, and all my co-workers in the Department of Mathematics, Technical University Darmstadt and Technical University Kaiserslautern who have for last three years made my stay so enjoyable. I will fail in my duty if I did not express my thanks to **Frau Semler** for all her help and understanding. I would like to avail of this opportunity to thank all **my friends** for their continuous support, motivation and encouragement during my stay in Germany.

Anita Kumari Singh

Contents

Introduction

The increasing need for transportation and mobility leads to a fast growth of traffic in many industrialized countries. Challenging economical and scientific problems are due to this fact and motivates intense research in this field. Mathematical models can provide an understanding of dynamics of the traffic and give insight into questions like – what causes congestion, what determines the time and location of traffic break down, how does a congestion propagate. The objective of applied mathematicians and engineers has been to develop traffic models in order to predict the evolution of traffic flow. This in turn helps in answering how to handle urgent traffic issues and supports strategies of organizing traffic flow. In addition, the organized traffic may reduce the travel time due to an optimized traffic distribution.

The existing literature is vast and characterized by various contributions taking into account modeling aspects, qualitative analysis of the existing models and simulations related to applications. Although, each and every aspect cannot be cited, however a brief overview of the intensive research is presented herein. Traffic flow models and related theories have been developed since the last fifty years. Various types of models differing on the level of description, applications and needs have been considered and discussed among mathematicians, physicists and engineers.

The most basic models are **microscopic models** describing the evolution of each vehicle under the influence of its leading vehicle. These models are being represented in terms of a large system of ordinary differential equations, for example in [16, 32, 40, 77, 78]. At the second level are **kinetic models** involving Boltzmann type equations for the phase space distribution function $f(x, v, t)$, which describes the number of vehicles at a position x, time t and velocity v, [46, 53, 54, 57, 60, 80, 86, 87, 97]. Analogous to fluid dynamics, **macroscopic models** based on the conservation laws (partial differential equations) have been proposed by many authors, see in [3, 38, 52, 68, 81, 89]. Kinetic models form the bridge between microscopic and macroscopic models. There exists a lot of research on the derivation of one model from the other. For example, derivation of macroscopic equations for density and velocity from kinetic equations has been shown in [36, 39, 53, 60]. In addition to this a connection between microscopic follow-the-leader model and continuum traffic flow models has been established in [55]. On the other side a derivation of a kinetic model based on stochastic microscopic model has been presented in [97].

Recently, few survey papers giving an overview of different types of mathematical models

have been presented [7, 56]. The paper [56] is mainly devoted to models based on kinetic equations and a concise description of macroscopic and microscopic models is given. In the review [7] authors have reported an overview of different methodological aspects of mathematical modeling and some specific macroscopic and kinetic models. The present modeling yet lacks to correctly capture the complex dynamics of the phenomena related to traffic flow. Authors have presented a critical analysis of different models which are available in the literature and possible research perspectives in order to develop new models free from existing flaws.

In general authors have treated the traffic flow on the different lanes of a road by averaging. However, this simplification is not applicable if there is disequilibrium in neighboring lanes. Therefore, **multilane traffic flow modeling** has been introduced on all three levels of models by various authors, for example [41, 46, 58]. Helbing et al. deduced a macroscopic model of traffic flow on unidirectional roads with multiple lanes from gas kinetic equations [41] including velocity fluctuation, lane changing, overtaking etc. Whereas Klar et al. have introduced a microscopic multilane model based on reaction thresholds of drivers and an Enskog-like kinetic multilane model including lane-change probabilities [58]. This derivation of multilane models is supplemented with the numerical results in [59].

The main focus of the present work is on macroscopic models that have been used successfully in the past years. Macroscopic models describe the traffic flow by continuous aggregate functions like average density, velocity and flow in the space-time domain. The dynamics of traffic flow is usually modeled by a nonlinear system of two or three partial differential equations (PDE). Most of models describe dynamics of the macroscopic variables on single unidirectional roads. The modeling and simulation of traffic flow models for road networks and optimal control problems for these models have been studied.

Research on the macroscopic traffic flow modeling started when Lighthill-Whitham and Richards (LWR) proposed a model based on the analogy of vehicular traffic flow and flow of particles in fluid, [68, 89]. These flow models deal with aggregate variables (macroscopic quantities). The basic assumption is that there exists an equilibrium speed-density relation. The governing equation is a nonlinear hyperbolic partial differential equation. It can explain formation of shock waves corresponding to congestion. Despite of this remarkable property, the LWR model fails to describe more complicated traffic flow patterns, e.g., stop-and-go traffic. This is due to the unrealistic assumption of an dependence between equilibrium speed and density. In order to overcome this, models consisting of an additional equation to describe the acceleration behavior were developed by many authors [3, 38, 81]. In [27] the author presented critics on each of these models based on applicability and invalidity in the different traffic flow situations.

Besides the growing understanding of single lane models the discussion of traffic models for road networks is fairly a new field. There exists only a few publications on this topic [22, 43, 45].

This thesis is organised in the following way:

The preliminary work is presented in **chapter 1**, which involves a discussion of different modeling aspects and investigation of coupling conditions necessary to model the traffic flow on road networks. A major drawback of macroscopic models based on partial differential equations (PDE models) is their high computational cost. Therefore, the emphasis is on models including more features of the PDE model but of significantly low computational costs. The latter point is important for optimizing large networks in real-time. A model based on ordinary differential equation (ODE) is derived from the PDE model for networks. The ODE model is obtained by using coarse spatial approximations. The ODE model consists of a system of coupled ordinary differential equations with boundary conditions at junctions. The flow in the network can be distributed by a control α at some designated junctions. It is shown that this model inherits similar features of the PDE model, e.g. traffic jam propagation. Furthermore, a simplified model based on algebraic equations is briefly reviewed, [44]. In addition, simplified algebraic model is reformulated again by neglecting traffic dynamics and reduced to a model with only linear boundary conditions, see section 1.5. This fact is useful while solving an optimization problem on road networks.

In **chapter 2**, cost functionals and corresponding optimization problems are defined which help to organize and route the traffic through a network. One of the optimization problem is related to finding a shortest path through the network subject to physical road conditions. The optimal control problem defined in the setting of the ODE model is a nonlinear bound constrained problem. The optimization problem corresponding to the reformulated simplified algebraic (RSA) model is a nonlinear optimization problem with linear and bound constraints. For an analysis and the numerical solution of the optimal control problem, optimization methods using gradient and Hessian information are used. In certain situations the evaluation of gradient is not straightforward and a adjoint calculus is used to compute gradients in such situations and even when the cost functional is subject to a set of state equations which may be differential equations. The adjoint equations corresponding to the ODE model are derived. Then algorithms are stated to solve the ODE bound constrained optimization problem. Details of classical optimization algorithms for the unconstrained problem supplemented with projection onto feasible sets for bound constraints are explained. For example, simple steepest descent methods, projected quasi-Newton methods, the limited-memory BFGS method L-BFGS-B by [15].

In **chapter 3** another method is described to solve the equality and bound constrained optimization problems formulated in chapter 2 and possibly further problems of this type. The constrained problem is transformed into a bound constrained exact penalty problem by adding equality constraints to the cost functional with the aid of exact penalty functions involving a penalty parameter $\beta > 0$. In the literature the exact penalty methods have been under investigation for several years, see [9, 13, 37, 72, 82, 100]. The definition, properties and conditions characterizing the exact penalty problem can be found

in [17, 23, 83, 84]. The optimal choice of penalty parameter has been an open question. Theoretically the threshold value for the penalty parameter is defined in terms of Lagrange multipliers λ^* for equality constraints [17, 48], but this involves knowledge of local minimizer of original constrained problem. However, an estimate for the threshold value for the penalty parameter can be given. The l_1-exact penalty function is used for investigation. The evident difficulty is the non-differentiability of the l_1-penalty function. The transformed bound constrained exact penalty problem cannot be solved using classical gradient based methods. In order to surmount this difficulty smooth approximations involving smoothing parameter α for the l_1- penalty function are defined. A smoothed bound constrained penalty problem parameterized by penalty and smoothing parameters is given and solved by a bound constrained solver At this point the complete algorithm is defined which provides an extension to work done by P. Spellucci [94]. Numerical results of the application of this algorithm to different test problems are presented in chapter 4.

In **chapter 4** numerical results including simulation and optimization results of different traffic flow network models as mentioned in chapter 1 are presented. In the first part, the simulation results of macroscopic PDE and ODE models are compared to estimate the quality of the ODE model. An academic test–network is introduced which has the property that the optimal controls α_i are known in advance. It is observed that the ODE model is a good approximation of the PDE model for free and congested traffic flow through this network. Moreover, the optimal controls coincide for a variety of traffic situations. The solution to the ODE model is computationally cheaper than the PDE model, see section 4.1.3. The gradient computed by solving the adjoint equations and by finite difference approximations is compared. However, the derivation of the adjoint calculus and results remain true for arbitrary networks. Finally different numerical optimization algorithms are compared and adapted to solve the nonlinear bound constrained optimal control problem for the ODE model.

In the second part of this chapter the equality and bound optimization problem of the reformulated simplified algebraic model is solved. The solution method consist of smoothed non-differentiable exact l_1-penalty function involving penalty and smoothening parameter in the framework of the algorithm discussed in chapter 3. Numerical experiments are performed for different choices of the initial penalty parameter and different smoothing approximation. Results are compared on using two bound constrained subproblem solvers namely, L-BFGS-B and PL2. It is found that the PL2 converges to the optimal solution for less number of iterations for all sizes of the test-network.

In **chapter 5** scalar conservation laws with discontinuous flux function are considered. In context of traffic flow, this conservation law is the LWR model, but with a flux as a function of the space variable x in addition to the car-density ρ. Such situations occur in the real-world when the total number of lanes on certain sections of the road changes due to changing weather conditions or construction site. Theoretical results have been proved on the existence and uniqueness of weak solutions for this class of conservation laws [33, 61, 96]. In [96] Towers presented a scalar finite-difference scheme based on Godunov

or Engquist-Osher numerical flux and the algorithm uses a scalar Riemann solver under the restriction that the flux function is concave. The application of this method requires staggered discretization. Moreover, there has been a lot of numerical treatment of discontinuous flux functions for hyperbolic equations, see for example [1, 74]. In [50] Karlsen et al. have derived a relaxation scheme and proved the convergence of this scheme. This scheme is supplemented by the development of characteristic based relaxation method by Seaid [92]. Here, a new domain decomposition method is introduced for solving such conservation laws. This method uses similar techniques recently used to treat junctions in networks. The whole domain is decomposed at the point of discontinuity of flux function. This yields continuous conservation laws on every subdomain and in addition special coupling conditions at the subdomain interfaces. The numerical performance of the method is illustrated on models with concave flux function and also, when the flux function is neither convex nor concave.

Finally the thesis is summarized and an outlook on possible extensions to this work is presented in **chapter 6**.

Two appendices are included in this dissertation. In **appendix A** the continuous adjoint equations are derived for a general minimization problem subject to state equations given in differential form. Gradients of the cost functional of the optimization problem are obtained by approximately solving adjoint equations for adjoint variables and using gradient equations which is also in integral form. In **appendix B** a brief review of a relaxation scheme introduced by Jin et al. [49] is presented. This method is used to solve the conservation laws with continuous flux functions in each subdomain.

Following articles from this work have been published or will be published in proceedings and journals:

- A. K. Singh, M. Herty and A. Klar, *Flow optimization on traffic networks*, PAMM (Proceedings of GAMM): GAMM 75^{th} Annual Scientific Conference, Vol. 4:264, (2004).
- M. Herty , A. Klar and A. K. Singh, *An ODE traffic network model*, to appear in Journal of Comp. Appl. Math., (2006).
- M. Herty, A. Klar, A. K. Singh, P. Spellucci, *Smoothed penalty algorithms for optimization of nonlinear models*, to appear in Journal of COAP, (2006).
- M. Herty, A. K. Singh, P. Spellucci, *Applied optimization of vehicular traffic flow using smoothing and penalization*, under review, (2005).
- M. Herty, M. Seaid, A. K. Singh, *Domain decomposition for conservation laws with discontinuous flux functions*, to appear in Journal of Appl. Num. Math., (2006).

Chapter 1

Models for Traffic Flow on Road Networks

This chapter is intended to give a brief overview of the basic models of vehicular traffic in networks of highways and the necessary coupling boundary conditions defining flow through junctions in the network. A new traffic flow model for networks as a simplification of a common macroscopic traffic flow model for networks is presented.

1.1 Introduction

Nowadays, there exists a broad range of traffic flow models describing the different features and properties of vehicular traffic flow. The class of so–called macroscopic models describe the traffic flow in terms of averaged quantities like density denoted by ρ or flow ρv. These models are written in terms of evolution equations which are obtained from the conservation of mass and momentum equations. Generally, this involves the solution of an initial and boundary value problem, which in turn provides the averaged macroscopic quantities for example, density ρ and velocity v. Modeling of the traffic flow on a unidirectional single road by the scalar conservation laws was first done by Lighthill–Whitham and Richards (LWR) [68, 89].

$$\rho_t + (\rho v)_x = 0. \tag{1.1.1}$$

The model is valid if there are no entrances and no exits along the road. In order to obtain a closed system for ρ, an assumption is made that v is a function of ρ only i.e.,

$$v = v^e(\rho),$$

where v^e is a local equilibrium velocity. One can investigate this density-velocity relation in the following way e.g.,

$$v^e(\rho) = v_{max}\left(1 - \frac{\rho}{\rho_{max}}\right).$$

If the traffic is sufficiently light, then cars can move with a maximum speed v_{max}, although in certain limit defined by laws and technical limits. As the traffic increases moderately,

the average driving speed is less than the desired speed. Moreover, in the heavy traffic the average speed is reduced to zero. On the basis of this observation the above assumption can be validated as a reasonable one. Although this assumption is unrealistic and could not model certain real physical situations as criticized by Daganzo [27]. For example, in the real situation, flow generally does not follow the change of density. Moreover, this could not take into account multilane passing and cannot explain traffic instabilities due to formation of vehicle clusters. This leads to the development of non-equilibrium models that could overcome the flaws in the LWR model. The first model is the Payne–Whitham (PW) model including an additional equation which deals with the rate of change of velocity in a similar way of modeling of the momentum dynamics in fluid flow is introduced in [81],

$$\begin{aligned} \rho_t + (\rho v)_x &= 0, \\ v_t + v_x + \frac{p^{'}(\rho)}{\rho}\rho_x &= \frac{(v^e(\rho) - v)}{\tau}, \end{aligned} \tag{1.1.2}$$

where $p(\rho)$ is a pressure law and τ is the relaxation time of v to the equilibrium velocity $v^e(\rho)$. This model is capable of modeling traffic instabilities, like vehicle cluster formation and also the LWR stable traffic flow. The PW model also has followed the criticism, because the solution to this system with the Riemann data yields negative velocities $v < 0$. Moreover, it allows wave solutions with the characteristic speeds higher than the vehicular speeds. In order to surmount this difficulty Aw–Rascle have defined $p(\rho)$ heuristically in analogy to the gas dynamics by $p(\rho) = \rho^\gamma$, $\gamma > 0$ and derived the following model [3],

$$\begin{aligned} \rho_t + (\rho v)_x &= 0, \\ \Big(\rho\big(v + p(\rho)\big)\Big)_t + \Big(\rho v(v + p(\rho))\Big)_x &= \rho\frac{(v^e(\rho) - v)}{\tau}. \end{aligned} \tag{1.1.3}$$

Other extensions proposed are models taking into account the finite driver reaction time in the acceleration equation which have been subject of intense discussions [38, 52].

In this work the LWR model is considered for the study. However, this model is a first step towards mathematical modeling of the traffic flow. The velocity is a function of density is valid on roads for which road variables, the smoothness of road and the number of lanes are constant. The experimental support for the LWR model, was later being performed by Greenberg for the Lincoln tunnel in New York, see in [35].

In addition to modeling on single roads first extensions to Lighthill-Whitham [68] traffic flow models on road networks arose in [22], [43, 44] and [45]. The conditions governing the traffic flow through the junction in the network are introduced [22, 45]. Holden & Risebro [45] have studied traffic waves originating at highway junctions by assuming the existence of an optimization problem at each junction without route choice behavior. This model has limited applications in reality. Coclite, Garavello & Piccoli have included the route choice behavior in addition to the idea of solving an optimization problem at each

junction [22]. The detailed review of these junction conditions is being presented in the following sections.

Schematic overview of this chapter is as follows. The section 1.2, describes the underlying PDE traffic network model and the simplifications. Furthermore, the coupling conditions at different type of junctions are discussed. The new traffic flow model is introduced in section 1.3. The fluid-dynamics (PDE) traffic flow model is simplified and a new traffic flow model based on the ordinary differential equations (ODE) is derived. This is obtained by a spatial discretization of an averaged density evolution and a suitable approximation of the coupling conditions at junctions of the network. It will be shown that the new ODE model inherits similar features of the PDE model, e.g. traffic jam propagation. In Section 1.4 a review of the simplified algebraic model based on the system of coupled algebraic equations which has been proposed by Herty & Klar is presented [44]. The simplified algebraic model is further simplified and reformulated in section 1.5.

1.2 Macroscopic PDE Models

The main interest is in models for vehicular traffic flow based on the partial differential equations and their extension to network of roads. The macroscopic models describe average properties of vehicular traffic flow. Following the modeling concept from LWR model [68], the function $\rho(x,t)$ describes the car–density at a point x and a time t. Here, the word **car** loosely represents any vehicle. The evolution of density ρ is given by a scalar conservation law resembling the conservation of cars. Before discussing more modeling details, a network of roads is defined mathematically.

A **network of roads** can be modeled as a finite, directed and connected graph [22, 44]. Each edge of the graph models a single road and each vertex a junction in the network. Road networks are represented as directed graphs $G(V,E)$, where V is a set of vertices which correspond to junctions in the network and E is a set of edges which correspond to roads in the network. It is assumed that vertices in V have restricted geometry with a local degree at most 3. Only junctions of two topologies are possible. A junction is either with one incoming and two outgoing roads or with two incoming and one outgoing roads. Let cardinality of the set E is denoted by n_E. Further, each road $j = 1 \dots n_E$ is modelled by an interval $[a_j, b_j]$, where a_j or b_j can be infinity if and only if the road is incoming to or outgoing from the network, respectively. A sample network might have a structure as indicated in figure 1.2.1. This network also has only one inflowing road to the network and one outgoing road from the network.

1.2.1 Flow on Each Road

The density on each road j is modelled by a function $\rho_j(x,t)$. Further, the maximal allowable density on road j corresponding to the situation where cars stand bumper–to–

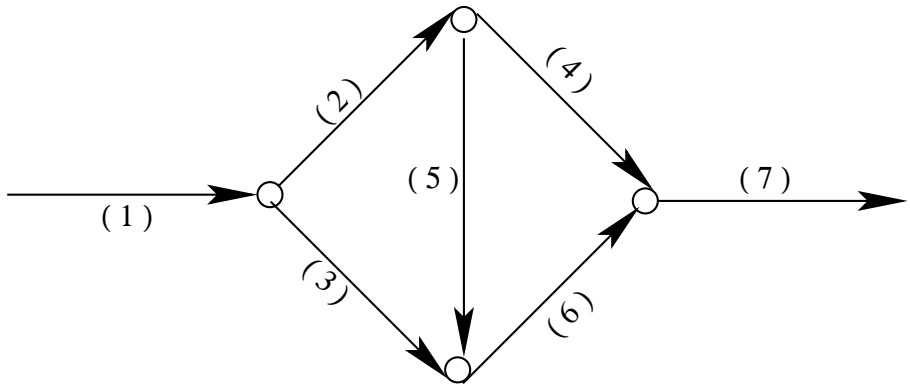

Figure 1.2.1: Geometry of a sample network.

bumper is denoted by $\rho_{\max,j}$. The number of cars crossing per unit time is called the traffic flow $f = \rho v$. In analogy to Lighthill & Whitham, May [68, 71] a relation is assumed (the so–called "fundamental diagram - flow-density curve") between three fundamental traffic variables, density $\rho_j(x,t)$, the average velocity of cars $v_j^e(x,t)$ and the flow $f_j(x,t)$ as,

$$f_j(x,t) = \rho_j(x,t)v_j(x,t) = \rho_j(x,t)v_j^e(x,t) \tag{1.2.4}$$

From now on the superscript e is skipped and as in the introduction it is assumed $v_j^e = v_j^e(\rho)$. Hence, f_j can be considered as a function of ρ_j only. Furthermore, one could study the situation where the flux function also depends on the position on the road, i.e., $f = f(x,\rho)$, which could be used to model bottleneck situations. Such fluxes on single roads are considered in chapter 5. The fundamental diagram changes in different situations, like change of road, time or weather. However, its shape shows some common features– for low densities, the flow increases quasi-linearly with the density, then it begins to grow slowly, goes through a maximum and then decays to zero at the maximal density. There has been intense discussion on the justification of the "fundamental diagram", but measurements on highways suggest that there exists a connection [62, 65] and that can be modelled by a concave flux function $f(\rho)$ with a single maximum. Therefore, we assume that there exists a family of flux–functions f_j such that for each road $j = 1, \ldots, n_E$

$$\left.\begin{array}{ll} (i) & f_j \text{ is continuously differentiable on } [0, \rho_{\max,j}], \\ (ii) & f_j(0) = f_j(\rho_{\max,j}) = 0, \\ (iii) & f_j \text{ is strictly concave }, \\ (iv) & \text{there exists } \sigma_j \in (0, \rho_{\max,j}) : f_j'(\sigma_j) = 0. \end{array}\right\} \tag{1.2.5}$$

It is observed from the above flux-function definition that there is no traffic flow at $\rho_j = 0$ and $\rho_j = \rho_{max,j}$. For other values of density $0 < \rho_j < \rho_{max,j}$ the traffic flow must be strictly positive. σ_j is the optimal density at which a maximum traffic flow occurs. Concavity of f means $d^2f/d\rho^2 < 0$, which implies that the $df/d\rho$ decreases as ρ increases. Greenberg has performed the validation of this fundamental diagram definition by the analysis of

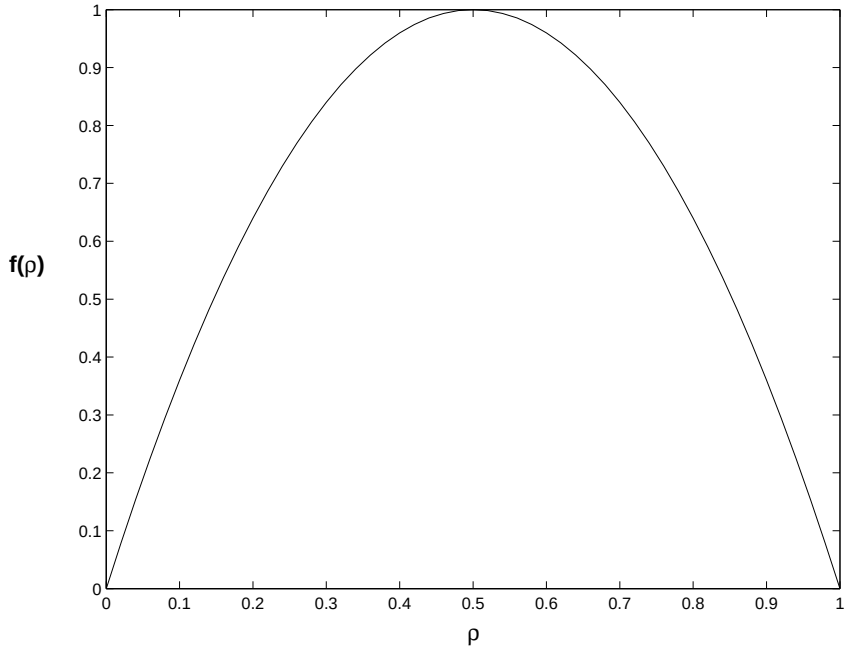

Figure 1.2.2: Fundamental diagram: flow-density curve .

experimental data [35]. Many other relations have been defined based on the empirical data [35, 45, 71]. Herein, a simple possible relation is considered which is given by,

$$v_j(\rho_j) = v_{max,j}\left(1 - \frac{\rho_j}{\rho_{max,j}}\right) \quad \Rightarrow \quad f_j(\rho_j) = \rho_j v_{max,j}\left(1 - \frac{\rho_j}{\rho_{max,j}}\right) \tag{1.2.6}$$

The fundamental diagram (figure 1.2.2) shows dependence of traffic flow on the density. Then, the macroscopic LWR model for traffic flow on a road j is given by the nonlinear conservation law,

$$\begin{aligned} \frac{\partial \rho_j(x,t)}{\partial t} + \frac{\partial f_j(\rho_j(x,t))}{\partial x} &= 0, \quad \forall x \in [a_j, b_j], \quad t \in [0,T],\ j \in E \\ \rho_j(x,0) &= \rho_{j,0}(x), \quad \forall x \in [a_j, b_j]. \end{aligned} \tag{1.2.7}$$

Equation (1.2.7) is hyperbolic under the assumption (1.2.5). This implies that it is able to describe the nonlinear density waves and discontinuities may arise even for smooth initial data.The density of cars is said to propagate as wave called density wave with a wave velocity. This velocity may be different from the velocity at which car moves. The wave velocity can be positive or negative depending upon the slope of $df/d\rho$. Also the initial data $\rho_j(x,0)$ in equation (1.2.7) can have one or more discontinuity. In case of a single discontinuity $\rho_j(x,0)$ at $c_j \in [a_j, b_j]$ is given as

$$\rho_j(x,0) = \begin{cases} \rho_l, & x < c_j, \\ \rho_r, & x \geq c_j \end{cases} \tag{1.2.8}$$

where ρ_l and ρ_r are constants on left and right hand side of the discontinuity at $x = 0$. A conservation law together with the piecewise constant data having a single discontinuity is known as the **Riemann Problem**. For example, equations (1.2.7) and (1.2.8) is a Riemann problem. The form of solution to the Riemann problem depends on the relation between ρ_l and ρ_r.

1.2.2 Flow through Junctions

In order to complete the model, one needs to define the flow through junctions in the network. Let us consider a single junction with n roads with incoming traffic to the junction labeled by $j = 1, \ldots, n$ with end b_j at the junction and m roads with outgoing traffic from the junction labeled by $j = n + 1, \ldots, n + m$ with end a_j at the junction as shown in figure 1.2.3.

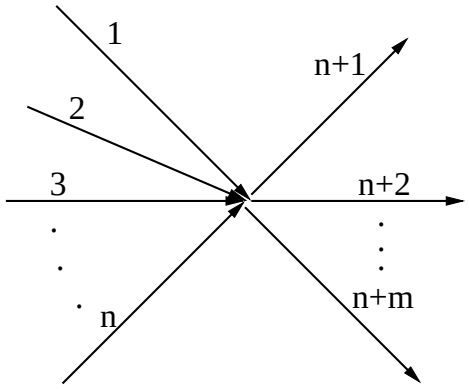

Figure 1.2.3: An example of a junction.

To guarantee the conservation of the number of cars at the junction the following condition is prescribed:

$$\sum_{j=1}^{n} f_j(\rho_j(b_j, t)) = \sum_{j=n+1}^{n+m} f_j(\rho_j(a_j, t)), \quad \forall\, t \geq 0. \tag{1.2.9}$$

This is the Rankine-Hugoniot condition at the junction. Equations (1.2.7) are coupled by the boundary conditions at junctions. However, the condition (1.2.9) does not suffice to determine a unique solution of the system of coupled PDE's to model the flow in the network. One needs to define additional conditions at the junction since we have $n + m$ unknowns and only one equation at the junction. Let

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

denotes the solution at the junction. If $\bar{\rho}$ is known, then a Riemann problem is solved for each road with $\bar{\rho}_j$ as the right state for incoming roads ($j \leq n$) and the left state for outgoing roads with ($j \geq n + 1$). The solution may consist of shock waves or rarefaction

waves emerging from the junction.

To define these additional conditions, there are two approaches proposed in [22, 45]. Holden & Risebro have denoted the additional condition by an entropy condition in correspondence with that of the conservation laws, which provides unique solution in later case in [45] . This entropy condition is derived by maximizing the flux locally at each junction subject to equation (1.2.9). The flux is measured by a concave differentiable flux. But, the approach of Coclite, Garavello & Piccoli [22] is followed in this work and briefly discussed here. Introduce a matrix $\mathcal{A} \in R^{m\times n}$, in [22] where

$$(\mathcal{A})_{ji} = \alpha_{ji} \quad j \in \{n+1, ..., n+m\}, \quad i \in \{1, ..., n\},$$

describes percentage of drivers who want to drive (and also must drive) from road i to road j . The matrix A is assumed to fulfill the following assumptions,

$$\alpha_{ji} \neq \alpha_{ji'}, \forall i \neq i' \text{ and } 0 < \alpha_{ji} < 1 \text{ and } \sum_{j=n+1}^{n+m} \alpha_{ji} = 1, \quad \forall i \in \{1, ..., n\}. \tag{1.2.10}$$

The weak formulation of the system of conservation laws will provide the right boundary conditions: A weak solution at a junction is a collection of functions $\rho_j : [0, \infty) \times [a_j, b_j] \to \mathbb{R}$ for $j = 1, \ldots, n+m$ s.t.

$$\sum_{j=1}^{n+m} \int_0^\infty \int_{a_j}^{b_j} \Big(\rho_j \partial_t \phi_j + f(\rho_j)\partial_x \phi_j\Big) dx dt = 0 \tag{1.2.11}$$

for each $\phi_j, j = 1, \ldots, n+m$ is a smooth function and having compact support in $\mathbb{R}\times(0, \infty)$ and is smooth across the junction, i.e., for $i = 1, \ldots, n$, $j = n+1, \ldots, n+m$,

$$\phi_i(b_i, \cdot) = \phi_j(a_j, \cdot), \;\; \partial_x \phi_i(b_i, \cdot) = \partial_x \phi_j(a_j, \cdot).$$

It is to note that (1.2.11) implies (1.2.9) if functions ρ_j are sufficiently regular. Now conditions are added which have to be satisfied to obtain a **unique** weak admissible solution at the junction. Let us assume that $\rho_j(t, \cdot)$ are functions of bounded variation and the following additional properties

$$f(\rho_j(a_j+, \cdot)) = \sum_{i=1}^{n} \alpha_{ji} f(\rho_i(b_i-, \cdot)) \quad j = n+1, \ldots, n+m, \tag{1.2.12}$$

$$\sum_{i=1}^{n} f(\rho_i(b_i-, \cdot)) + \sum_{i=n+1}^{n+m} f(\rho_i(a_i+, \cdot)) \text{ is maximal w.r.t. (1.2.12)}, \tag{1.2.13}$$

are satisfied. These two expressions represent the distribution of the traffic flow on outgoing roads according to constants α_{ji} at the same time maximizing the flux at the junction. The following result is concerning existence and uniqueness of the soultion is known from [22].

Theorem 1.1 (Theorem. 5.1 of [22]) *Consider a flux function f_j satisfying equation (1.2.7) and a road network in which all junctions have at most two ingoing and two outgoing roads. Let $\bar{\rho} = (\rho_1, \ldots, \rho_I)$ be an initial data in L^1_{loc} and $T > 0$ is fixed. Then there exists a unique admissible solution $\rho = (\rho_1, \ldots, \rho_I)$, $\rho_j : [a_j, b_j] \times [0, T] \to \mathbb{R}$ with $\rho(\cdot, 0) = \bar{\rho}$.* □

The main step in the proof of existence and uniqueness of admissible weak solutions is the consideration of constant initial data $\rho_{j,0}$ and a single junction only. An admissible solution can be constructed as follows. For each road j an intermediate state $\overline{\rho}_j \in \mathbb{R}$, $j = 1, \ldots, n + m$ is introduced. The solution $\rho_j(x, t)$ to the problem (1.2.7) and (1.2.9) is given as a solution to the Riemann problem on each road j. For incoming roads the initial conditions for the Riemann problem are

$$\rho_j(x, 0) = \begin{cases} \rho_{j,0}, & x \leq b_j, \\ \bar{\rho}^b_j, & x > b_j, \end{cases} \tag{1.2.14}$$

and similarly for outgoing roads with the state $\overline{\rho}^a_j$,

$$\rho_j(x, 0) = \begin{cases} \bar{\rho}^a_j, & x < a_j, \\ \rho_{j,0}, & x \geq a_j. \end{cases} \tag{1.2.15}$$

Hereby, certain restrictions on values $\bar{\rho}^a_j$, $\bar{\rho}^b_j$ are imposed. It is assumed that $\bar{\rho}^a_j$, $\bar{\rho}^b_j$ are independent of time and all waves in the solution of Riemann problems have to emerge from the junction, i.e., have non–positive speed for ingoing and non–negative speed for outgoing roads.

One can note that the existence theorem 1.1 only covers junctions with total of atmost four connected roads. However, the construction of states $\bar{\rho}^a_j$, $\bar{\rho}^b_j$ is not explicit in a case with more than three connected roads. Therefore, the discussion is restricted to the cases of three connected roads and the precise formulas for the intermediate states $\bar{\rho}^a_j$, $\bar{\rho}^b_j$ are given. By composition of such junctions one can easily model all other kinds of possible junctions. The derivation of exact formulation of $\bar{\rho}^a_j$, $\bar{\rho}^b_j$ is deferred to the section 1.3.1.

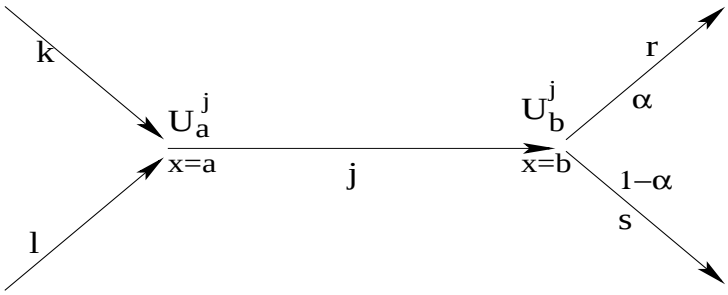

Figure 1.2.4: Coupling condition at the junction.

Here, the following notations are introduced which are used in order to define intermediate states. On fixing a road j and considering the situation depicted in figure 1.2.4, i.e., at

$x = a_j$ a junction connects roads j, k, l and at $x = b_j$ another junction connects roads j, r, s. For given values ρ_j, ρ_k, ρ_l and ρ_j, ρ_r, ρ_s define functions U_a^j and U_b^j respectively, by

$$\begin{aligned} U_a^j(\rho_j, \rho_k, \rho_l) &= \overline{\rho}_j^a, \\ U_b^j(\rho_j, \rho_r, \rho_s, \alpha) &= \overline{\rho}_j^b, \end{aligned} \tag{1.2.16}$$

where $0 \leq \alpha \leq 1$ is fixed and where $\overline{\rho}_j^a$ and $\overline{\rho}_j^b$ are given by the discussion in section 1.3.1.

Now, the traffic model is rephrased as a coupled system of partial differential equations with explicit boundary values. Finally, the PDE model on a single road j which is connected on both sides to two junctions as in figure 1.2.4 reads

$$\left.\begin{aligned} &\Big(\rho_j(x,t)\Big)_t + \Big(f(\rho_j(x,t))\Big)_x = 0, && \forall x \in [a_j, b_j],\ t > 0, \\ &\rho_j(x,0) = \rho_{j,0}(x), && \forall x \in [a_j, b_j], \\ &\overline{\rho}_j^a = \rho_j(a,t) = U_a^j\Big(\rho_j(a,t-), \rho_k(b,t-), \rho_l(b,t-)\Big), && \forall t > 0, \\ &\overline{\rho}_j^b = \rho_j(b,t) = U_b^j\Big(\rho_j(b,t-), \rho_r(a,t-), \rho_s(a,t-), \alpha\Big), && \forall t > 0. \end{aligned}\right\} \tag{1.2.17}$$

The solution of the PDE traffic network model is time consuming and cannot be done in real–time even with the appropriate schemes. Computation times are reported later in chapter 4, which supports this fact. The situation is more severe in the case of solving optimal control problems governed by the PDE models, since in each optimization step several simulations of the governing equations are needed. Therefore, a simplified model obtained by spatial discretization of the PDE model is introduced in the next section and the corresponding coupling conditions at junctions. To be more precise, based on an averaged density evolution of the traffic flow on each road, a simple finite spatial discretization of equation (1.2.7) is performed and an ODE model is deduced.

1.3 Macroscopic ODE Model

The goal is to derive a comparatively simpler model by a suitable approximation of the PDE model. The simplified model should have similar qualitative behavior, but should be computationally cheaper than the PDE model. The starting point is the LWR model,

$$\begin{aligned} \frac{\partial \rho_j(x,t)}{\partial t} + \frac{\partial f_j(\rho_j(x,t))}{\partial x} &= 0, \qquad \forall x \in [a_j, b_j], \quad t \in [0,T],\ j \in E \\ \rho_j(x,0) &= \rho_{j,0}(x), \qquad \forall x \in [a_j, b_j]. \end{aligned} \tag{1.3.18}$$

An approximation to this model is derived by a three-point spatial discretization of each road and suitable coupling conditions. The final model consists of a system of coupled ordinary differential equations.

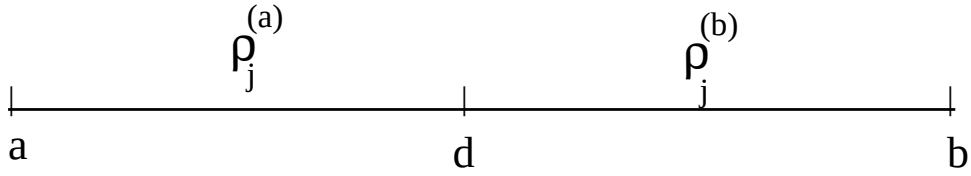

Figure 1.3.5: Spatial discretization on each road j with density approimations as $\rho_j^{(a)}$ and $\rho_j^{(b)}$.

For notational simplicity here the subscripts for a_j, b_j and $L_j = b_j - a_j$ are dropped. On integrating (1.3.18) over $[a, d]$ and $[d, b]$, $a < d = \frac{a+b}{2} < b$ one obtains,

$$\partial_t \rho_j^{(a)}(t) = -\frac{2}{L}\Big(f(\rho_j(d,t)) - f(\rho_j(a,t))\Big), \tag{1.3.19}$$

$$\partial_t \rho_j^{(b)}(t) = \frac{2}{L}\Big(f(\rho_j(d,t)) - f(\rho_j(b,t))\Big), \tag{1.3.20}$$

where L is the length of the road and the spatial approximations $\rho_j^{(a)}$, $\rho_j^{(b)}$ are

$$\rho_j^{(a)}(t) = \frac{2}{L}\int_a^d \rho_j(x,t)dx \quad \text{and} \quad \rho_j^{(b)}(t) = \frac{2}{L}\int_d^b \rho_j(x,t)dx. \tag{1.3.21}$$

Equations (1.3.19), (1.3.20) are not closed and consist of additional unknowns $\rho_j(d,t)$. For $\rho_j(d,t)$ it is assumed that the half–sum is a reasonable approximation,

$$\rho_j(d,t) = \frac{1}{2}(\rho_j^{(a)}(t) + \rho_j^{(b)}(t)). \tag{1.3.22}$$

Initial conditions are obtained by averaging

$$\rho_{j,0}^{(a)} = \frac{2}{L}\int_a^d \rho_{j,0}(x)dx \quad \text{and} \quad \rho_{j,0}^{(b)} = \frac{2}{L}\int_d^b \rho_{j,0}(x)dx \tag{1.3.23}$$

Finally, the values $\rho_j(a,t)$ and $\rho_j(b,t)$ are obtained by the coupling conditions discussed in the next section and defined by,

$$\bar{\rho}_j^a(t) = U_a^j\Big(\rho_j^{(a)}(t), \rho_k^{(a/b)}(t), \rho_l^{(a/b)}(t)\Big), \tag{1.3.24}$$

$$\bar{\rho}_j^b(t) = U_b^j\Big(\rho_j^{(b)}(t), \rho_r^{(a/b)}(t), \rho_s^{(a/b)}(t), \alpha\Big). \tag{1.3.25}$$

where a and b are chosen for the incoming and the outgoing roads at the junction respectively. For the above formulas, let us assume a situation as in figure 1.2.4. Hence equations (1.3.25), (1.3.24), (1.3.23), (1.3.22), (1.3.20) and (1.3.19) define a closed system of coupled ordinary differential equations. In order to solve equations (1.3.19)-(1.3.25) are discretized using a fixed step–width τ. The discretized equations for the ODE model for the road j are given by the following system of coupled equations,

$$\left.\begin{array}{rcl} \rho_j^{(a)}(t+\tau) &=& \rho_j^{(a)}(t) - \frac{2\tau}{L}\left(f(\frac{\rho_j^{(a)}(t)+\rho_j^{(b)}(t)}{2}) - f(\overline{\rho}_j^a(t))\right), \\ \rho_j^{(b)}(t+\tau) &=& \rho_j^{(b)}(t) + \frac{2\tau}{L}\left(f(\frac{\rho_j^{(a)}(t)+\rho_j^{(b)}(t)}{2}) - f(\overline{\rho}_j^b(t))\right), \\ \overline{\rho}_j^a(t) &=& U_a^j\Big(\rho_j^{(a)}(t), \rho_k^{(a/b)}(t), \rho_l^{(a/b)}(t)\Big), \\ \overline{\rho}_j^b(t) &=& U_b^j\Big(\rho_j^{(b)}(t), \rho_r^{(a/b)}(t), \rho_s^{(a/b)}(t), \alpha\Big), \end{array}\right\} \tag{1.3.26}$$

where τ should satisfy the CFL condition (1.3.27) [67], since the above discretization is a finite difference scheme for a conservation law, i.e., it is required that τ should satisfy the following condition:

$$\tau \leq \frac{L}{2\,\max_\rho f'(\rho)} \tag{1.3.27}$$

In the numerical results it can be seen that the naive discretization by an explicit Euler–scheme as in (1.3.26) does not produce comparable results to a Godunov–discretization of (1.3.18). Indeed, it is well–known that this scheme is oscillating. Therefore, a Lax–Friedrich [67] discretization of the time derivative has been proposed. Finally, the following system (ODE model) is obtained for a road j connected to two junctions as shown in figure 1.2.4 and τ as in (1.3.27).

$$\left.\begin{array}{rcl} \rho_j^{(a)}(t+\tau) &=& \left(\frac{\overline{\rho}_j^a(t)+\rho_j^{(b)}(t)}{2}\right) - \frac{2\tau}{L}\left(f(\rho_j^{(b)}(t)) - f(\overline{\rho}_j^a(t))\right), \\ \rho_j^{(b)}(t+\tau) &=& \left(\frac{\rho_j^{(a)}(t)+\overline{\rho}_j^b(t)}{2}\right) + \frac{2\tau}{L}\left(f(\rho_j^{(a)}(t)) - f(\overline{\rho}_j^b(t))\right), \\ \overline{\rho}_j^a(t) &=& U_a^j(\rho_j^{(a)}(t), \rho_k^{(a/b)}(t), \rho_l^{(a/b)}(t)), \\ \overline{\rho}_j^b(t) &=& U_b^j(\rho_j^{(b)}(t), \rho_r^{(a/b)}(t), \rho_s^{(a/b)}(t), \alpha). \end{array}\right\} \tag{1.3.28}$$

Remark 1.1 *Note that (1.3.28) is different from any discretization of the partial differential equation (1.2.7) due to the approximation of $\rho_j(d,t)$ in (1.3.22) and due to the definition of the boundary values $\bar{\rho}_j^a(t)$, $\bar{\rho}_j^b(t)$.*

1.3.1 Coupling conditions

In this section values for intermediate density $\bar{\rho}^a$, $\bar{\rho}^b$ are derived and assigned to solve the Riemann problem at the junction. As mentioned in section 1.2.2 junctions with degree 3 are only considered. There are two possibilities of junctions with a total of three connected roads: either one road disperses into two roads or two roads merge into one road.

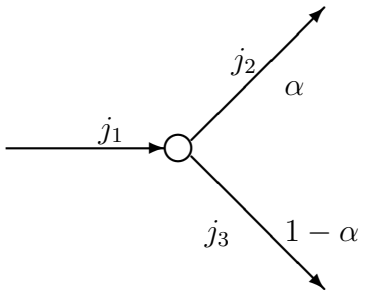

Figure 1.3.6: Dispersing type of junction.

Dispersing junction

In the case of dispersing junction, the incoming road is labeled as j_1 and two outgoing roads are labeled as j_2, j_3 as shown in figure 1.3.6.

According to Coclite, Garavello & Piccoli [22] the matrix $\mathcal{A}$ is $\mathcal{A} = [\alpha, 1-\alpha]$ where $0 \leq \alpha \leq 1$. Each strictly concave flux function f_j has a unique, single maxima denoted by σ_j and f_j is invertible on $[0, \sigma_j]$ and $[\sigma_j, 1]$. The inverse is denoted by $f_j^{-1,+}$ and $f_j^{-1,-}$ on $[0, \sigma_j]$ and $[\sigma_j, 1]$, respectively.

$\rho_{j,0}$ is assumed as a constant and constant fluxes c_j are introduced which are given by,

$$c_{j_1} = \begin{cases} f_{j_1}(\rho^{(b)}_{j_1,0}), & \rho^{(b)}_{j_1,0} < \sigma_{j_1} \\ f_{j_1}(\sigma_{j_1}), & \rho^{(b)}_{j_1,0} > \sigma_{j_1} \end{cases}$$

$$c_k = \begin{cases} f_k(\rho^{(a)}_{k,0}), & \rho^{(a)}_{k,0} > \sigma_k \\ f_k(\sigma_k), & \rho^{(a)}_{k,0} < \sigma_k, \end{cases} \qquad k = j_2, j_3$$

$$\gamma = \min\{c_{j_1}, \frac{c_{j_2}}{\alpha}, \frac{c_{j_3}}{1-\alpha}\}. \tag{1.3.29}$$

Then the following closed formulas for $\overline{\rho}^a_j$, $\overline{\rho}^b_j$ are obtained from Coclite, Garavello & Piccoli [22]

$$\overline{\rho}^b_{j_1} = \begin{cases} \rho^{(b)}_{j_1,0}, & \rho^{(b)}_{j_1,0} < \sigma_{j_1}, \gamma = c_{j_1} \\ f^{-1,+}_{j_1}(\gamma), & \text{else} \end{cases} \tag{1.3.30}$$

$$\overline{\rho}^a_{j_2} = \begin{cases} \rho^{(a)}_{j_2,0}, & \rho^{(a)}_{j_2,0} > \sigma_{j_2}, \gamma = c_{j_2}/\alpha \\ f^{-1,-}_{j_2}(\alpha\gamma), & \text{else} \end{cases} \tag{1.3.31}$$

$$\overline{\rho}^a_{j_3} = \begin{cases} \rho^{(a)}_{j_3,0}, & \rho^{(a)}_{j_3,0} > \sigma_{j_3}, \gamma = c_{j_3}/(1-\alpha) \\ f^{-1,-}_{j_3}((1-\alpha)\gamma), & \text{else} \end{cases} \tag{1.3.32}$$

As discussed in more detail in [22, 42, 44], the above conditions guarantee by construction that, (1.2.9) is satisfied, all waves have the "correct" wave speeds and the flux on incoming roads is maximal subject to the conditions.

Merging junction

In case of merging junction, incoming roads to the junction are labeled as j_1, j_2 and the outgoing road as j_3 as drawn in figure 1.3.7.

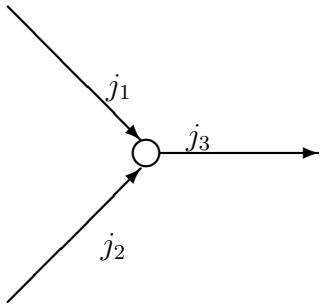

Figure 1.3.7: Merging type of junction.

The initial densities on roads j are given by $\rho_{j,0}$ for $j \in \{j_1, j_2, j_3\}$ and are assumed to be constant. Let $\gamma_{j,0} = f(\rho_{j,0})$ and let c_j be,

$$c_k = \begin{cases} f_k(\rho_{k,0}^{(b)}), & \rho_{k,0}^{(b)} < \sigma_k \\ f_k(\sigma_k), & \rho_{k,0}^{(b)} > \sigma_k, \end{cases} \quad k = j_1, j_2$$

$$c_{j_3} = \begin{cases} f_{j_3}(\rho_{j_3,0}^{(a)}), & \rho_{j_3,0}^{(a)} > \sigma_{j_3} \\ f_{j_3}(\sigma_{j_3}), & \rho_{j_3,0}^{(a)} < \sigma_{j_3} \end{cases}$$

Combining results of [44] and [22] and distinguish, if

$$c_{j_1} + c_{j_2} \leq c_{j_3},$$

then

$$\gamma_{j_1} = c_{j_1}, \quad \gamma_{j_2} = c_{j_2} \quad \text{and} \quad \gamma_{j_3} = \gamma_{j_1} + \gamma_{j_2} \tag{1.3.33}$$

and if $\quad c_{j_1} + c_{j_2} > c_{j_3}$, then

$$\gamma_{j_1} = \gamma_{j_2} = \min\{c_{j_1}, c_{j_2}, c_{j_3}/2\} \tag{1.3.34}$$

$$\gamma_{j_3} = \gamma_{j_1} + \gamma_{j_2} \tag{1.3.35}$$

Note that the coupling in the second case (1.3.35) resembles a first–in–first–out principle. In that particular situation the inflow $\gamma_{j_1} + \gamma_{j_2}$ is larger than the maximal possible outflow. Therefore, it is required that the cars enter in an alternating way into the outgoing road j_3. This explains the factor $1/2$ appearing in the calculation of the actual flow at the junction.

Finally, states $\bar{\rho}_j^a$, $\bar{\rho}_j^b$ are defined by

$$\overline{\rho}_k^b = \begin{cases} \rho_k^{(b)}, & \rho_k^{(b)} < \sigma_k, \gamma_k = c_k \\ f_k^{-1,+}(\gamma_k), & \text{else,} \end{cases} \qquad k = j_1, j_2 \tag{1.3.36}$$

$$\overline{\rho}_{j_3}^a = \begin{cases} \rho_{j_3}^{(a)}, & \rho_{j_3}^{(a)} > \sigma_{j_3}, \gamma_{j_3} = c_{j_3}/2 \\ f_{j_3}^{-1,-}(\gamma_{j_3}), & \text{else} \end{cases} \tag{1.3.37}$$

From the above discussion $\bar{\rho}_j^a$, $\bar{\rho}_j^b$ are determined uniquely and a solution $\rho_j(x,t)$ for constant initial data can be obtained by solving the Riemann problems. The solution to the problem with non–constant initial data can be obtained by wave or front–tracking [12] methods.

Remark 1.2 *According to [22] the Theorem 1.1 is true in case of $A = A(t)$, i.e., time–dependent distributions of the flux at junctions. This simplified model also remain valid even in this case. For simplicity, the case of time–independent controls are dealt with. Moreover, the coupling conditions allow shock waves (corresponding to the traffic jams) to pass through a junction. This implies that a crowed outgoing road may generate a traffic jam on an incoming road.*

The ODE model (1.3.28) uses the same function $U_{a,b}^j(\cdot)$ as of the PDE model as in [42]. Hence, the ODE model also inherits the property of traffic jams moving backwards through the junction.

1.4 Simplified Algebraic Model

The simplified algebraic model for the traffic flow on networks has been derived from the PDE model by Herty & Klar. A brief review of the model is presented in this section [44].

The solution of optimal control problems defined for the large scale networks based on the PDE model leads to a high computational costs. Hence, Herty & Klar [44] have proposed a simplified dynamics of traffic flow and have performed comparison of simulation and numerical results with the original model. Further equations modeling the flow are derived using only (1.2.9).

$\rho_{j,0}$ is assumed as an approximation of the density ρ_j and t_j as the arrival time of a wave defining evolution of the traffic flow on each road j. Assume that ρ_0 is a given inflow to the network on the ingoing road and $\rho_{j,0} = 0 \quad \forall j \in E$, which means that the network is empty initially. The time interval considered for simulating the flow on the network is $[0, T]$. Now, the traffic flow model is converted to a system of algebraic equations based on the following assumptions. It has been assumed that no backward moving shock waves are allowed, which means no traffic jam situation occurs. This assumption leads to the following bounds,

$$0 \leq \rho_{j,0} \leq \sigma_j \quad \text{where} \quad \sigma_j = \text{argmax } f_j(\rho_j) \quad \text{and} \quad 0 \leq t_j \leq T \quad \forall j \in E \tag{1.4.38}$$

$\rho_{j,0}$ can be determined solely by the coupling conditions at the junction subject to equation (1.4.38). Under the assumption (1.4.38) Coclite, Garavello & Piccoli coupling condition is reduced as follows. For the ingoing road to the network,

$$\rho_{1,0} = \rho_0. \tag{1.4.39}$$

In case of a dispersing junction with an incoming road k, outgoing roads l, m one obtains,

$$\rho_{l,0} = f_l^{-1}(\alpha f_k(\rho_{k,0})), \quad \rho_{m,0} = f_m^{-1}((1-\alpha) f_k(\rho_{k,0})), \tag{1.4.40}$$

where suitable control $0 < \alpha < 1$ is applied at the dispersing junction steering the flux distribution. These relations are well defined with respect to the flux function defined in the section 1.2 by the fundamental diagram. Similarly for a merging junction with incoming roads k, l and an outgoing road m the following expression is obtained,

$$\rho_{m,0} = f_m^{-1}(f_k(\rho_{k,0}) + f_l(\rho_{l,0})). \tag{1.4.41}$$

In order to model dynamics, $t_{j,0}$ is determined by tracking a single shock front on road j. For the ingoing road to the network,

$$t_{1,0} = 0. \tag{1.4.42}$$

In case of a dispersing junction with road j splitting into two roads k, l, the arrival time of the wave at the start of roads k, l is given by,

$$t_l = t_k = t_{j,0} + \frac{b_j - a_j}{s_j}, \tag{1.4.43}$$

where $s_j = f_j(\rho_{j,0})/\rho_{j,0}$ is a Rankine-Hugeniot wave speed. In case of merging junction when two roads k, l are merging into a road j,

$$t_j = \left(t_{k,0} + \frac{b_k - a_k}{s_k}\right)\frac{\rho_k}{\rho_k + \rho_l} + \left(t_{l,0} + \frac{b_l - a_l}{s_l}\right)\frac{\rho_l}{\rho_k + \rho_l}. \tag{1.4.44}$$

This expression is derived based on the assumption that there is only one shock wave and no rarefaction waves. For more details do refer to [44]. Equations (1.4.39)-(1.4.44) define a system of algebraic equations. Due to nonlinearities in expressions the evaluation of $\rho_{j,0}$ and $t_{j,0}$ results in large computation time for large network simulations even. To overcome this the coupling conditions are linearized in the subsequent section.

1.5 Reformulated Simplified Algebraic Model (RSA Model)

Let us neglect the dynamics in the simplified algebraic model[44]

$$t_{j,0} = 0, \quad \forall j \in E$$

and define the model in terms of flux q_j only. Every road in the network has a maximum capacity, i.e., it allows a maximal flow. This means a maximal flux to each road can be assigned as $q_{c_j} > 0$ and assign a flux variable q_j to each road satisfying the following constraint,

$$0 \leq q_j \leq q_{c_j}. \tag{1.5.45}$$

The maximal flux is assumed to be known. Further is is assumed that there is a constant inflow q_0 as boundary conditions on all incoming roads to the network,

$$q_1 = q_0 \tag{1.5.46}$$

At junctions the fluxes of the different roads are coupled such that the mass is conserved. The coupling conditions (1.4.40)-(1.4.41) at junctions in the form of fluxes yield a system of linear equations. For a junction with three connected roads labelled by k, l and j yields,

$$q_k + q_l = q_j \tag{1.5.47}$$

Let $n_V = |V| + 1$ and $n_E = |E|$ and coupling conditions (1.5.46), (1.5.47) are rewritten as the linear equality constraints,

$$h(q) = H^T q + h_0 = 0 \tag{1.5.48}$$

where $q = (q_j)_j \in \mathbb{R}^{n_E}$, $h \in \mathbb{R}^{n_V}$, $H \in \mathbb{R}^{n_E \times n_V}$ and $h_0 \in \mathbb{R}^{n_V}$ is the vector of inflows to the network. For example, corresponding to the sample network in figure 1.2.1, H^T has the following structure,

$$H^T = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & -1 & 0 & -1 & 1 \end{pmatrix}.$$

and

$$h_0^T = \begin{pmatrix} q_0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

1.6 Summary

The modeling aspect of the traffic flow in networks and the existing models based on PDE's and algebraic equations are discussed and briefly reviewed. An ODE model for the network traffic flow is deduced from the PDE model and boundary conditions at the junction are defined. Junctions with three roads are considered only with two incoming and one outgoing or with one incoming and two outgoing. In order to optimize the traffic flow in the network, the necessary cost functionals for the above discussed models and the respective optimization problems is defined in chapter 2.

Chapter 2

Cost Functionals and Gradient Evaluation

In this chapter optimization problems for road networks governed by models presented in the chapter 1 are formulated. These problems are solved using gradient based methods. Gradient of each cost functional of ODE is evaluated by an adjoint calculus. At the end the optimality conditions and bound constrained optimization algorithms are discussed.

2.1 Optimal Control Problem for the ODE Model

In the following a network is assumed to have only one incoming and one outgoing road. As mentioned in the chapter 1 the network consists of junctions with only three roads where either two roads merge into one road or two roads disperse into one road. An optimal control problem for the ODE model introduced in section 1.3 is defined below.

2.1.1 Cost Functional

It is assumed in the following that the traffic can be distributed at certain dispersing junctions of the network and hence can be controlled at that particular dispersing junction. Let us consider the network has l dispersing junctions. The constant $\alpha_i,\ i \in V$ defined by (1.2.10) and $0 \leq \alpha_i \leq 1,\ i = 1, \ldots, l$ is a flux steering factor at the dispersing junction [22]. In case of a dispersing junction with only three incident roads, the incoming flux is distributed as α_i times the incoming flux to one outgoing road and $(1-\alpha_i)$ times incoming flux on the other outgoing roads. This will be the control parameter while optimizing the flow in the network.

In practical applications the value of α_i is a recommendation factor that can be taken as a suggestion in the car–navigation systems or as in signs at the corresponding highway intersections. For simplicity it is assumed that the traffic is actually distributed according to the value of α_i. Although, there are situations where not all cars follow the recommendation and a more sophisticated model would have to be taken into account for the random behavior of cars at the junction. But, the discussion below is interesting for

traffic management and investigations of "optimal" utilization of a given network.

Suppose the inflow density profile at time $t = 0$ is ρ_0 and a time horizon $T > 0$ is given. The control parameter $\alpha_i \in [0, 1]$ which are applied at each dispersing junction $i = 1, \ldots, l$ of the network appear in functions U_b^j defined by (1.3.25).

The goal is to find a fastest way through the network with respect to the traffic and road conditions. Based on basic assumption of the LWR model and using linear density-velocity relation (1.2.6), this implies a low density on each road. A well known measure for the better utilization of a single road j of the network is the time and space averaged density given by the following

$$\int_0^T \int_{a_j}^{b_j} \rho_j(x,t)dxdt.$$

where ρ_j is the density approximation on each road j. Hence, summing up for all roads in the network a cost functional $\mathcal{J}$ can be defined as

$$\mathcal{J}(\alpha; T, \rho_0) = \sum_{j=1}^{n_E} \int_0^T \int_a^b \rho_j(x,t)dxdt. \tag{2.1.1}$$

The function $\mathcal{J}(\alpha; T, \rho_0)$ measures the averaged time and space densities in the whole network. The fastest way through the network can be obtained by minimizing $\mathcal{J}(\alpha; T, \rho_0)$, see remark 2.1. The equation (2.1.1) defines the cost functional for the PDE model on the road networks. Here interest is in controls α such that the functional $\mathcal{J}(\alpha)$ is minimized and the precise optimization problem corresponding to the PDE model for the road network is given as,

$$\min_{\alpha} \mathcal{J}(\alpha; T, \rho_0) \text{ subject to } 0 \leq \alpha \leq 1 \text{ and (1.2.17)} \tag{2.1.2}$$

where inequality is true componentwise, i.e., for each $\alpha_i,\ i = 1 \cdots l$.

Remark 2.1 *The functional $\mathcal{J}(\alpha; T, \rho_0)$ is popular in the traffic engineering community [65]. Since the flux functions are concave, high densities are related to small velocities v_j, i.e., $\rho_j v_j = f_j(\rho_j)$. Therefore, minimizing (2.1.1) yields a traffic situation with a large average speed. Similarly, the functional $\mathcal{J}$ penalizes backward moving waves in the network. These waves can be interpreted as traffic jams.*

It is easy to verify that in case of a single inflow arc j_1, outflow arc j_{n_E} and sufficiently regular solutions ρ_j,

$$\mathcal{J}_0(\alpha; T, \rho_0) = \int_0^T f_{j_1}(\rho(a_{j_1}, t))dt - \int_0^T f_{j_{n_E}}(\rho(b_{j_{n_E}}, t))dt. \tag{2.1.3}$$

Remark 2.2 *According to (2.1.3), the functional $\mathcal{J}_0(\alpha; T, \rho_0)$ measures the possible flow passing the network depending on routing decisions at the junction. This is also a frequently considered cost functional for optimizing the traffic flow in a network.*

The cost functional for the ODE model is obtained from an approximation of $\mathcal{J}(\alpha; T, \rho_0)$. Hence, on discretizing (2.1.1) and using space averaged approximation for density in the ODE model, the discretized form of cost functional $\mathcal{J}_t$ is

$$\mathcal{J}_t(\alpha; T, \rho_0) = \sum_{t=1}^{T} \sum_{j=1}^{n_E} \frac{L_j}{2} \tau \left(\rho_j^{(a)}(t) + \rho_j^{(b)}(t) \right) \tag{2.1.4}$$

where $L_j = b_j - a_j$, τ satisfies (1.3.27) and the definition of $\rho_j^{(a)}$ and $\rho_j^{(b)}$ is given by (1.3.21). The minimization problem corresponding to the ODE model for networks is

$$\left.\begin{array}{l} \min_\alpha \mathcal{J}_t(\alpha; T, \rho_0) \\ \text{ subject to } \quad 0 \le \alpha_i \le 1, \quad i = 1, \ldots, l, \\ \text{and state equations, } \quad \forall j \in E \\ \rho_j^{(a)}(t+\tau) = \left(\frac{\overline{\rho}_j^a(t) + \rho_j^{(b)}(t)}{2} \right) - \frac{2\tau}{L} \Big(f(\rho_j^{(b)}(t)) - f(\overline{\rho}_j^a(t)) \Big), \\ \rho_j^{(b)}(t+\tau) = \left(\frac{\rho_j^{(a)}(t) + \overline{\rho}_j^b(t)}{2} \right) + \frac{2\tau}{L} \Big(f(\rho_j^{(a)}(t)) - f(\overline{\rho}_j^b(t)) \Big), \\ \overline{\rho}_j^a(t) = U_a^j(\rho_j^{(a)}(t), \rho_k^{(a/b)}(t), \rho_l^{(a/b)}(t)), \\ \overline{\rho}_j^b(t) = U_b^j(\rho_j^{(b)}(t), \rho_r^{(a/b)}(t), \rho_s^{(a/b)}(t), \alpha). \end{array}\right\} \tag{2.1.5}$$

A solution to this problem determines an optimal distribution of a traffic flow in a network in presence of jam situations too. The traffic distribution in the sample networks is optimized using different optimization methods in chapter 4.

2.2 Adjoint and Gradient Equations

In section 2.1 the optimal control problem for the ODE model is introduced. In this section gradient of cost functional of the ODE model defined by (2.1.4) is evaluated with respect to the control parameter α. This gradient is used later to compute the numerical solution of (2.1.5).

In some optimal control problems the cost functional is an implicit function of control variables. Then the computation of the gradient of the cost functional is not straightforward. A technique which evaluates the gradient of the cost functional is explained in the next subsection. In general this technique is known as adjoint calculus. It involves solution of a system of linear equations for a new set of introduced variables known as adjoint variables. Hence, the gradient evaluation includes computation of expressions involving adjoint variables.

2.2.1 Discrete Adjoint Equations

To avoid superfluous notations let us first consider a general minimization problem (2.2.6) and then use the result for optimal control problem (2.1.5). Analogous to the explanation given by Bonnans et. al in [11], the methodology for deriving the system of adjoint

equations is presented here.

Consider the problem

$$\min_{\alpha} \mathcal{F}(\alpha) \text{ subject to (2.2.7)} \tag{2.2.6}$$

where $\alpha \in \mathbb{R}^l$ is the control variable and the state equation is given by

$$\begin{aligned} y_t &= F_t(y_{t-1}, \alpha), \quad \text{for} \quad t = 1, \ldots, T, \\ y_0 &= y(0) \qquad \text{given.} \end{aligned} \tag{2.2.7}$$

where y_t are state variables and y_0 is the initial value given at time $t = 0$. For each t, F_t is a differentiable, nonlinear function $F_t : \mathbb{R}^m \times \mathbb{R}^l \to \mathbb{R}^m$. The state variable is sometimes given in the differential form. In such a case one can use continuous adjoint equations which are presented in appendix A.

Further it is assumed that the differentiable cost functional, $\mathcal{F} : \mathbb{R}^m \times \mathbb{R}^l \to \mathbb{R}$, has the following form,

$$\mathcal{F} = \sum_{t=1}^{T} \mathcal{F}_t(y_t, \alpha). \tag{2.2.8}$$

It is assumed that $\mathcal{F}$ and F_t are smooth functions. Let $g_t \in \mathbb{R}^l$ be the gradient of $\mathcal{F}$ with respect to the control variable α and which is to be evaluated. The optimality system can be derived as follows. On differentiating (2.2.7) and using the differential notations $u = d\alpha \in \mathbb{R}^l$, $z = dy \in \mathbb{R}^m$ one obtains

$$\begin{aligned} z_t &= (F_t)'_y(y_{t-1}, \alpha) z_{t-1} + (F_t)'_\alpha(y_{t-1}, \alpha) u_t, \\ z_0 &= 0. \end{aligned} \tag{2.2.9}$$

Herein, the $(F_t)'_y(y_{t-1}, \alpha) \in \mathbb{R}^{m \times m}$ and $(F_t)'_\alpha(y_{t-1}, \alpha) \in \mathbb{R}^{m \times l}$ are Jacobian.

Similarly, on differentiating (2.2.8) one obtains

$$d\mathcal{F} = \sum_{t=1}^{T} (\nabla_y \mathcal{F}_t(y_t, \alpha), z_t)_m + \sum_{t=1}^{T} (\nabla_\alpha \mathcal{F}_t(y_t, u_t), u_t)_l. \tag{2.2.10}$$

This involves additional unknowns, z_t. To obtain the adjoint equation z is eliminated in the following way:

1. Let us set the following notations:

   $$G_t = (F_t)'_y(y_{t-1}, \alpha), \quad H_t = (F_t)'_\alpha(y_{t-1}, \alpha), \quad \gamma_t = \nabla_y \mathcal{F}_t(y_t, u_t), \quad h_t = \nabla_\alpha \mathcal{F}_t(y_t, u_t).$$

2. Multiplying each linearized state equation in (2.2.9) by a vector $p_t \in \mathbb{R}^m$ and summing up over t,

   $$0 = -(p_T, z_T) + \sum_{t=1}^{T-1} (p_t, z_t)_m + \sum_{t=1}^{T-1} (G_{t+1}^\top p_{t+1}, z_t)_m + \sum_{t=1}^{T} (H_t^\top p_t, u_t)_l. \tag{2.2.11}$$

3. Adding (2.2.10) and (2.2.11) one get,

$$d\mathcal{F} = (-p_T+\gamma_T, z_T)_m+\sum_{t=1}^{T-1}(-p_t+G_{t+1}^{\top}p_{t+1}+\gamma_t, z_t)_m+\sum_{t=1}^{T}(H_t^{\top}p_t+h_t, u_t)_l. \quad (2.2.12)$$

4. To eliminate z_t from (2.2.12), p is chosen in such a way that coefficients of z_t vanish. This implies that the coefficients of z_t are equated to zero. This results in the following system of linear equations,

$$\left.\begin{array}{rcl} p_T & = & \gamma_T, \\ p_t & = & G_{t+1}^{\top}p_{t+1}+\gamma_t \quad \text{for} \quad t = T-1, \ldots, 1. \end{array}\right\} \quad (2.2.13)$$

5. The differential form of $\mathcal{F}$ is given by $d\mathcal{F} = (g, u)$. As a result the gradient is obtained in the following form

$$g_t = H_t^{\top}p_t + h_t, \quad \text{for} \quad t = 1, \ldots, T. \quad (2.2.14)$$

Equations (2.2.13) and (2.2.14) are called the **adjoint equation** and the **gradient equation** respectively. p_t involved in these equations is known as an adjoint variable. This adjoint variable p_t can be obtained by solving the equation (2.2.13) recursively backward in time. The evolution of gradient vector is obtained by solving the equation (2.2.14) recursively forward in time for gradient vector. Hence, gradient of cost functional is given as sum over time, i.e.,

$$g = \sum_{t=1}^{T} g_t(y, \alpha). \quad (2.2.15)$$

The above discussion is applied to the minimization problem (2.1.5). Considering the ODE–model (1.3.28), the state variables are denoted by

$$y_t^j = \begin{pmatrix} \rho_j^{(a)}(t) \\ \rho_j^{(b)}(t) \end{pmatrix} = \begin{pmatrix} y_t^{1,j} \\ y_t^{2,j} \end{pmatrix}, \quad \forall j = 1, \ldots, n_E. \quad (2.2.16)$$

The dimension of the state variable $y_t \in \mathbb{R}^m$ for each $'t'$ is $m = 2n_E$ where n_E is the number of roads in the network. The control variable is $\alpha \in \mathbb{R}^l$ where l is the number of dispersing junctions in the network. The control variable in the following derivations is treated as independent of time. Here, static flux steering parameter is considered only. Even for time dependent controls $\alpha(t)$ the following adjoint equations remain unchanged.

The abstract calculus introduced above is applied to the ODE model and derivatives G_t, H_t, γ_t and h_t appearing in adjoint equation and gradient equations corresponding to the minimization problem (2.1.5) are computed. Equations (1.3.28) can be rewritten as

$$y_t^{1,j} = F_t^{1,j}(y_{t-1}^{1,j}, y_{t-1}^{2,j}, \alpha),$$
$$y_t^{2,j} = F_t^{2,j}(y_{t-1}^{1,j}, y_{t-1}^{2,j}, \alpha),$$

$$y_t^j = F_t^j(y_{t-1}^j, \alpha) \quad \text{for} \quad t = 1, \ldots, T. \tag{2.2.17}$$

The non–zero elements in G_t are discussed. The block in G_t corresponding to incoming road 1 to the network is

$$(G_{t+1})_1 = \left[\begin{array}{cc} 0 & \left(\frac{1}{2} - \frac{2\tau}{L} f_1'(\rho_1^{(b)}(t))\right) \end{array} \quad 0 \cdots 0 \right], \tag{2.2.18}$$

and the block corresponding to the outgoing road I from the network is

$$(G_{t+1})_{2I} = \left[\begin{array}{cc} 0 \cdots 0 & \left(\frac{1}{2} + \frac{2\tau}{L} f_I'(\rho_I^{(a)}(t))\right) \end{array} \qquad \left(\frac{1}{2} - \frac{2\tau}{L} f_I'(\rho_I^{(b)}(t))\right) \right]. \tag{2.2.19}$$

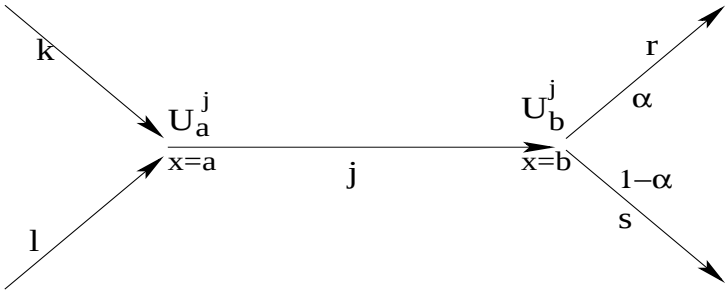

Figure 2.2.1: Coupling condition at the junction.

The block corresponding to the road j as in figure 2.2.1 is:

$$(G_{t+1})_{2j-1,2j} = \left[\begin{array}{cc} \left(\frac{1}{2} + \frac{2\tau}{L} f_j'(\overline{\rho}_j^a(t))\right) \frac{\partial \overline{\rho}_j^a(t)}{\partial \rho_{j,k,l}^{(a)}(t)} & \frac{1}{2} - \frac{2\tau}{L} f_j'(\rho_j^b(t)) \\ \frac{1}{2} + \frac{2\tau}{L} f_j'(\rho_j^a(t)) & \left(\frac{1}{2} - \frac{2\tau}{L} f_j'(\overline{\rho}_j^b(t))\right) \frac{\partial \overline{\rho}_j^b(t)}{\partial \rho_{j,r,s}^{(b)}(t)} \end{array} \right]. \tag{2.2.20}$$

The non–zero elements of H_t are given by

$$(H_t)_{ji} = \left[\begin{array}{c} \left(\frac{1}{2} + \frac{2\tau}{L} f_j'(\overline{\rho}_j^a)\right) \frac{\partial \overline{\rho}_j^a}{\partial \alpha_i} \\ \left(\frac{1}{2} - \frac{2\tau}{L} f_j'(\overline{\rho}_j^b)\right) \frac{\partial \overline{\rho}_j^b}{\partial \alpha_i} \end{array} \right]. \tag{2.2.21}$$

The derivatives of cost functional $\mathcal{J}_t$ are given by the following formulas

$$\gamma_t = \nabla_y \mathcal{J}_t = \frac{b-a}{2} \tau \left[\begin{array}{c} 1 \end{array} \right]_{m \times 1} \tag{2.2.22}$$

$$h_t = \nabla_{\alpha_i} \mathcal{J}_t = \left[\begin{array}{c} 0 \end{array} \right]_{l \times 1} \tag{2.2.23}$$

The value of boundary controls at the junction are defined by functions, $U_a^j = \bar{\rho}_j^a$ and $U_b^j = \bar{\rho}_j^b$ in equation (1.2.16). It remains to discuss the derivatives of boundary controls U_a^j, U_b^j with respect to $\rho^{(a)}$, $\rho^{(b)}$ and α_i for $i = 1 \cdots l$. The dispersing and merging junction are distinguished according to the discussion in the section 1.3.1. Due to the possibility of backward moving waves, the derivatives are discontinuous.

For dispersing junction with control parameter α_i, one incoming road 1 and two outgoing roads $2, 3$ it can be written as

$$\begin{aligned}\nabla U^1(\rho_1^{(b)},\rho_2^{(a)},\rho_3^{(a)},\alpha_i) \;=\; &\Bigg(\begin{bmatrix} 1 & \gamma = c_1, \rho_1^{(b)} < \sigma_1 \\ 0 & \text{else}\end{bmatrix},\\ &\begin{bmatrix} d_{\rho_2^{(a)}} f_1^{-1,+}(\frac{1}{\alpha_i} f_2(\rho_2^{(a)})) & \gamma = \frac{c_2}{\alpha_i}, \rho_2^{(a)} > \sigma_2 \\ 0 & \text{else}\end{bmatrix},\\ &\begin{bmatrix} d_{\rho_3^{(a)}} f_1^{-1,+}(\frac{1}{1-\alpha_i} f_3(\rho_3^{(a)})) & \gamma = \frac{c_3}{1-\alpha_i}, \rho_3^{(a)} > \sigma_3 \\ 0 & \text{else}\end{bmatrix},\\ &\begin{bmatrix} 0 & \gamma = c_1 \\ d_{\alpha_i} f_1^{-1,+}(\frac{1}{\alpha_i} c_2) & \gamma = \frac{c_2}{\alpha_i} \\ d_{\alpha_i} f_1^{-1,+}(\frac{1}{1-\alpha_i} c_3) & \gamma = \frac{c_3}{1-\alpha_i}\end{bmatrix}\Bigg),\end{aligned}$$

$$\begin{aligned}\nabla U^2(\rho_1^{(b)},\rho_2^{(a)},\rho_3^{(a)},\alpha_i) \;=\; &\Bigg(\begin{bmatrix} d_{\rho_1^{(b)}} f_2^{-1,-}(\alpha_i f_1(\rho_1^{(b)})) & \gamma = c_1, \rho_1^{(b)} < \sigma_1 \\ 0 & \text{else}\end{bmatrix},\\ &\begin{bmatrix} 1 & \gamma = \frac{c_2}{\alpha_i}, \rho_2^{(a)} > \sigma_2 \\ 0 & \text{else}\end{bmatrix},\\ &\begin{bmatrix} d_{\rho_3^{(a)}} f_2^{-1,-}(\frac{\alpha_i}{1-\alpha_i} f_3(\rho_3^{(a)})) & \gamma = \frac{c_3}{1-\alpha_i}, \rho_3^{(a)} > \sigma_3 \\ 0 & \text{else}\end{bmatrix},\\ &\begin{bmatrix} d_{\alpha_i} f_2^{-1,-}(\alpha_i c_1) & \gamma = c_1 \\ 0 & \gamma = \frac{c_2}{\alpha_i} \\ d_{\alpha_i} f_2^{-1,-}(\frac{\alpha_i}{1-\alpha_i} c_3) & \gamma = \frac{c_3}{1-\alpha_i}\end{bmatrix}\Bigg),\end{aligned}$$

$$\begin{aligned}\nabla U^3(\rho_1^{(b)},\rho_2^{(a)},\rho_3^{(a)},\alpha_i) \;=\; &\Bigg(\begin{bmatrix} d_{\rho_1^{(b)}} f_3^{-1,-}((1-\alpha_i) f_1(\rho_1^{(b)})) & \gamma = c_1, \rho_1^{(b)} < \sigma_1 \\ 0 & \text{else}\end{bmatrix},\\ &\begin{bmatrix} d_{\rho_2^{(a)}} f_3^{-1,-}(\frac{1-\alpha_i}{\alpha_i} f_2(\rho_2^{(a)})) & \gamma = \frac{c_2}{\alpha_i}, \rho_2^{(a)} > \sigma_2 \\ 0 & \text{else}\end{bmatrix},\\ &\begin{bmatrix} 1 & \gamma = \frac{c_3}{1-\alpha_i}, \rho_3^{(a)} > \sigma_3 \\ 0 & \text{else}\end{bmatrix},\\ &\begin{bmatrix} d_{\alpha_i} f_3^{-1,-}((1-\alpha_i) c_1) & \gamma = c_1 \\ d_{\alpha_i} f_3^{-1,-}(\frac{1-\alpha_i}{\alpha_i} c_2) & \gamma = \frac{c_2}{\alpha_i} \\ 0 & \gamma = \frac{c_3}{1-\alpha_i}\end{bmatrix}\Bigg).\end{aligned}$$

For a merging junction where the roads 1 and 2 merge in road 3, one obtains

$$\begin{aligned}\nabla U^1(\rho_1^{(b)},\rho_2^{(b)},\rho_3^{(a)}) &= \Bigg(\begin{bmatrix} 1 & \gamma = c_1, \rho_1^{(b)} < \sigma_1 \\ 0 & \text{else} \end{bmatrix}, \\ &\quad \begin{bmatrix} d_{\rho_2^{(b)}} f_1^{-1,+}(f_2(\rho_2^{(b)})) & \gamma = c_2, \rho_2^{(b)} < \sigma_2 \\ 0 & \text{else} \end{bmatrix}, \\ &\quad \begin{bmatrix} d_{\rho_3^{(a)}} f_1^{-1,+}(\frac{f_3(\rho_3^{(a)})}{2}) & \gamma = \frac{c_3}{2}, \rho_3^{(a)} > \sigma_3 \\ 0 & \text{else} \end{bmatrix} \Bigg),\end{aligned}$$

$$\begin{aligned}\nabla U^2(\rho_1^{(b)},\rho_2^{(b)},\rho_3^{(a)}) &= \Bigg(\begin{bmatrix} d_{\rho_1^{(b)}} f_2^{-1,+}(f_1(\rho_1^{(b)})) & \gamma = c_1, \rho_1^{(b)} < \sigma_1 \\ 0 & \text{else} \end{bmatrix}, \\ &\quad \begin{bmatrix} 1 & \gamma = c_2, \rho_2^{(b)} < \sigma_2 \\ 0 & \text{else} \end{bmatrix}, \\ &\quad \begin{bmatrix} d_{\rho_3^{(a)}} f_2^{-1,+}(\frac{f_3(\rho_3^{(a)})}{2}) & \gamma = \frac{c_3}{2}, \rho_3^{(a)} > \sigma_3 \\ 0 & \text{else} \end{bmatrix} \Bigg).\end{aligned}$$

For ∇U^3 two different cases are considered: If $c_1 + c_2 \leq c_3$ then $\gamma_3 = c_1 + c_2$ and the gradient is given by

$$\begin{aligned}\nabla U^3(\rho_1^{(b)},\rho_2^{(b)},\rho_3^{(a)}) = \Big(& d_{\rho_1^{(b)}} f_3^{-1,-}(f_1(\rho_1^{(b)}) + f_2(\rho_2^{(b)})), \\ & d_{\rho_2^{(b)}} f_3^{-1,-}(f_1(\rho_1^{(b)}) + f_2(\rho_2^{(b)})), \\ & d_{\rho_3^{(a)}} f_3^{-1,-}(f_1(\rho_1^{(b)}) + f_2(\rho_2^{(b)})) = 0 \Big).\end{aligned}$$

On the other hand, if $\gamma_3 = \min(c_1, c_2, \frac{c_3}{2})$ then the gradient is given by

$$\begin{aligned}\nabla U^3(\rho_1^{(b)},\rho_2^{(b)},\rho_3^{(a)}) = \Bigg(& \begin{bmatrix} d_{\rho_1^{(b)}} f_3^{-1,-}(f_1(\rho_1^{(b)})) & \gamma = c_1, \rho_1^{(b)} < \sigma_1 \\ 0 & \text{else} \end{bmatrix}, \\ & \begin{bmatrix} d_{\rho_2^{(b)}} f_3^{-1,-}(f_2(\rho_2^{(b)})) & \gamma = c_2, \rho_2^{(b)} < \sigma_2 \\ 0 & \text{else} \end{bmatrix}, \\ & \begin{bmatrix} 1 & \gamma = \frac{c_3}{2}, \rho_1^{(a)} > \sigma_3 \\ 0 & \text{else} \end{bmatrix} \Bigg).\end{aligned}$$

This finishes the discussion of gradient and adjoint equations for the optimization problem (2.1.5) corresponding to the ODE model.

Remark 2.3 *An example for the flux function $f_j(\rho_j) = 4\rho_j(1-\rho_j/\rho_{max,j})$ is being considered (setting $\rho_{max,j} = M_j$),*

$$d_{\rho_k} f_l^{-1,\pm}(\nu(\alpha_i).f_k(\rho_k)) = \frac{\mp 1}{4}\left(\frac{M_l \nu(\alpha_i).f_k'(\rho_k)}{\sqrt{M_l^2 - M_l.\nu(\alpha_i).f_k(\rho_k)}}\right) \tag{2.2.24}$$

$$d_{\alpha_i} f_l^{-1,\pm}(\nu(\alpha_i)c_k) = \frac{\mp 1}{4}\left(\frac{M_l \partial_{\alpha_i}\nu(\alpha_i)c_k}{\sqrt{M_l^2 - M_l\nu(\alpha_i)c_k}}\right) \tag{2.2.25}$$

where

$$\partial_{\alpha_i}\nu(\alpha_i) = \begin{cases} 1 & if \quad \alpha_i \\ -1 & if \quad 1-\alpha_i \\ \frac{-1}{\alpha_i^2} & if \quad \frac{1}{\alpha_i} \quad or \quad \frac{1-\alpha_i}{\alpha_i} \\ \frac{1}{(1-\alpha_i)^2} & if \quad \frac{1}{1-\alpha_i} \quad or \quad \frac{\alpha_i}{1-\alpha_i} \end{cases} \tag{2.2.26}$$

and

$$f_k'(\rho_k) = 4\left(1 - \frac{2\rho_k}{\rho_{max,k}}\right) \tag{2.2.27}$$

While simulating, the network is calculated starting from the road 1 and advancing through the other roads. Using the above formulas it is easier to evaluate the gradient of cost functional for arbitrary networks. The above formulas rely only on the roads connected to and from a junction.

2.3 Optimization Problem for the RSA Model

In this section an optimization problem for the simplified dynamics of the traffic flow in networks is defined. The cost functional measuring minimal traveling time of cars in the network with respect to reformulated simplified algebraic model is deduced from (2.1.1) under the restrictions of the simplified algebraic model.

2.3.1 Cost Functional

The simplified algebraic model presented in section 1.4 (in chapter 1) is formulated based on assumptions that there is only one shock wave on each road j and no backward moving shock waves are allowed in the network. A single wave is assumed to travel with speed s_j and reaches road j at time t_j. Hence, summing up for all roads in the network and on rewriting the equivalent form of (2.1.1) as in [44],

$$\mathcal{J}(\alpha; T, \rho_0) = \sum_{j=1}^{n_E}(T - t_{j,0})(b_j - a_j)\rho_{j,0} - \frac{\rho_{j,0}}{2s_j}(b_j - a_j)^2, \tag{2.3.28}$$

where T is the overall time-horizon for the optimization, ρ_0 is the inflow given to the network and $\rho_{j,0}$, t_j are given by equations (1.4.39)-(1.4.44) and each road j in the network

is modelled by $[a_j, b_j]$. The control parameter α at the dispersing junction appears in expressions for $\rho_{j,0}$ is given by (1.4.40). The equation (2.3.28) is the cost functional of the simplified algebraic model. In [44] the definition of cost functionals (2.1.1) and (2.3.28) are compared and it was found that they show same qualitative behavior under assumptions. The optimal control problem for the simplified algebraic model is also a nonlinear bound constrained optimization problem stated as

$$\min_{\alpha} \mathcal{J}(\alpha; T, \rho_0) \quad \text{subject to bound constraints} \quad 0 \leq \alpha \leq 1.$$

Hence, the cost functional for the RSA model can be obtained by reformulating (2.3.28) in terms of flux q_j. First a family of functions τ_j describing the transit times [47] which is the average time the flux q_j needs to pass the road j in the network is introduced. $\tau_j(q_j)$ are nonlinear functions depending on the road j and the actual flux q_j. This means $\tau_j(q_j) = 1/v(f_j^{-1}(q_j))$ and the relation with the corresponding density is $\rho_j = q_j\tau_j(q_j)$. A typical example for τ_j is

$$\tau_j(q_j) = \frac{1}{2\left(1 + \sqrt{1 - q_j/q_{c_j}}\right)}, \tag{2.3.29}$$

corresponding to the flux function

$$q_j = 4\rho_j\left(1 - \frac{\rho_j}{\rho_{max,j}}\right). \tag{2.3.30}$$

After expressing (2.3.28) in terms of q_j and using τ_j, one obtains the cost functional for the reformulated simplified algebraic model,

$$\mathcal{J}(q; T, q_0) = \sum_{j=1}^{n_E}\left(T - \frac{\tau_j(q_j)}{2}\right)\tau_j(q_j)q_j, \tag{2.3.31}$$

where q_0 is the incoming flux to the network. It is to note that the control variable α does not appear in the above formulation and the optimization variable is q_j only. In addition q_j need to satisfy linear coupling constraints (1.5.48) at the junction and bound constraints (1.5.45). Hence, the constrained problem can be stated as

$$\begin{aligned} \min_{q} \quad \mathcal{J}(q; T, q_0) &= \sum_{j=1}^{n_E}\left(T - \frac{\tau_j(q_j)}{2}\right)\tau_j(q_j)q_j \qquad (2.3.32)\\ &\text{subject to}\\ 0 \leq q_j &\leq q_{c_j}\\ \text{and} \quad h(q) &= H^T q + h_0 = 0 \end{aligned}$$

Remark 2.4 *The cost functional and the linear constraints are derived from the macroscopic LWR–model. The equations (2.3.31,1.5.45,1.5.48) are a simplification of the nonlinear, complex system of scalar hyperbolic equations as described in chapter 1 and the simplified model is valid under certain restrictions only. For a comparison do refer to the work done by Herty & Klar in [44].*

At the end the nonlinear optimization problems subject to bound and linear equality constraints (2.3.32) are obtained. Here, one has to take into account two contradictory goals; first is minimizing the functional and second is satisfying constraints. One approach will be dealing with both goals simultaneously. This leads to use of the concept of the penalty methods. The penalty method transforms the constrained optimization problem into an unconstrained or bound constrained problem which is comparatively easy to solve. Further details on the penalty methods are deferred to the chapter 3.

In order to optimize the above defined optimization problem (2.3.32) gradients of cost functional $\mathcal{J}(q)$ (2.3.31) with respect to q are needed.

2.3.2 Gradient of Cost Functional

Gradient of $\nabla\mathcal{J}(q)$ for the flux defined by (2.3.30) are evaluated as below,

$$\nabla\mathcal{J}(q) = \sum_{j=1}^{n_E} \partial_{q_j}\mathcal{J}(q_j), \tag{2.3.33}$$

where

$$\begin{aligned}
\partial_{q_j}\mathcal{J}(q_j) &= -q_{c_j}^3 \Big(3q_{c_j}q_jT + Tq_jB(q_{c_j},q_j) - 4Tq_{c_j}B(q_{c_j},q_j) \\
&\quad 4Tq_{c_j}^2 + 2q_{c_j}B(q_{c_j},q_j) + 2q_{c_j}^2 \Big) / (q_{c_j} + (q_{c_j},q_j))^3 B(q_{c_j},q_j), \\
\text{where} &\quad B(q_{c_j},q_j) = \sqrt{q_{c_j}^2 - q_{c_j}q_j}.
\end{aligned}$$

2.4 Note on Bound Constrained Optimization

The management of traffic flow in the network leads to optimization problems as defined in sections (2.1) and (2.3). The optimization problem (2.1.5) is a bound constrained problem. Also the problem (2.3.32) can be transformed into a bound constrained optimization problem using the concept of penalty functions. In order to obtain optimal values bound constrained optimization methods are used to solve the optimization problems. Let us consider in general a bound constrained problem to be solved,

$$\min_x \mathcal{J}(x) \quad \text{subject to} \quad l_i \le x_i \le u_i \quad i \in \{1, 2, \cdots, N\}. \tag{2.4.34}$$

The basic idea of numerical optimization algorithms to deal with simple bound constraints is presented in this section. Let the feasible set for the bound optimization problem is represented by,

$$\Omega = \{x \in \mathbb{R}^N \mid l_i \le x_i \le u_i\}.$$

Projection: Let $\mathcal{P}$ denotes the projection onto feasible set Ω defined by the simple bound constraints , i.e., the map takes x into the nearest point in Ω to x. The mapping is defined

as

$$\mathcal{P}(x)_i = \begin{cases} l_i, & x_i \leq l_i, \\ x_i, & l_i < x_i < u_i, \\ u_i, & x_i \geq u_i, \end{cases} \tag{2.4.35}$$

The i^{th} constraint is active at $x \in \Omega$ if either $x_i = u_i$ or $x_i = l_i$, otherwise it is inactive. The set of indices i such that the i^{th} constraint is active is defined and denoted by

$$\mathcal{A}(x) = \{i, \mid x_i = l_i \quad \text{or} \quad x_i = u_i\}.$$

Similarly, the set of indices for inactive constraint $\mathcal{I}(x) = \overline{\mathcal{A}}(x)$ is the complement of $\mathcal{A}(x)$.

Necessary condition for optimality

The first order necessary condition including the bound constraints in effect is stated as below.

Theorem 2.1 (Theorem.5.2.1 of [51]) *Let $\mathcal{J}$ be twice continuously differentiable function and x^* is a local minimum of $\mathcal{J}$ subject to Ω. Then*

$$\nabla\mathcal{J}(x^*)(x - x^*) \geq 0 \quad \forall x \in \Omega(\tag{2.4.36}$$

□

2.4.1 Steepest Descent Method for Bound Constrained Optimization with Armijo Rule

Steepest descent method is one of the gradient based optimization method where the step size t is chosen in such a way that the maximum decrease of the cost functional is achieved at each step. The direction in which the gradient $\nabla\mathcal{J}(x)$ points is the direction of maximum rate of increase of $\nabla\mathcal{J}$ at x. Hence, the $-\nabla\mathcal{J}(x)$ is the direction of maximum rate of decrease. Let g denotes the gradient of $\mathcal{J}$ functional to be minimized, which can be calculated w.r.t to x, $g = \nabla\mathcal{J}$. An iterative scheme can be written as follows [11, 21, 30, 51]:

$$x_{k+1} = \mathcal{P}(x - tg(x_k)), \quad \text{where} \quad t > 0.$$

The new iterate x_{k+1} is projected onto the feasible set using projection operator $\mathcal{P}$ as defined by (2.4.35). The step size t is computed according to the general principle $\mathcal{J}(x_{k+1}) < \mathcal{J}(x_k)$ known as *descent property*. One can choose the step size using **Armijo strategy** [2]. Let $t = \beta^m$, where $0 < \beta < 1$ and $m \geq 0$ is the smallest nonnegative integer such that there is *sufficient decrease* in $\mathcal{J}$, which is evaluated by

$$\mathcal{J}(x_k - t\nabla\mathcal{J}(x_k)) - \mathcal{J}(x_k) < -\eta t(||\nabla\mathcal{J}(x_k)||_2)^2, \tag{2.4.37}$$

where $\eta \in (0, 1)$ is a parameter and is chosen to be $\eta = 10^{-4}$ as in [51].

Termination of iteration [51]

A practical numerical stopping criteria for the iteration of x_{k+1} is to check whether the following inequality holds,

$$||\nabla \mathcal{J}(x_{k+1})|| < \epsilon$$

where ϵ is the prespecified threshold. This is true only for problems without bound constraints. This termination criteria has to be modified for bound constrained problems. For bound constrained problems $\nabla \mathcal{J}$ need not be zero at the solution, a natural substitute is to terminate the iteration if the difference between x and $x(1)$ is small. Let τ_r define the desired reduction. To avoid the situation that algorithm may not terminate a natural remedy is to use the relative error criterion. Given $r_0 = ||x_0 - x_0(1)||$, the termination criteria based on relative and absolute tolerances is

$$||x - x(1)|| \leq \tau_a + \tau_r r_0. \tag{2.4.38}$$

where τ_a is a absolute error tolerance and $x(t) = \mathcal{P}(x - tg(x))$.
When the simulator is ready to evaluate $\mathcal{J}$ and $\nabla \mathcal{J}$, a bound constrained optimization algorithm is as follows :

Algorithm: Steep(x, $\mathcal{J}$, τ)

Step 0 : (Initialization) Set the initial iterate x_k, k=0 and a stopping tolerance (τ_a, τ_r).

Step 1 : Compute $r_0 = ||x_k - x_k(1)||$

Step 2 : Compute $\mathcal{J}$ and $g_k = \nabla \mathcal{J}$ w.r.t to initial x_k.

Step 3 : At each iteration:

1. Perform the stopping test for (2.4.38): STOP, otherwise proceed.
2. Initialize the step size β.
3. Perform **Line Search** : (knowing β, x_k, $\mathcal{J}(x_k)$, g_k) using Armijo rule.
4. Set $x_{k+1} = \mathcal{P}(x_k - tg_k)$; $k = k + 1$.
5. Go to **Step 2**.

Algorithm: Line Search(β, x, $\mathcal{J}(x)$, g)

Step 0 : Set $m = 0$, $\eta = 10^{-4}$, $t_{new} = \beta$.

Step 1 : do

1. Compute $||\nabla \mathcal{J}(x)||_2$.
2. $t_c = t_{new}$.
3. $x_{new} = \mathcal{P}(x - g.t_c)$
4. Compute $\mathcal{J}(x_{new})$.

5. Set $m = m + 1$ and compute $t_{new} = t_c\beta$.

while (2.4.37) not satisfied

Step 3: return t_c

The above defined algorithm is an inherited implementation of steepest descent algorithm to bound constrained problems. It is also known as **Gradient Projection algorithm**. It shares all properties of steepest algorithm for unconstrained problems.

2.4.2 Scaled Projected Gradient Method with Armijo Rule

The idea in the above defined gradient projection algorithm does not work equally well to iterations of the form

$$x_{new} = \mathcal{P}(x_c - tH_c^{-1}\nabla\mathcal{J}(x_c)),$$

where H_c is a symmetric positive definite (spd) matrix. Unlike the unconstrained case positive definiteness of Hessain H_c is not sufficient. According to [51] one needs to compute the active set and the reduced Hessian. For

$$0 \leq \epsilon < min(u_i - l_i)/2.$$

Let us define

$$\mathcal{A}^{\epsilon}(x) = \{i \mid u_i - x_i \leq \epsilon, \quad or \quad x_i - l_i \leq \epsilon\}.$$

Thus for given ϵ and a Hessian matrix $H_c = \nabla^2\mathcal{J}(x)$ we model the reduced Hessian R_c by

$$R_c = \begin{cases} \delta_{ij}, & i \in \mathcal{A}^{\epsilon}(x_c), \quad or \quad j \in A^{\epsilon}(x_c), \\ (H_c)_{ij}, & otherwise. \end{cases} \tag{2.4.39}$$

The explicit dependence on x_c, ϵ and H_c is important, so write it as $R(x_c, \epsilon, H_c)$. Thus, the new iterate for x is given by

$$x_{new} = \mathcal{P}(x_c - tR_c(x_c, \epsilon, H_c)^{-1}\nabla\mathcal{J}(x_c)). \tag{2.4.40}$$

It is required that $\mathcal{J}(x_{new}) < \mathcal{J}(x_c)$ for sufficiently small t.

An algorithm based on these ideas is known as **Scaled Gradient Projection Algorithm**. The algorithm is as follows:

Algorithm: Scalgradproj(x_0, $\mathcal{J}$, τ)

Step 0 : (Initialization) Set the initial iterate x_0, k=0.

Step 1 : Compute $\mathcal{J}$ and $g_0 = \nabla\mathcal{J}(x_0)$.

Step 2 : Compute $r_0 = ||x_0 - x_0(1)||$.

step 3 : Set $0 \leq \epsilon < min(u_i - l_i)/2$.

Step 4 : At each iteration:

1. Perform the stopping test: if $||x - x(1)|| \leq \tau_a + \tau_r r_0$ stop, otherwise proceed.
2. Compute $H_k = \nabla^2 \mathcal{J}(x_k)$.
3. Compute Reduced Hessian $R(x_k, \epsilon, H_k) \quad using \quad (2.4.39)$.
4. Solve $R(x_k, \epsilon, H_k)d_k = -g_k$.
5. Initialize the stepsize β.
6. Perform **Line Search** : (knowing β, x_k, $J2(x_k)$, d_k) Find the smallest integer $m \geq 0$ such that Armijo rule holds for $t = \beta^m$.
7. Set $x_{k+1} = \mathcal{P}(x_k + td_k)$ as in (2.4.40)
8. Compute $\mathcal{J}(x_k)$ and $g_k = \nabla \mathcal{J}(x_k)$, go to first step in **Step 4**.

Note: The computation of the Hessian approximations are performed using different updates of H_k, for example finite difference Newton's method or Quasi-Newton-BFGS method as explained below.

Finite-difference Projected Newton's method

In Projected Newton's method, the Hessian matrix $\nabla^2 \mathcal{J}(x)$ at $x \in \mathbb{R}^N$ is approximated by the finite difference of gradients along N linearly independent directions. In computations central finite difference approximations for evaluating the Hessian matrix H_c are used.

It is known that the Hessian matrix provides the useful curvature information. However, the major drawback for the Hessian matrix is that it is computationally very expensive for large N or in some cases it is tedious to compute analytically or by finite difference approximations.

Quasi-Newton method: BFGS - Armijo

Quasi-Newton methods do not compute the Hessian matrix, but it generates a series of Hessian matrix approximations [11, 15, 21, 51]. A nonsingular approximation to the Hessian is maintained say $H \in \mathbb{R}^N$ which satisfies the so called Newton-equation,

$$H_{k+1}\Delta x_k = \Delta g_k,$$

where

$$\Delta x_k = x_{k+1} - x_k,$$

and

$$\Delta g_k = g_{k+1} - g_k.$$

Then, the search direction is given by the solution of $R_k(x_k, \epsilon, H_k)d = -g_k$ where R_k is the Reduced Hessian. Let us choose the initial Hessian approximation H_0 as a symmetric positive definite matrix, practically it is usually the identity matrix. Hence, first iteration is nothing but the steepest descent iteration. After x_{k+1} has been computed, a new Hessian approximation H_{k+1} is obtained using the BFGS update formula [51],

$$H_{k+1} = H_k + \frac{\Delta g_k \Delta g_k^T}{\Delta g_k^T \Delta x_k} - \frac{H_k \Delta x_k \Delta x_k^T H_k}{\Delta x_k^T H_k \Delta x_k} \tag{2.4.41}$$

The BFGS update maintains the positive definiteness of the Hessian matrix. Unlike other Quasi-Newton updates, the BFGS update is reasonably robust for change in the steplength in line search (converges for less number of iterations for different values of initial steplength values to the line search).

If $H_k = \nabla_R^2 f(\alpha_k)$, then the method converges Q-quadratically to x^*. But if one uses BFGS update formula for the Hessian matrix, then the method converges Q-superlinearly. In case of the steepest descent method where $H_k = I$ the convergence is only linear.

Chapter 3

Smoothed Exact Penalty Algorithm

In this chapter a numerical optimization algorithm is introduced to solve the equality and bound constrained problems as appeared for example in chapter 2 for the traffic flow in the road networks. This chapter starts with the basic theory and necessary conditions for the existence of a local optimum of the constrained optimization problem. The proposed algorithm is based on smoothing of the non-differentiable exact l_1-penalty function and solving the resulting problem by a bound constrained optimization method. This algorithm is an extension for the work done by P. Spellucci in [94]. The theoretical estimates on how to update the smoothing and penalty parameter involved in the numerical optimization algorithm are given.

3.1 Introduction

An algorithm for solving the nonlinear optimization problems supplemented with only equality and bound constraints will be introduced. Particularly for those problems that arise in the context of the traffic flow in the road networks [43, 47]. This algorithm can extend any bound constrained optimization method to the solve equality and bound constrained optimization problems. The proposed method is purely primal.

One of the popular methods of finding a constrained minimum point is the method of penalty functions. A given constrained problem with the equality and bound constraints is transformed into a bound constrained problem by adding a penalty term to the cost functional. The exact penalty methods have been under investigation for several years, see [9, 13, 37, 72, 82, 100] and references therein. Although the history of these functions is quite rich, there is a topic of controversy. The root cause for this is the non-differentiable nature of these functions, since the most powerful methods in non-linear programming require differentiable functions. This motivates to use the smoothing concept [66] for classical exact penalty functions.

The model problem of interest is related to the traffic flow on the road networks. In particular there exists a hierarchy of different models and corresponding optimization problems in the context of traffic flow on the road networks in [31, 44] and in sections 2.1, 2.3.

Herein, the focus is on the arising nonlinear, possibly non convex minimization problem with the equality and bound constraints. An optimization problem given in Chapter 2 by (2.3.32) and proposed in [44] is solved in this chapter.

The overview of this chapter is as follows. Section 3.2 presents briefly the supporting theoretical aspects to solve the constrained optimization problems. Section 3.3 introduces the concept of the exact penalty functions which leads to the transformation of a constrained problem into an unconstrained problem. In section 3.4 different smoothing kernels for the non-differentiable exact penalty function are considered and their properties are discussed in connection with the optimization method. The transformed problem in section 3.3 can be solved numerically by a optimization algorithm proposed in the section 3.5. This algorithm involves penalty and smoothing parameters. The appropriate choice of the initial value and the update strategy of the penalty and smoothing parameter are important. Corresponding to the traffic flow problem the possible estimate for the initial penalty parameter are presented in section 3.6.

3.2 Theoretical Background

An algorithm is described for solving the (2.3.32) model problem given as a nonlinear optimization problem with linear equality and bound constraints,

$$\min_{q\in\mathbb{R}^n} \ \mathcal{J}(q) \tag{3.2.1a}$$

$$\text{subject to} \quad h(q) = H^T q + h^0 = 0 \tag{3.2.1b}$$

$$0 \le q \le q_c \tag{3.2.1c}$$

where $\mathcal{J} : \mathbb{R}^n \to \mathbb{R}, h : \mathbb{R}^n \to \mathbb{R}^m$. The objective function $\mathcal{J}$ and constraint function h are assumed to be two times continuously differentiable on $\mathbb{R}^n$. Let us assume that $H \in \mathbb{R}^{n\times m}$ is of full rank and that a feasible point q^0 exists. It is assumed also that the intersection of a level set of $\mathcal{J}$ at q^0 with the feasible set is a compact set. This implies the existence of at least one minimizer q^* ([90], Theorem 2.41 and 4.14).

The following notations are introduced here. The affine linear function $g(q) : \mathbb{R}^n \to \mathbb{R}^{2n}$ is defined as

$$g(q) \ = \ (-q^T, (q-e)^T)^T \quad \text{with } e \ = \ (1,\ldots,1)^T$$

and constraints (3.2.1c) are reformulated as

$$g(q) \le 0. \tag{3.2.2}$$

Further on defining $g^+(q) = \max\{0, g(q)\}$ with the max function acting componentwise.

The Lagrangian to (3.2.1)is defined by

$$\mathcal{L}(q,\lambda,\mu) = \mathcal{J}(q) + h(q)^T\lambda + g(q)^T\mu \tag{3.2.3}$$

and so

$$\begin{aligned}\nabla_q \mathcal{L}(q,\lambda,\mu) &= \nabla_q \mathcal{J}(q) + D_h(q)^T\lambda + D_g(q)^T\mu \\ \nabla^2_{qq}\mathcal{L}(q,\lambda,\mu) &= \nabla^2_{qq}\mathcal{J}(q) + \sum_{i=1}^{m}\lambda_i\nabla^2_{qq}h_i(q) + \sum_{j=1}^{2n}\mu_j\nabla^2_{qq}g_j(q)\end{aligned}$$

where

$$\begin{aligned}D_h(q) &= (\nabla h_1(q), \nabla h_2(q), \ldots, \nabla h_m(q)), \\ D_g(q) &= (\nabla g_1(q), \nabla g_2(q), \ldots, \nabla g_{2n}(q))\end{aligned}$$

The second order sufficient optimality conditions for the problem (3.2.1) are as follows (see Bonnans et al., Chong [9, 11, 21]):

Theorem 3.1 (KKT Conditions) *If $(q^*, \lambda^*, \mu^*) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^{2n}$ is such that*

$$\begin{aligned}\nabla_q \mathcal{L}(q^*,\lambda^*,\mu^*) &= 0 && \text{(3.2.4a)}\\ h(q^*) &= 0, && \text{(3.2.4b)}\\ g(q^*) &\leq 0, && \text{(3.2.4c)}\\ g(q^*)^T\mu^* = 0, \quad \mu^* &\geq 0, && \text{(3.2.4d)}\\ d^T\nabla^2_{qq}\mathcal{L}(x,\lambda,\mu)d &> 0 && \text{(3.2.4e)}\end{aligned}$$

$\forall d \in \{y \in \mathbb{R}^n \mid \nabla h_j(q^*)y = 0, \nabla g_j(q)^T y = 0, j \in J\}$, where $J = \{j \mid g_j(q^*) = 0, \mu_j^* > 0\}$, then q^* is a local minimizer of problem (3.2.1). □

3.3 Exact Penalty Methods

An algorithm is needed to solve the constrained problem (3.2.1) by taking into account the presence of equality constraints. A brief explanation on the penalty methods is given for solving the constrained optimization problem using techniques from the unconstrained optimization. These methods solve the optimization problem (3.2.1) by means of the resolution of a sequence of subproblems with the simple constraints (or even unconstrained) [48, 84]. Constraints are placed in the cost function through a *penalty function*, so that any violation of such constraints is penalized. The penalty function is defined such that it is zero on the feasible set (which is defined by a set of constraints) and positive outside the feasible set.

Specifically, the constrained optimization problem is approximated by an unconstrained optimization problem which is usually easier to solve. The resulting unconstrained problem is of the form,

$$\Phi(q,\beta) = \mathcal{J}(q) + P(q,\beta) \tag{3.3.5}$$

where P is the penalty function and $\beta > 0$ is the penalty parameter. The solution to an unconstrained optimization problem

$$\min_q \Phi(q, \beta), \quad \text{for fixed} \quad \beta$$

may not be exactly equal to the solution of the original constrained problem. Whether the solution to the $\Phi(q, \beta)$ is a good approximation to the true solution of the original problem or not depends on the choice of the penalty function and the penalty parameter. The theory developed for classical penalty methods states that the sequence of penalty parameters must be unbounded in order to guarantee global convergence. This result follows from the following theorem.

Theorem 3.2 (Theorem 22.2 from [21]) *Suppose the cost functional $\mathcal{J}$ is continuous function and $\beta_k \to \infty$ as $k \to \infty$. Then, the limit of any convergent subsequence of the sequence $\{q_k\}$ is a solution to the constrained problem.* □

However, due to practical limitations and tediousness of computations penalty subproblems for large values of β cannot be solved. As a remedy to these limitations exact penalty functions are defined in the literature [9, 13, 37, 72, 82, 100].

An **exact penalty function** for the problem (3.2.1) is a function $P(q, \beta)$, where $\beta > 0$ is the penalty parameter with the property that there exists a lower bound $\overline{\beta}$ such that for $\beta > \overline{\beta}$ any local minimizer of (3.2.1) is also a local minimizer of the penalty subproblem, see [17, 37].

In the literature of nonlinear programming, the construction of non-differentiable exact penalty functions has been an attractive idea since early works by Zangwill in 1969 [100]. This lead to further advocation done in [8, 24, 37]. Also it has been proved in [8] that any exact penalty function is non-differentiable. In return, these exact penalty functions have disadvantage in the evaluation of gradients of transformed cost functional as they are no longer smooth functions.

References for the definition and properties of exact penalty functions can be found in [13, 84]; optimality conditions, characterizing local minimizers of different penalty problems in [17, 37, 83, 84]. Related algorithms can be found for example in [17] and references therein. There are special methods which are suitable to deal with nondifferentiability for example in [26, 24, 72]. Further, there exists linearization methods for exact penalty functions, for example [73]. Another approach is using smooth approximation to these nondifferentiable exact penalty functions [18, 19], which will pursue in the following.

The most commonly used exact penalty problem in the literature is l_1-penalty function

[37, 70, 72, 85]:

$$\Phi(q,\beta) = \mathcal{J}(q) + \beta\left(||h(q)||_1 + ||g^+(q)||_1\right) \tag{3.3.6}$$
$$= \mathcal{J}(q) + \beta\left(\sum_{i=1}^{m}|h_i(q)| + \sum_{i=1}^{p}|g_i^+(q)|\right)$$

Hereafter further study is performed with respect to the exact l_1-penalty function. The conditions to ensure that Φ has a local minimum at a local minimum of original constrained problem (3.2.1) for large, but finite values of β are discussed in [23]. The following result shows the necessary condition for the exactness of a bound constrained penalty problem Φ corresponding to (3.2.1). This infers that the local solutions of the nonlinear programming problem are equivalent to the local minimizers of the penalty problem [13, 17, 37].

Theorem 3.3 *Let (q^*, λ^*, μ^*) satisfy the KKT conditions (Theorem 3.1) for a local minimizer of problem (3.2.1). Then for $\beta > \bar{\beta}$ with*

$$\bar{\beta} \;=\; ||(\mu^*, \lambda^*)||_\infty$$

q^* is a strict unconstrained local minimizer of $\Phi(q;\beta)$. □

This theorem shows that the threshold value for the penalty parameter depends on the a priori *unknown* norm of the Lagrange multipliers at a local minimum. It is possible to take advantage of the structure of the simple constraints to define bound constrained penalty problems. The following result from [48] is used, which allows constrained subproblems in exact penalty methods.

Theorem 3.4 *Let q^* satisfy the KKT conditions (Theorem 3.1) for a local minimizer of problem (3.2.1) with the Lagrange multipliers λ^* and μ^* corresponding to $h(q)$ and $g(q)$ where $g(q)$ describe the simple bound constraints.*

Then there exists $\bar{\beta} > 0$ depending on λ^* only, such that for $\beta > \bar{\beta}$, q^* is a strict local minimizer of the problem

$$\min_{q\in\mathbb{R}^n} \mathcal{J}(q) + \beta\left(\|h(q)\|_1\right) \;\; \textit{subject to} \tag{3.3.7a}$$
$$g(q) \leq 0 \tag{3.3.7b}$$

□

For details and proofs of this theorem refer to [17, 48]. The theoretical lower bound depends only on the Lagrange multipliers for equality constraints at a local minimum. The bound constraints are excluded from being penalized.

3.4 Different Smoothing of the l_1-Penalty Function

Let us start by considering the exact l_1-penalty function. The corresponding l_1-penalized problem is

$$\Phi(q;\beta) = \mathcal{J}(q) + \beta \sum_{i=1}^{m} |h_i(q)| \quad \text{subject to} \quad 0 \leq q \leq q_c, \tag{3.4.8}$$

where β is a penalty parameter and $q_c = (q_c)_j$ is a given constant.

Among many numerical methods for solving unconstrained optimization problems, gradient based methods have reliable overall performance. But any of the existing gradient methods cannot be directly used to solve (3.4.8). The evident difficulty is the non-differentiability of $\Phi(q,\beta)$. Therefore, by using smooth approximations the l_1-penalized problem $\Phi(q,\beta)$ is transformed into a sequence of minimization problems for smooth approximations of Φ. In the literature, different smooth approximations have been proposed to non–differentiable function $y \to \max\{0, y\}$, see [18, 19, 70, 85]. Here the approach proposed in [94] is used and extended by the introduction of other smooth approximations to l_1-penalty function. For notational simplicity $y = h_j(q)$ has been used in this section.

Based on this idea, corresponding smooth approximations (p_k for $k = \{1, 2, 3, 4\}$) for the absolute value function $y \to |y|$ are listed below.

1. Neural Network smoothing kernel [18]

$$p_1(y;\alpha) = \frac{1}{\alpha}\Big(\ln(1+e^{\alpha y}) + \ln(1+e^{-\alpha y})\Big) = \frac{1}{\alpha}\left(\ln(2) + \ln(1+cosh(\alpha y))\right)$$

   This function has also been used in [94] and is discussed again to obtain a more exhaustive comparison.

2. Chen–Harker–Kanzow–Smale smoothing kernel [20]

$$p_2(y,\alpha) = \frac{1}{\alpha}\sqrt{(\alpha y)^2 + 4}$$

3. Pinar-Zenios smoothing kernel [70, 85]

$$p_3(y,\alpha) = \frac{1}{\alpha}\begin{cases} (-\alpha y - 1/2) & \text{if } \alpha y < -1 \\ (\alpha y)^2/2 & \text{if } |\alpha y| \leq 1 \\ (\alpha y - 1/2) & \text{if } \alpha y > 1 \end{cases}$$

4. Zang smoothing kernel [98]

$$p_4(y,\alpha) = \frac{1}{\alpha}\begin{cases} -\alpha y & \text{if } \alpha y < -1/2 \\ ((\alpha y)^2 + 1/4\alpha) & \text{if } |\alpha y| \leq 1/2 \\ \alpha y & \text{if } \alpha y > 1/2 \end{cases}$$

All functions p_k ($k \in \{1, 2, 3, 4\}$) are strictly convex and approximate the absolute value function based on the value of a smoothing parameter α. The parameter α controls the accuracy of approximation. The maximal approximation errors for the absolute value function are as follows:

$$||y| - p_1(y, \alpha)| \leq \frac{2\ln(2)}{\alpha} \qquad \forall y \tag{3.4.9a}$$

$$||y| - p_2(y, \alpha)| \leq \frac{2}{\alpha} \qquad \forall y \tag{3.4.9b}$$

$$||y| - p_3(y, \alpha)| \leq \frac{1}{2\alpha} \qquad \forall y \tag{3.4.9c}$$

$$||y| - p_4(y, \alpha)| \leq \frac{1}{4\alpha} \qquad \forall y \tag{3.4.9d}$$

One can observe that the $p_4(y)$ approximation produces theoretically the best approximation. However, there has to be a trade-off between the absolute value of α and the exactness property of the smoothed problem. The details on the numerical choice of α will be presented later.

The above listed approximations can be viewed graphically in the following plots. The left hand plots in figures {3.4.1, 3.4.2, 3.4.3, 3.4.4} are the smooth approximations for non-differentiable absolute value function $|y|$ when smooth parameter is chosen as $\alpha = \{10, 100\}$. Also the respective approximation error for smooth parameter chosen as $\alpha = 10$ have been plotted as right hand plots in figures {3.4.1, 3.4.2, 3.4.3, 3.4.4}.

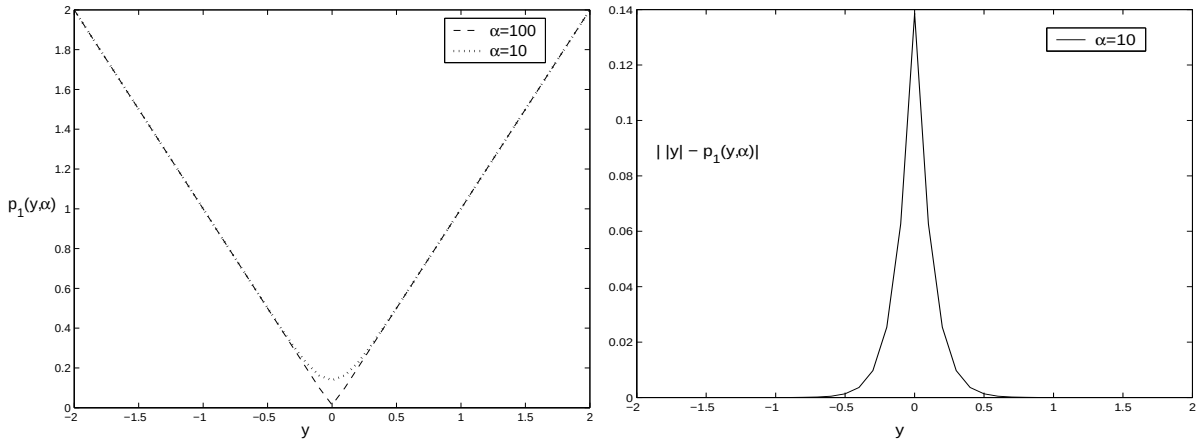

Figure 3.4.1: (Left plot): Neural network smoothing approximation $p_1(y, \alpha)$ for $|y|$ (Right plot): Approximation error for smooth parameter $\alpha = 10$.

Also, the gradient and the Hessian of the penalty terms are bounded:

$$|\nabla p_k(y, \alpha)| \leq 1, \qquad 0 < \nabla^2 p_k(y, \alpha) \leq C_k \alpha, \quad k \in \{1, 2, 3, 4\} \tag{3.4.10}$$

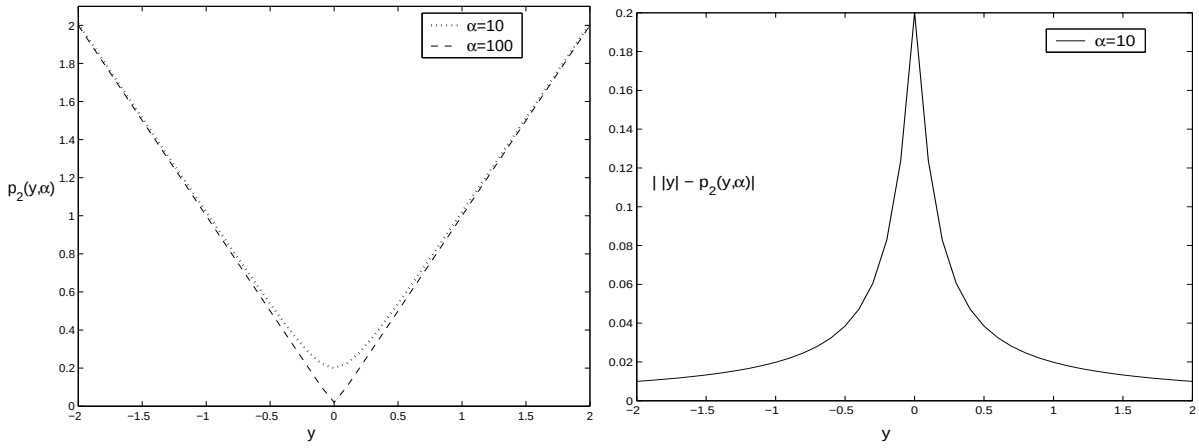

Figure 3.4.2: (Left plot): Chen-Harker-Smale smoothing approximation $p_2(y, \alpha)$ for $|y|$ (Right plot): Approximation error for smooth parameter $\alpha = 10$.

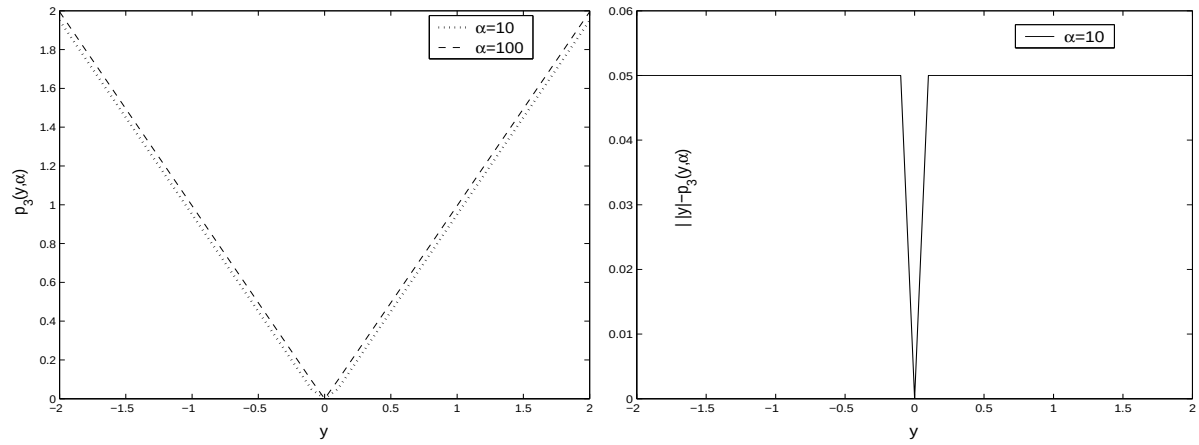

Figure 3.4.3: (Left plot): Pinar Zenios smoothing approximation $p_3(y, \alpha)$ for $|y|$ (Right plot): Approximation error for smooth parameter $\alpha = 10$.

where C_k are constants. For example, using Neural Network smoothing kernel [18] and performing simple calculations one can observe that (3.4.10) holds,

$$p_1'(y;\alpha) = \frac{\sinh(\alpha y)}{\cosh(\alpha y) + 1} \in (-1, 1) \tag{3.4.11a}$$

$$p_1''(y;\alpha) = \frac{\alpha}{\cosh(\alpha y) + 1} \in (0, \alpha/2) \tag{3.4.11b}$$

One should notice that p_3 and p_4 are not two times continuously differentiable globally, but are continuously differentiable near the interesting range $y = 0$.

Using the above defined smooth approximations for $|y|$, the l_1-penalized problem transforms into the corresponding smoothed penalty problem for each fixed $k \in \{1, 2, 3, 4\}$

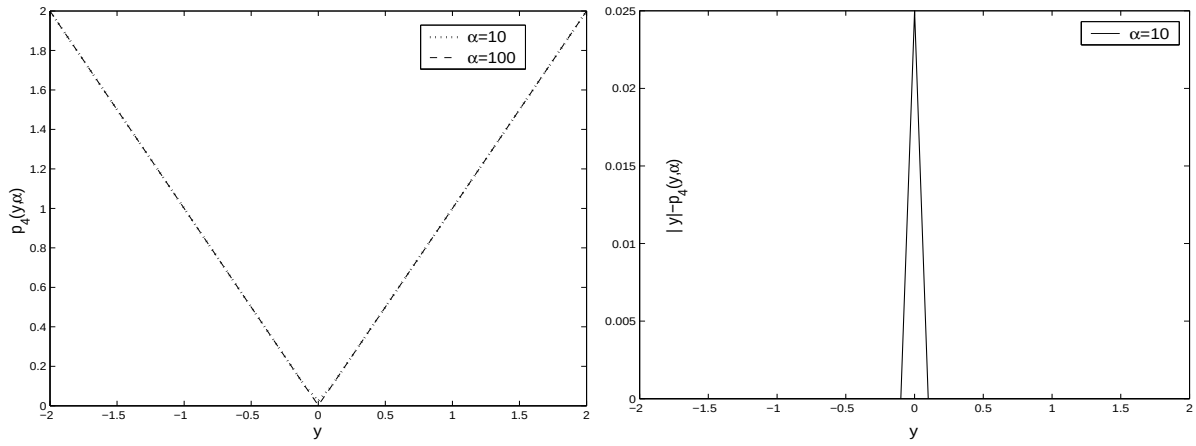

Figure 3.4.4: (Left plot): Zang smoothing approximation $p_4(y, \alpha)$ for $|y|$ (Right plot): Approximation error for smooth parameter $\alpha = 10$.

as,

$$\min_q \Psi(q; \alpha, \beta) := \mathcal{J}(q) + \beta \sum_{j=1}^{m} p_k(h_j(q), \alpha) \quad \text{subject to} \quad 0 \leq q \leq q_c \tag{3.4.12}$$

For example, on choosing the smooth approximation p_1 the smoothed problem is expressed as,

$$\min_q \Psi(q; \beta, \alpha) := \mathcal{J}(q) + \frac{\beta}{\alpha} \left(\sum_{j=1}^{m} \ln\Big(1 + \cosh\left(\alpha h_j(q)\right)\Big) \right) \quad \text{subject to} \quad 0 \leq q \leq q_c$$

Finally, for fixed α, β and smoothing kernel p_k, (3.4.12) is only a bound constrained optimization problem, which can be solved by various efficient bound constrained optimization methods. The ill-conditioning in the Hessian matrix of Ψ will occur for large values of $\beta\alpha$ only if one of the equality constraints is nearly satisfied.

A general framework for solving (3.2.1) based on exact penalization and smoothing approach is given in the following section.

3.5 Adaptive Penalty Algorithm

The nonlinear problem $\min_q \Psi$ defined by (3.4.12) is solved repeatedly with adaptively adjusting the two parameters β and α.

It is well–known that every local minimizer of the *equality constrained* nonlinear problem is also a local minimizer for the associated exact l_1-penalized problem for fixed, *but sufficiently large* penalty parameter β. This result motivates the following algorithm using

different smoothing kernels. In addition, this numerical algorithm combines advantages of exploiting simple constraints and the idea of penalty functions. During the execution of the algorithm, the penalty β and the smoothing α parameters updations differ from the work done in [18, 19, 85].

It will be proved that it is sufficient to increase α whenever the constraint violation $\|h(q)\|_1 \leq c/\alpha$ occurs. On the other hand, if this condition is violated β has to be increased. The generic penalty algorithm is as follows:

Algorithm (3.5.13)

Step 0. Let $k = 0$. Given an initial value q^0, $\beta^0 > 0$ and $\alpha^0 > 0$. Furthermore, fix parameter $\eta_1 \geq 1, \eta_2 >> 1, \eta_3 >> 1$.

Step 1. While $\|h(q^k)\|_1 >$ tol

Step 2. Solve problem (3.4.12) with fixed $\beta = \beta^k, \alpha = \alpha^k$ and obtain q^{k+1}

Step 3. Update α^k or β^k depending on the infeasibility of q^{k+1}

Step 3a. If $\|h(q^{k+1})\| \leq \eta_1/\alpha^k$, then update α^k

$$\begin{aligned} \alpha^{k+1} &= \eta_2 \alpha^k \\ \beta^{k+1} &= \beta^k \end{aligned}$$

Step 3b. Else update β^k

$$\begin{aligned} \alpha^{k+1} &= \alpha^0 \\ \beta^{k+1} &= \eta_3 \beta^k \\ q^{k+1} &= q^k \end{aligned}$$

Step 4. $k = k + 1$ and goto step 1.

In step 2 one can apply any bound constrained optimization method, since the problem (3.4.12) only involves the smooth function Ψ. This algorithm can also be applied to other general bound and equality constrained optimization problems. The details of bound-constrained methods used for step 2 are discussed in the section 3.7.

Another crucial point in the performance of the algorithm is the choice of the initial guess for the penalty parameter β^0. This aspect will be treated in the next section 3.6. The condition used in Step (3a) is based on a theoretical result given below for a strict local minimizer of the equality constrained optimization problem [94].

Proposition 1 *Let the general assumptions of section 3.2 be satisfied and β be sufficiently large. Then every strict local minimizer q^* of (3.4.12) satisfies*

$$\|h(q^*)\|_1 \leq \frac{c}{\alpha} \tag{3.5.16}$$

for a constant c independent of α. □

Proof: Let q^0 be any feasible point of (3.4.12). Due to convexity of the feasible region, on setting $d := q^0 - q^*$ and considering the convex combination $q^* + \lambda d \Rightarrow 0 \leq q^* + \lambda d \leq 1$ for all $\lambda \in [0, 1]$. Further, since $q^* + \lambda d$ is a feasible for (3.4.12) and q^* is a strict local minimizer,

$$\Psi(q^* + \lambda d; \beta, \alpha) > \Psi(q^*; \beta, \alpha), \qquad \forall \quad \lambda \in (0, \lambda_1). \tag{3.5.17}$$

Define $L := \max\{\|\nabla \mathcal{J}(\xi)\| : \xi \in B_{\lambda_1}(q^*)\}$.

Since q^0 and q^* are feasible points for (3.4.12) this holds,

$$\|q^0 - q^*\|_1 = \|d\|_1 \leq n.$$

On using estimates (3.4.9a)-(3.4.9d) and for some large constant c_1, it follows,

$$\Psi(q^* + \lambda d; \beta, \alpha) \leq \Phi(q^* + \lambda d; \beta) + \frac{c_1 \beta}{\alpha}, \tag{3.5.18}$$

$$\Phi(q^*; \beta) \leq \Psi(q^*; \beta, \alpha) + \frac{c_1 \beta}{\alpha}. \tag{3.5.19}$$

Also $\Phi(q^* + \lambda d; \beta)$ can be written as,

$$\begin{aligned} \Phi(q^* + \lambda d; \beta) &= \mathcal{J}(q^* + \lambda d) + \beta \|h(q^* + \lambda d)\|_1, \\ &\leq \mathcal{J}(q^*) + \lambda L \|d\|_1 + \beta(1 - \lambda)\|h(q^*)\|_1, \\ &\leq \Phi(q^*; \beta) - \beta\lambda \|h(q^*)\|_1 + \lambda L n. \end{aligned} \tag{3.5.20}$$

On adding (3.5.18)-(3.5.20) and canceling like terms, it can be concluded as

$$\Psi(q^* + \lambda d; \beta, \alpha) \leq \Psi(q^*; \beta, \alpha) + \frac{2c_1 \beta}{\alpha} - \beta\lambda \|h(q^*)\|_1 + \lambda L n.$$

From (3.5.17) it follows that:

$$\|h(q^*)\|_1 \leq \frac{Ln}{\beta} + \frac{2c_1}{\lambda\alpha}, \qquad \forall \qquad \lambda \in (0, \lambda_1).$$

Hence using assumption that β is sufficiently large, finally one obtains

$$\|h(q^*)\|_1 \leq C/\alpha, \tag{3.5.21}$$

wherein $C := 2c_1/(\min\{1, \lambda_1\})$.

■

3.5.1 Update Rule in case of Equality Constraints

Now the case of equality constraints is considered only , i.e., the problem (3.2.1a) and (3.2.1b). In this particular case β^0 can be chosen, such that $\beta^0 = \beta^k \ \forall k$. Hence, there is no need to update β^k. The following preliminary result is needed.

Proposition 2 *Let (q^*, λ^*) satisfy the KKT conditions (Theorem 3.1) for a local minimizer of problem (3.2.1a) and (3.2.1b).*
Then for $\alpha \to \infty$ there exists a local minimizer $q(\alpha)$ of $\Psi(q; \beta, \alpha)$ and $q(\alpha) \to q^*$ as $\alpha \to \infty$. □

Proof: First, let us consider the exact penalty function Φ. By Theorem 3.3 it is known that for $\beta > \bar{\beta} \geq \|\lambda^*\|_\infty$, q^* is a strict local minimizer of Φ. Using the same arguments as in the proof of Proposition 1, one obtains

$$\Psi(q^*; \beta, \alpha) < \Psi(q; \beta, \alpha) + 16m\beta/(3\alpha) \tag{3.5.22}$$

$q \neq q^* \in B_r(q^*)$, r sufficiently small, $\beta \geq \bar{\beta}$ and α arbitrary. Let us assume now, that there exists a $q \neq q^*$ in a neighborhood of β^* and there exists a $\beta \geq \bar{\beta}$ such that $\Psi(q^*, \beta, \alpha) - \Psi(q, \beta, \alpha) = d(q, \beta, \alpha) > 0$. Then $0 < d < 16m\beta/(3\alpha)$ is obtained, which goes to zero for $\alpha \to \infty$.
■

Now, an estimate for the Lagrange multiplier λ^* independent of the actual minimum q^* is derived.

Proposition 3 *Consider the equality constrained optimization problem $\min_q \mathcal{J}(q)$ subject to $h(q) = H^T q + h^0 = 0$. Assume that q^* is a local minimizer and satisfies the KKT conditions (Theorem 3.1). Denote by λ^* the Lagrange multiplier for $h(q)$.*
Then the following estimate holds for $\lambda^* \in \mathbb{R}^m$:

$$\frac{\|\lambda^*\|_\infty}{\sqrt{m}} \leq \|\lambda^*\|_2 \leq \|H^\# \nabla \mathcal{J}(q^*)\|_2 \leq \|H^\#\|_2 \|\nabla \mathcal{J}(q^*)\|_2 \tag{3.5.23}$$

□

Proof: By assumption the first order optimality conditions are satisfied. Hence, the following is true,

$$\nabla \mathcal{J}(q^*) - H\lambda^* = 0.$$

Since H^T has full rank, $H^T H$ is invertible and one obtains

$$H^T H \lambda^* = H^T \nabla \mathcal{J}(q^*).$$

Using the $\|\cdot\|_2$ norm on both sides of the equation,

$$\|\lambda^*\|_2 \leq \|H^\# \nabla \mathcal{J}(q^*)\|_2$$

Here $H^\#$ denotes the Moore-Penrose pseudoinverse. The next estimate follows from submultiplicativity.
■

If the gradient $\nabla\mathcal{J}$ can be bounded uniformly Proposition 3 gives an a priori estimate for the penalty parameter that is independent of q^* and λ^*. This result will be used later and general estimates for the example problem will be given.

However, the introduced Algorithm (3.5.13) is well-defined due to Theorem 3.4.

3.6 Estimate of Penalty Parameter β^0 for Model Problem

In the following a strip-like network as shown in Figure 3.6.5 is considered. This geometry has the property that there exists only one in- and one outflow arc. Furthermore, the distribution of the flux in the network can be controlled at each node in the top row. The optimal distribution is known for the following set of parameters: All streets have the same maximal flux, $q_{c_j} = 1.0$. The following notation for dimension of vectors and matrices are used $n = n_E$, $m = n_V$ where n_E is the number of edges and n_V i denotes the number of equality constraints $n_V = |V| + 1$. Using the same labeling of the edges as in Figure 3.6.5 and prescribing an inflow of q_0, the optimal solution is [44],

$$q_{opt} = (q^1, \ldots, q^{n_E}) = (q_0, \frac{q_0}{2}, \frac{q_0}{2}, 0, \frac{q_0}{2}, \frac{q_0}{2}, 0, \ldots, \frac{q_0}{2}, \frac{q_0}{2}, 0, \frac{q_0}{2}, \frac{q_0}{2}, q_0).$$

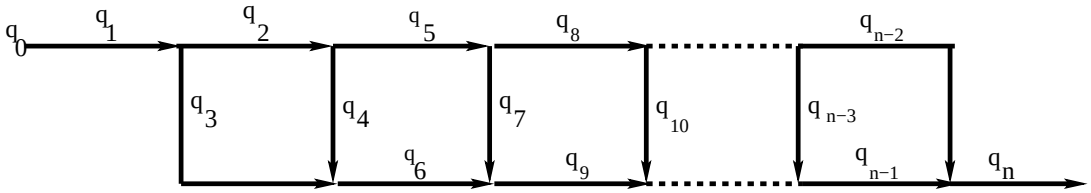

Figure 3.6.5: Geometry of a strip-like network with $n := n_E$ edges

The linear equality constraints (1.5.48) considered are,

$$h(q) = H^T q + h^0 = 0 \tag{3.6.24}$$

where $q \in \mathbb{R}^{n_E}$. For notational convenience lets introduce the integer s which is the number of building blocks for the strip-like network, Therefore, a network with s blocks consists of $n_E = 3s + 4$ edges. For a network with the labeling as in Figure 3.6.5 the following structure of $H \in \mathbb{R}^{n_E \times n_V}$ and $h^0 \in \mathbb{R}^{n_V}$ are obtained. Later, a specific structure of H^T will be used to obtain an estimate on the Lagrange multiplier λ^* as outlined in Proposition 3.

H^T is sparse and given by several blocks where it is distinguished between the inflow edges (first two rows), the central parts (third to $(2s+2)-$th row) and the outflow edges (last row). The sparsity pattern can be seen in figure 3.6.6.

$$H^T_{1:2,1:3} = \begin{pmatrix} +1 & & \\ -1 & +1 & +1 \end{pmatrix} \tag{3.6.25a}$$

$$\forall k \in \{1, \ldots, s\}, i := 2k+1, j := 3k-1 :$$

$$H^T_{i:i+1,j:j+4} = \begin{pmatrix} -1 & & +1 & +1 & \\ & -1 & -1 & & +1 \end{pmatrix} \tag{3.6.25b}$$

$$H^T_{2s+3,3s+2:3s+4} = \begin{pmatrix} -1 & -1 & 1 \end{pmatrix} \tag{3.6.25c}$$

$$h_0 = (-q_0, 0, \ldots, 0)^T \tag{3.6.25d}$$

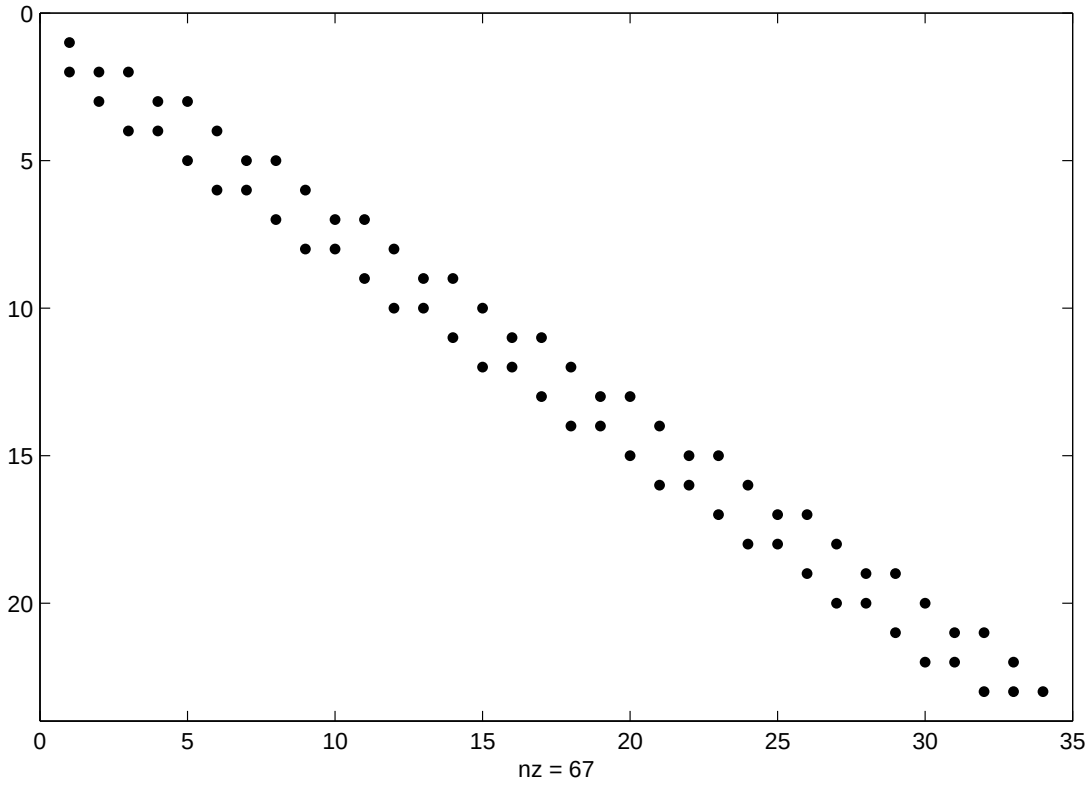

Figure 3.6.6: Sparsity pattern of the matrix H^T of equality constraints. Example given for $s = 10, n_E = 34, n_V = 23$ and $nz = 67$ non-zero elements.

Using the same notation as in section 3.5 and combining the results of Proposition 1 and Theorem 3.3, an a priori lower bound for β^0 is given by

$$\beta^0 = \|\lambda^*\|_\infty \leq \|H^\#\|_2 \| \max_q \|\nabla \mathcal{J}(q)\|_2.$$

Since one would not want to compute the singular values of $H^\#$ an approximation as an initial guess for β^0 can be derived as given below. The approximation is based on the

following estimate:

$$\begin{aligned}\|\lambda^*\|_\infty &\leq \|H^\#\|_2 \| \max_q \|\nabla \mathcal{J}(q)\|_2 \\ &\leq \sqrt{n_V}\|(H^TH)^{-1}\|_2 \|H^T\nabla\mathcal{J}(q^*)\|_\infty \\ &\leq 3\sqrt{n_V}\|\nabla\mathcal{J}(q^*)\|_\infty \|(H^TH)^{-1}\|_2\end{aligned}$$

The matrix $H^TH \in \mathbb{R}^{n_V \times n_V}$ consists of five diagonals only

$$H^TH = \begin{pmatrix} \varrho_1 & \sigma_1 & \gamma_1 & & & & \\ \sigma_1 & \varrho_2 & \sigma_2 & \gamma_2 & & & \\ \gamma_1 & \sigma_2 & \varrho_3 & \sigma_3 & \gamma_3 & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & \ddots & \ddots & \gamma_{n_V-2} \\ & & & \ddots & \ddots & \ddots & \sigma_{n_V-1} \\ & & & & \gamma_{n_V-1} & \sigma_{n_V-1} & \varrho_{n_V} \end{pmatrix}$$

where ϱ_i, σ_i are given by

$$\varrho_i = \begin{cases} 1 & i = 1 \\ 3 & i \neq 1 \end{cases} \tag{3.6.26a}$$

$$\sigma_i = \left\{ \begin{array}{ll} -1 & i = 1, 2 \text{ and } n_V - 2, n_V - 1 \\ -1 & i = 3, 5, 7 \ldots, n_V - 4 \\ 0 & i = 4, 6, 8, \ldots, n_V - 3 \end{array} \right\} \tag{3.6.26b}$$

and where $\gamma_1 = 0, \gamma_i = -1 \;\; \forall i = 2, \ldots, n_V - 2$.

The ith eigenvalue of H^TH is denoted by μ_i. Since

$$\|(H^TH)^{-1}\|_2 = 1/\min\{\mu_i : i = 1, \ldots, n_V\}.$$

Now, one needs to find the minimal eigenvalue $\mu_{\min}$ of H^TH. Numerically, H^TH can be reduced to a symmetric, tri-diagonal matrix and then the bisection algorithm is used to determine the smallest eigenvalue. The reduction can be done by successive Givens rotations. They can be chosen, such that the band structure of the matrix remains, see [79]. In this case at most $(n_V^2 - n_V)/4$ Givens rotations are needed. Even though those calculations can be done before the actual optimization algorithm is executed, the effort is still of order n_V^2.

To derive an analytical estimate on $\mu_{\min}$, two steps of the (truncated) Lanczos algorithm are calculated, see [79]. As initial vector $q^0 = (1, \ldots, 1)^T \in \mathbb{R}^{n_V}$ is chosen. In the first step one obtains $Q_1 \in \mathbb{R}^{n_V \times 1}$ with

$$Q_1^T H^T H Q_1 = \begin{pmatrix} 1/n_V \end{pmatrix}$$

After second step $Q_2 \in \mathbb{R}^{n_V \times 2}$ is obtained as

$$Q_2^T H^T H Q_2 = \begin{pmatrix} 1/n_V & \sqrt{n_V - 1}/n_V \\ \sqrt{n_V - 1}/n_V & \frac{2n_V + (n_V - 1)(3n_V - 1)}{n_V^2 - n_V} \end{pmatrix} =: T$$

The minimal eigenvalue μ_T of T gives an approximation (μ_T) for $\mu_{\min}$:

$$\mu_T = \frac{3n_V - 1 - \sqrt{9n_V^2 - 14n_V + 9}}{2(n_V - 1)} \tag{3.6.27}$$

In Figure 3.6.7 the approximation μ_T and the minimal eigenvalue $\mu_{\min}$ for a matrix $H^T H$ of dimension $n_V \times n_V$ and with the structure (3.6.26) are compared.

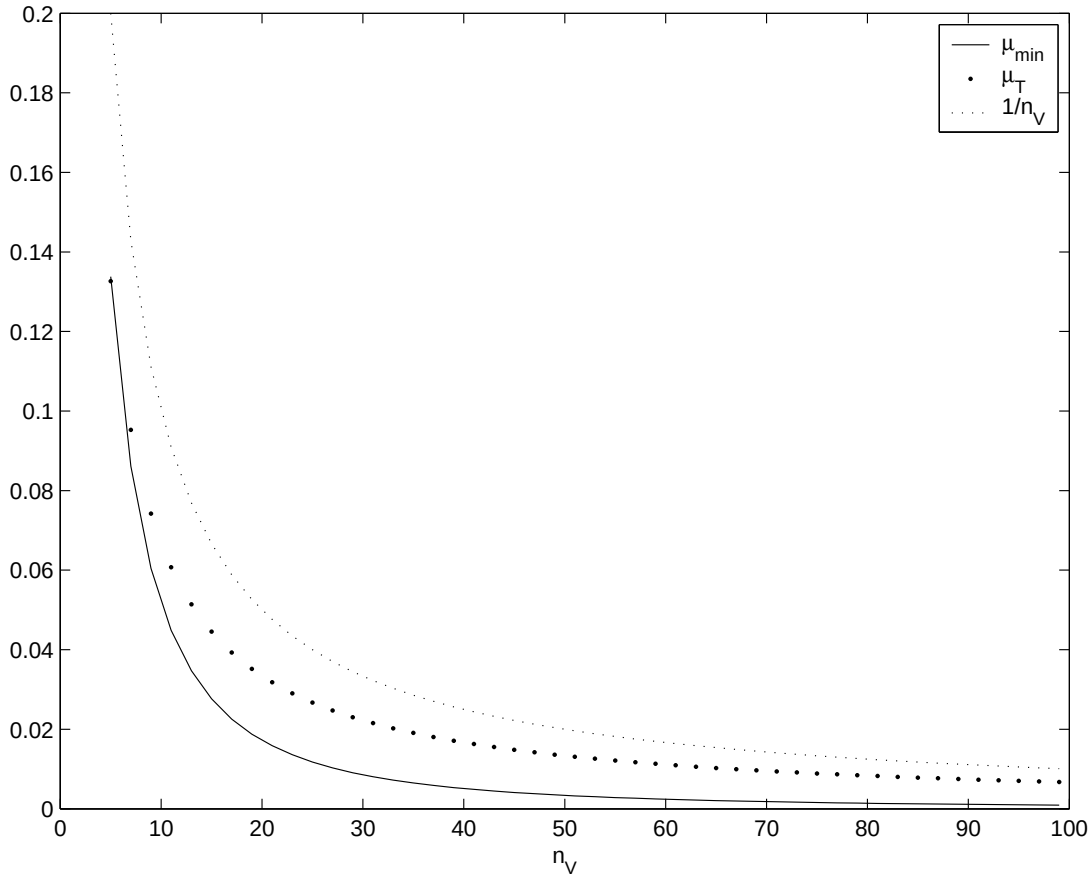

Figure 3.6.7: Comparison of the minimal eigenvalue $\mu_{\min}$ of $H^T H$ and the approximations by the truncated Lanczos method for various dimensions n_V.

Finally, the initial penalty parameter β^0 is chosen as

$$\beta^0 := 3\sqrt{n_V}/\mu_T \max_q \|\nabla \mathcal{J}(q)\|_\infty \tag{3.6.28}$$

Since μ_T is only an approximation to $\mu_{\min}$, (3.6.28) is not an upper bound on $\|\lambda^*\|_\infty$. Hence in the algorithm (3.5.13) the penalty parameter might increase. However, due to the approximation properties of the Lanczos method, (3.6.28) is still a good initial guess for β^0.

When solving problem (3.2.1) where $\mathcal{J}$ and $h(q)$ are given by (2.3.31) and (1.5.48), respectively, the initial β^0 is chosen as (3.6.28) in the algorithm (3.5.13) as in the equality constrained case.

3.7 Solving the Bound Constrained Subproblems

For numerical results two different minimization methods are used in Step 2 in the algorithm (3.5.13). The choice of algorithms is motivated by the problem of traffic flow models for networks mentioned in chapters 1 and 2, which is nonlinear and possibly of large scale.

First, the L-BFGS-B code [101] is adapted to solve (3.4.12). The L-BFGS-B method uses limited memory approximations for the second order information. At each iteration a limited memory BFGS approximation to the Hessian matrix is updated. A search direction is computed using a gradient projection method which identifies set of active variables (as defined in chapter 2. Then a line search is performed along the obtained search directions. More implementation details such as termination criteria and the Hessian matrix updations can be found in [14, 15, 101].

Second, a variant of the truncated Newton-Gradient projection method is used which is based on Lanczos decomposition of the Hessian matrix of the cost functional (developed by R. Felkel and P. Spellucci [30, 94]). This algorithm is called as $PL2$ and do refer to the literature for more details [30, 95]. This bound constrained solver combines projected gradient steps with a truncated Newton method. The bound constraints of the problem are handled with projected gradient steps based on the multiple activation/inactivation strategy in order to identify set of active variables [76]. At each step of Newton's method the linear systems are solved approximately by a truncated Lanczos decomposition of the reduced Hessian matrix of the objective function Ψ. PL2 uses eigenvalue information of the Hessian matrix to evaluate directions of descent. The extremal eigenvalues are computed approximately using Lanczos decomposition based on Ritzvalue approximation. This solver moves in upto three descent directions which includes directions of negative curvature. Using the negative curvature directions local non convexity of the cost function can be exploited. PL2 uses two linesearch algorithms one is an Armijo step size rule which uses independent step sizes for each move and second one is used only if the current direction of descent has negative curvature. This method was developed to solve the large scale problems and it does not rely on matrix factorizations. The details of the Lanczo's method are discussed in many books , e.g., [34, 79].

3.8 Summary

A nonlinear, bound and equality constrained optimization problem related to traffic flow in networks is considered. A primal method for solving nonlinear constrained optimization has been proposed which is based on the exact penalization and different smoothing kernels. In order to solve this numerically an algorithm is introduced and theoretical estimates for its parameters are presented. A best approximation for the threshold value of the penalty parameter is given by (3.6.28). Based on the findings the updation of penalty and smoothing parameter and termination criteria depending on the infeasibility of constraints have been suggested. Approximation properties of different smoothing kernels are reported. Following this idea common numerical optimization methods for bound constrained and unconstrained problems can be used to solve the constrained problems.

Chapter 4

Numerical Results: Simulations and Optimization

This chapter deals with the numerical results which includes model simulations and optimization results for the traffic flow on a sample network. Results of the ODE model (discussed in chapter 1) are compared with the PDE model (LWR model) in order to see the appropriateness of the ODE model. In addition, the penalty algorithm mentioned in chapter 3 is used to solve constrained optimization problem for the RSA model.

4.1 Simulation and Optimization Results for ODE Model

The numerical results presented are simulation of the ODE model described in section 1.3 on a sample network of figure 4.1.6. Later on the optimization results obtained on solving the corresponding optimal control problems (in section 2.1) are stated. If not stated otherwise, the following set of values are used for numerical examples:

$$f_j(\rho_j) = 4\rho_j(1 - \frac{\rho_j}{M_j}),\ M_j = 1.0,\ L_j = 1.0, \sigma_j = M_j/2, T = 5,\ \rho_{j,0} = 0.0,$$

where M_j denotes the maximal density on the road j, i.e., $M_j = \rho_{\max,j}$. The network is assumed to be empty initially, hence $\rho_{j,0} = 0.0$ for each road j in the network. The step–width of the time–discretization τ in the ODE model (1.3.26) and (1.3.28) is set to $\tau = 1/10$.

The PDE–model (1.2.17) is discretized using a first order Godunov–scheme on an equidistant grid with $N_x \times N_t$ grid points. The cost functional $\mathcal{J}$ as defined in (2.1.1), is discretized using a trapezoidal–rule with equidistant spacing of grid points.

4.1.1 Comparison of the ODE–Model (1.3.26) and (1.3.28)

Let us consider the situation at a single junction which can be either of dispersing or merging type, joining three roads 1, 2 and 3 as shown in figure 4.1.1.

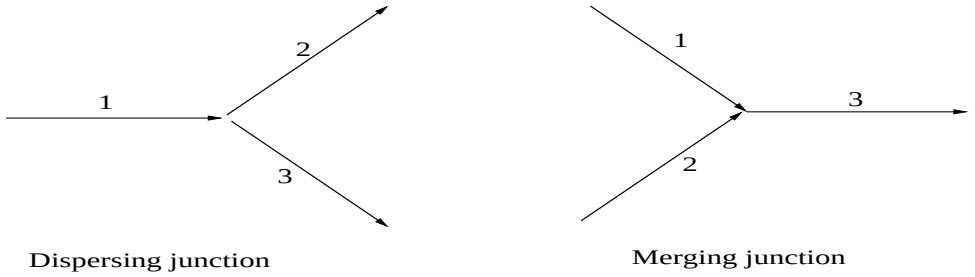

Figure 4.1.1: Two types of junctions: Dispersing and Merging.

In case of a dispersing junction in figure 4.1.1, the flux steering factor is fixed to $\alpha = 0.3$ and a constant inflow density $\rho_1 = 0.3$ is assigned on the incoming road 1 to the junction. The evolution of car-density on each road is plotted in figures 4.1.2 and 4.1.3 where the flow is modelled by the ODE model (1.3.26) and the (modified) ODE model (1.3.28), respectively.

For the merging junction in figure 4.1.1, the inflow density is chosen as $\rho_{1,0} = 0.1$, $\rho_{2,0} = 0.1$ on both incoming roads 1 and 2, respectively. The evolution of car-density is plotted in figures 4.1.4 and 4.1.5 when the flow is governed by the ODE model (1.3.26) and the (modified) ODE model (1.3.28), respectively. In figures 4.1.2 and 4.1.4 an unphysical bump is seen on roads of the dispersing and merging junction for the ODE model (1.3.26). The bump is due to incorrect averaging term at the mid point of roads involved in the ODE model (1.3.26). The bump is not present in figures 4.1.3 and 4.1.5 corresponding to the (modified) ODE model (1.3.28). After the modification of the ODE model (1.3.26) on replacing the averaging term by an appropriate coupling at the mid-point, one observes a smooth transition profile through the junction. This is expected when comparing with the PDE model.

4.1.2 Comparison of the ODE (1.3.28) and PDE (1.2.17) Models

In this section, the PDE model and the (modified) ODE model (1.3.28) are compared on a sample network drawn in figure 4.1.6. To perform this comparison the contour lines for the cost functionals $\mathcal{J}$ and $\mathcal{J}_t$ corresponding to both models as defined by equations (2.1.1) and (2.1.4) respectively are plotted. The considered sample network in figure 4.1.6 has two controls α_1 at the junction $J1$ and α_2 at the junction $J2$. Hence, the objective functional $\mathcal{J}$ and $\mathcal{J}_t$ can be computed for all possible combinations of the two controls (α_1, α_2). This allows to investigate whether the ODE model (1.3.28) has similar properties with respect to the PDE–model (1.2.17). In particular, two different situations are considered: one is a free–flow situation and another is a congested flow i.e., traffic jam situation.

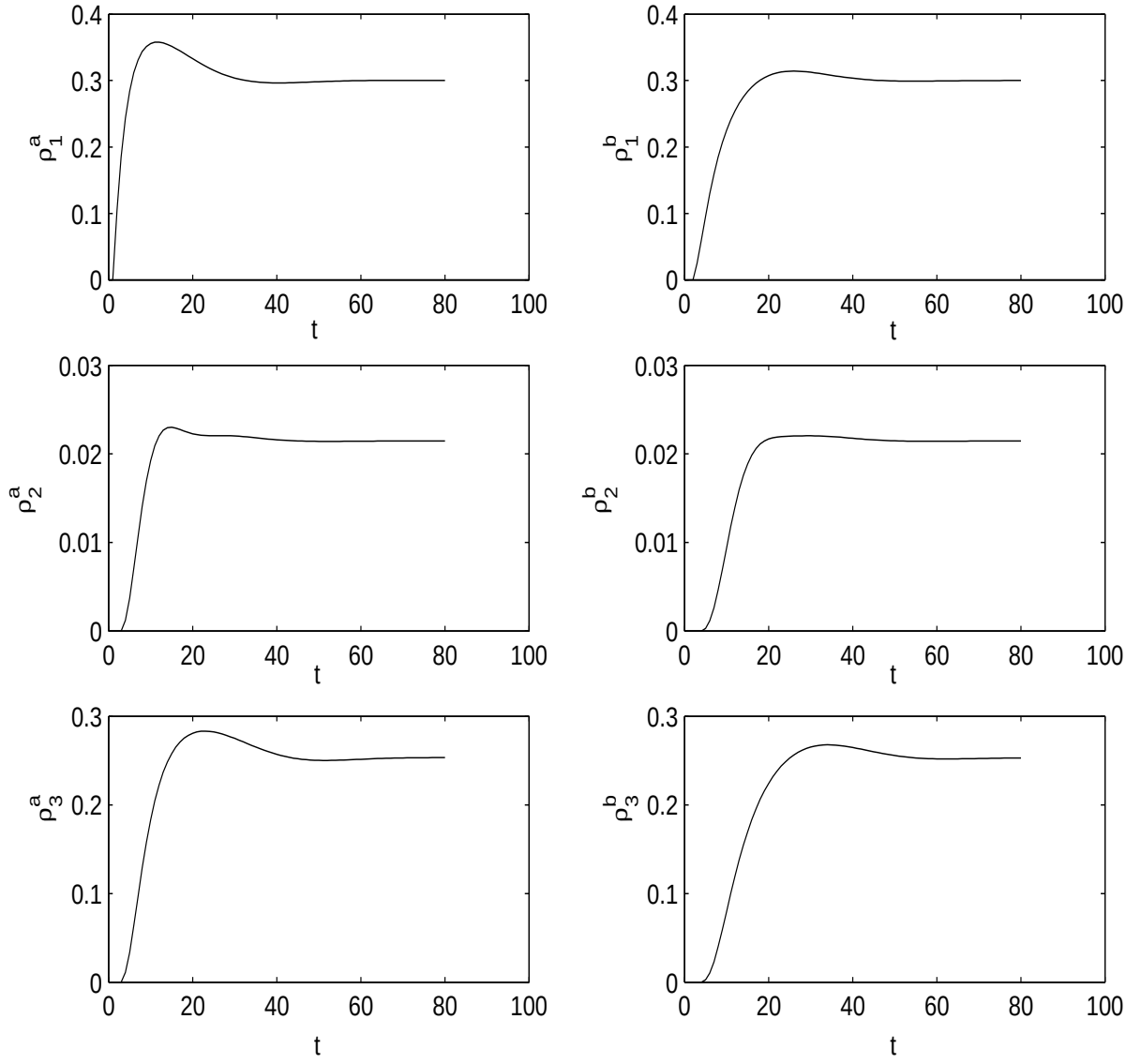

Figure 4.1.2: The density of cars vs. time is plotted on each road joined by a dispersing junction where flow is governed by the ODE model (1.3.26). Incoming density is $\rho_1 = 0.3$ and flux steering factor is $\alpha = 0.3$. ρ_j^a and ρ_j^b denotes spatial approximations of density on each road $j = 1, 2, 3$ and initially $\rho_j^a = \rho_j^b = 0$.

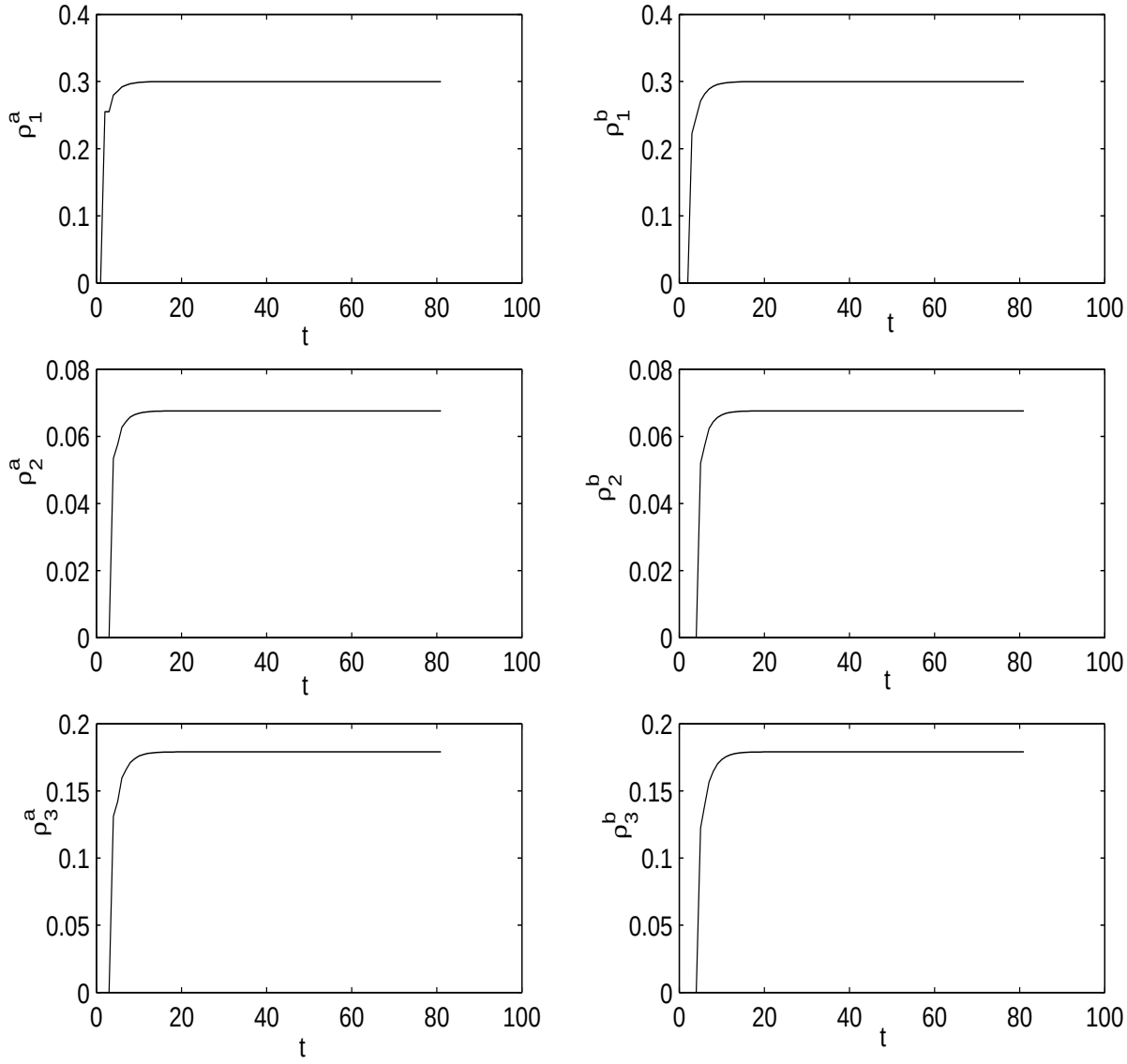

Figure 4.1.3: The density of cars vs. time is plotted on each road joined by a dispersing junction where flow is governed by the ODE Model (1.3.28). Incoming density is $\rho_1 = 0.3$ and and flux steering factor is $\alpha = 0.3$. ρ_j^a and ρ_j^b denotes spatial approximations of density on each road $j = 1, 2, 3$ and initially $\rho_j^a = \rho_j^b = 0$.

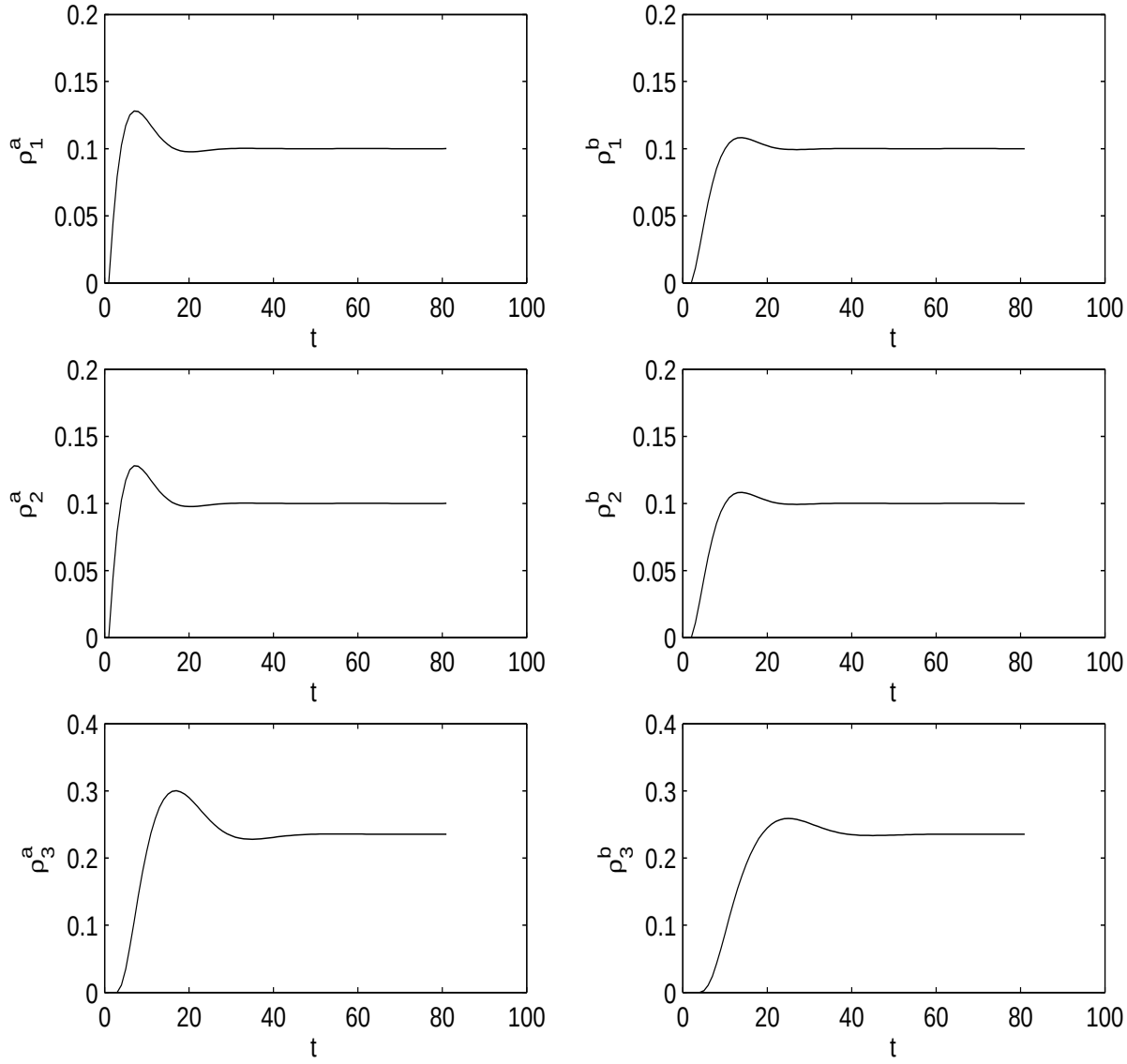

Figure 4.1.4: The density of cars vs. time is plotted on each road joined by a merging junction where flow is governed by the ODE Model (1.3.26). Incoming density is $\rho_1 = \rho_2 = 0.1$. ρ_j^a and ρ_j^b denotes spatial approximations of density on each road $j = 1, 2, 3$ and initially $\rho_j^a = \rho_j^b = 0$.

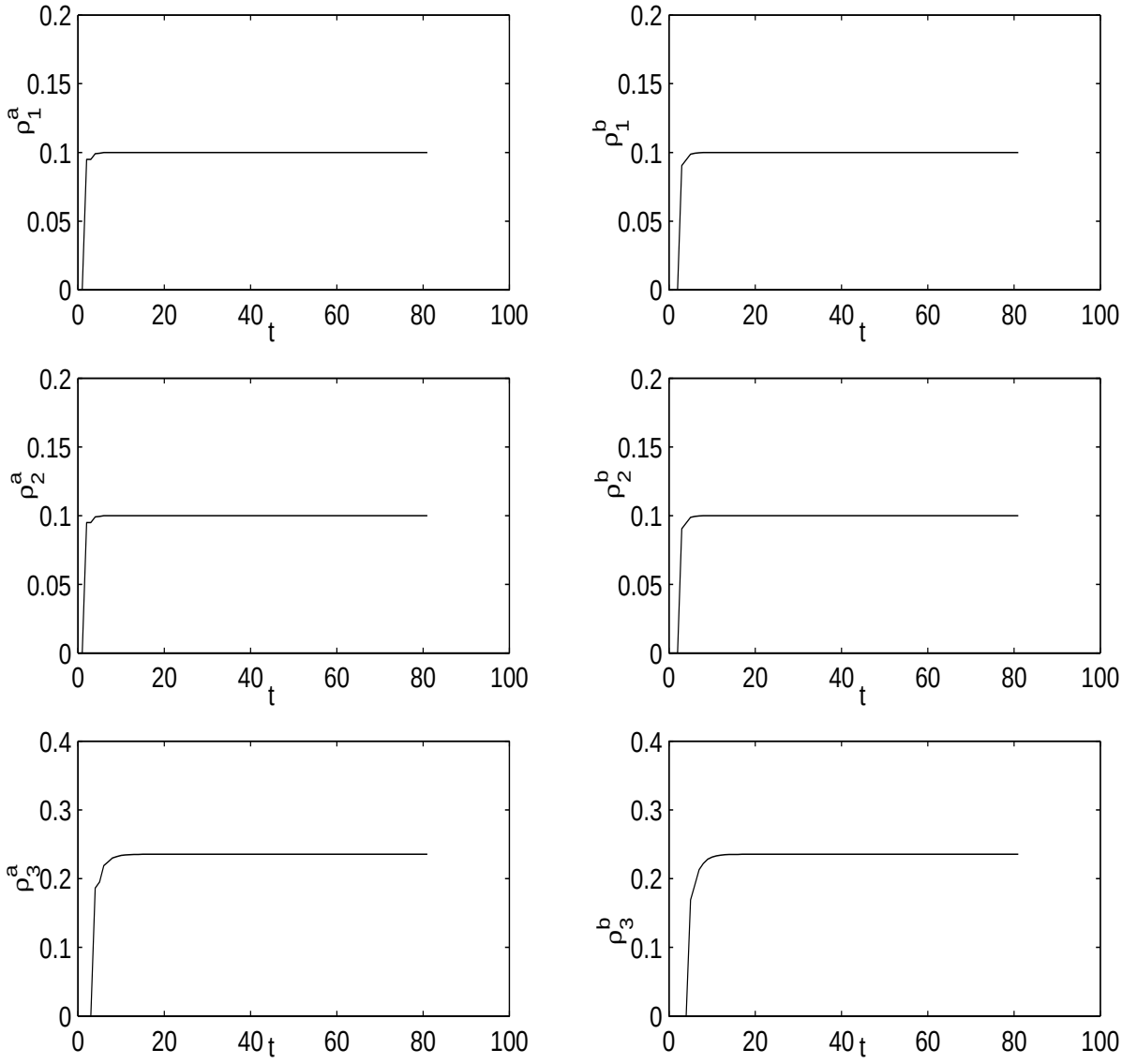

Figure 4.1.5: The density of cars vs. time is plotted on each road joined by a merging junction where flow is governed by the ODE Model (1.3.28). Incoming density is $\rho_1 = \rho_2 = 0.1$ ρ_j^a and ρ_j^b denotes spatial approximations of density on each road $j = 1, 2, 3$ and initially $\rho_j^a = \rho_j^b = 0$.

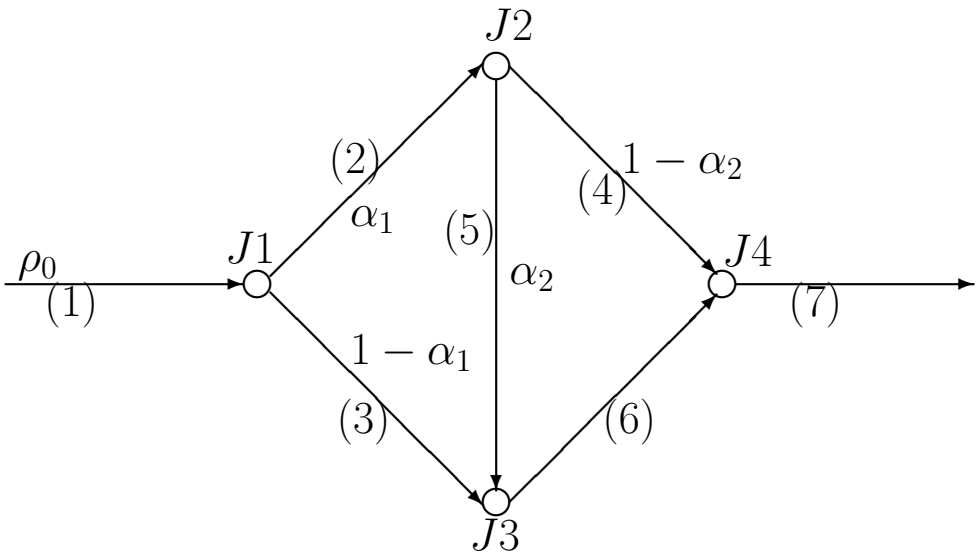

Figure 4.1.6: Sample network of roads where $J1, J2, J3, J4$ denote junctions, (α_1, α_2) are flux steering factors and ρ_0 is the inflow density in the network

In the free–flow situation, the incoming flux on road 1 is given by $f_1(\rho_0) = 0.96$ which is less than the capacities $M_j = 1$ of each road j. Therefore, no traffic jam can occur independent of the applied controls (α_1, α_2). The contour plots of cost functionals are given in figure 4.1.7. A qualitative correspondence is observed for both models. Note that even the optimal controls $(\alpha_1, \alpha_2) = (0.5, 0)$ also coincide in this case.

The second situation of congested network is achieved by varying the maximal densities on each road. There is a reduction of the maximal density at some of the internal roads:

$$M_1 = M_2 = M_4 = M_6 = M_7 = 2, \quad M_3 = 1 \quad \text{and} \quad M_5 = 0.5.$$

The incoming flux in the network is again $f_1(\rho_0) = 0.96$. The contour lines of functionals $\mathcal{J}$ and $\mathcal{J}_t$ are plotted in the figure 4.1.8. The white parts of the plot corresponds to controls (α_1, α_2), where the traffic jam has reached the road number 1 in the figure 4.1.6. These traffic jams appear in both the ODE and the PDE model for $\alpha_1 \leq 0.46$. Additionally, the PDE model simulates those jams for $\alpha_1 > 0.9$ in contrast to $\alpha_1 > 0.95$ for the ODE model. For the remaining controls $0.46 < \alpha_1 \leq 0.95$ a very similar behavior is observed.

For this congested network, shock wave positions are plotted at various times after a jam has occurred at the junction $J2$. The right plot of figure 4.1.9 illustrates the movement of the backward moving shock wave on road 2 because of traffic jam on road 5. Subsequently the backward moving shock wave on road 2 crosses the junction $J1$ and moves backward on the road 1 which can be seen in the left plot of figure 4.1.9.

Remark 4.1 *The ODE model shows similar qualitative properties for the two interesting cases of a free–flow traffic and a congested traffic. In the latter case, the backward moving*

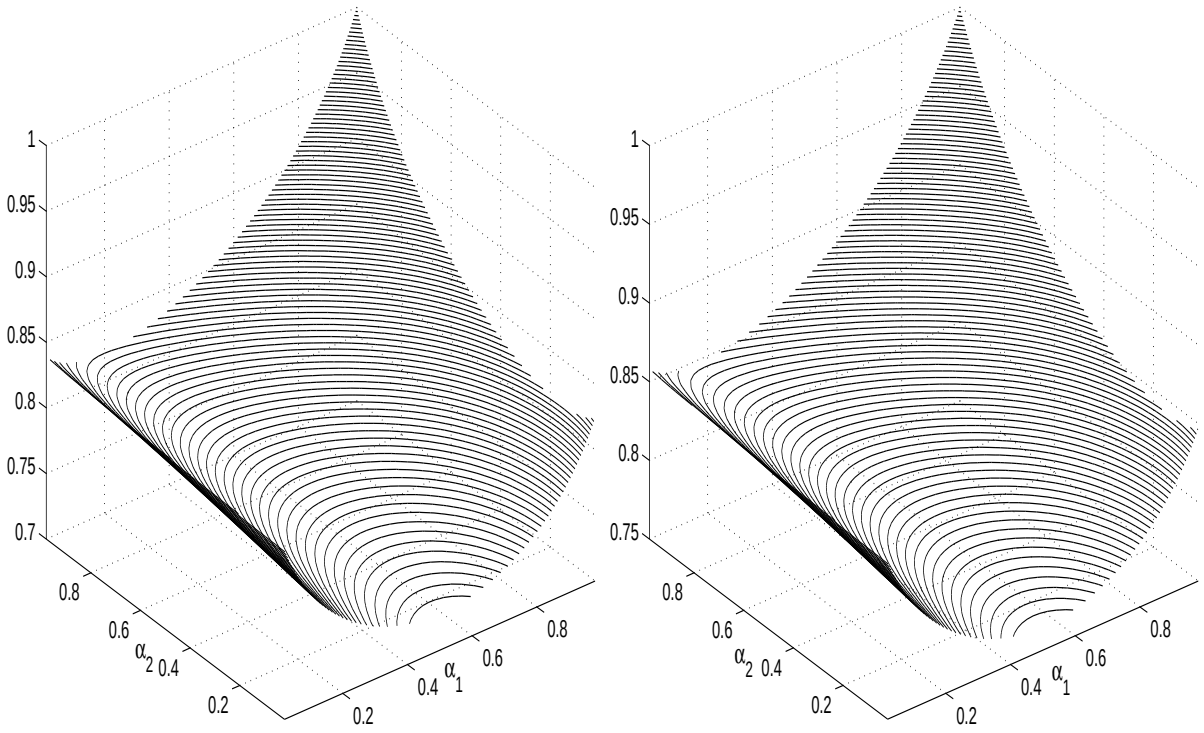

Figure 4.1.7: Contour lines of cost functionals $\mathcal{J}_t$ and $\mathcal{J}$ for ODE (left plot) and PDE (right plot) models for different combinations of (α_1, α_2), respectively.

waves, i.e., traffic jams, which appear in the ODE model, resembles a major feature of the PDE model.

4.1.3 Comparison of Computation time

Besides the qualitative comparison of the ODE model and the PDE model given in the previous section, computation time for the simulation of different size of networks are presented. A scalable network as in figure 4.1.10 with n controls α_i, $i = 1, \ldots, n$ in the top row and a total of $3n - 1$ roads is considered. Let us assume a constant inflow traffic into the network. Values for controls α_i, $i = 1, \cdots, n$ are fixed. The time needed for the simulation of the ODE model (1.3.28) is measured and compared with the time needed for the simulation of the PDE model (1.2.17). The comparison depends upon the number of discretization points used for the Godunov scheme and on the size of the network. In table 4.1.1 computation time is tabulated for the space discretization with $N_x = 100$ gridpoints on each road in PDE model and networks of the sizes 7 to 61 are considered. All computations are performed on a 1.8 Ghz Intel Pentium M using Fortran 77. The computation time for the PDE model is larger by a factor $9 - 30$ than those for the ODE model. The difference in the computation time increases with the size of the network. This is a major drawback of the PDE model when the focus is on the optimization problems, where one needs to solve the PDE model in each iteration of the optimization.

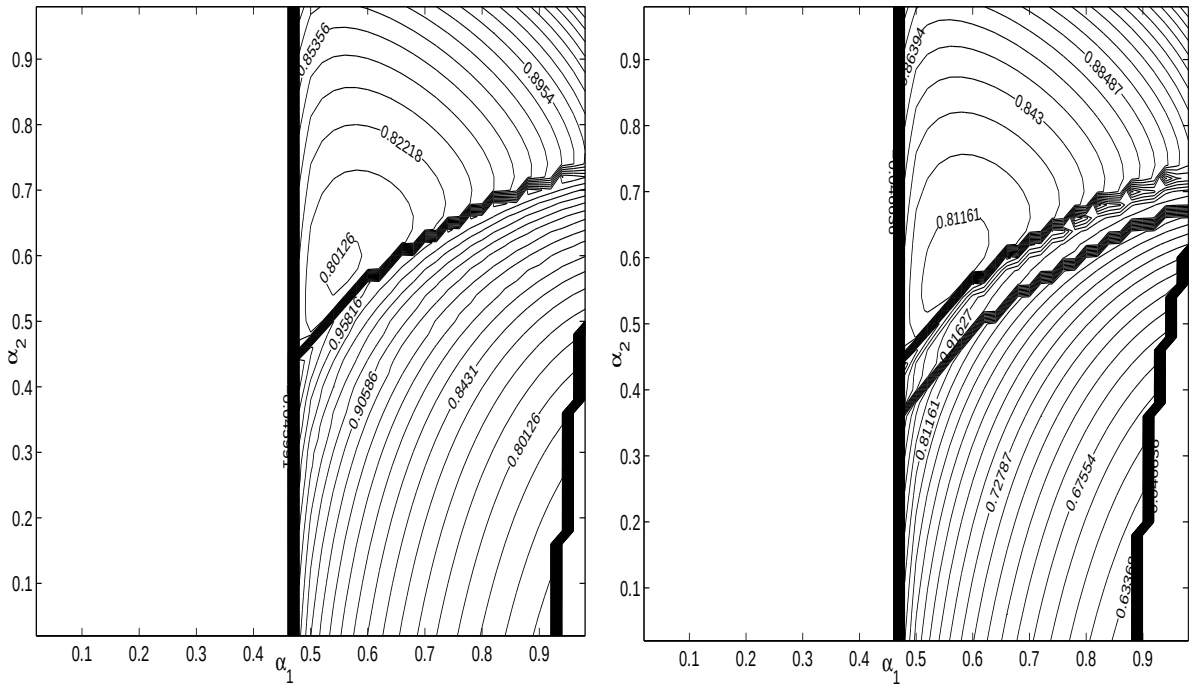

Figure 4.1.8: Contour lines of cost functionals $\mathcal{J}_t$ and $\mathcal{J}$ for ODE (left plot) and PDE (right plot) models with occurrence of congestions for different combinations of (α_1, α_2), respectively.

Table 4.1.1: Computation time of ODE model and PDE model simulations on networks of different sizes.

No. of roads	Computation time ODE model(seconds)	Computation time PDE model(seconds)
7	0.01	0.929
31	0.33	8.98
61	1.07	29.82

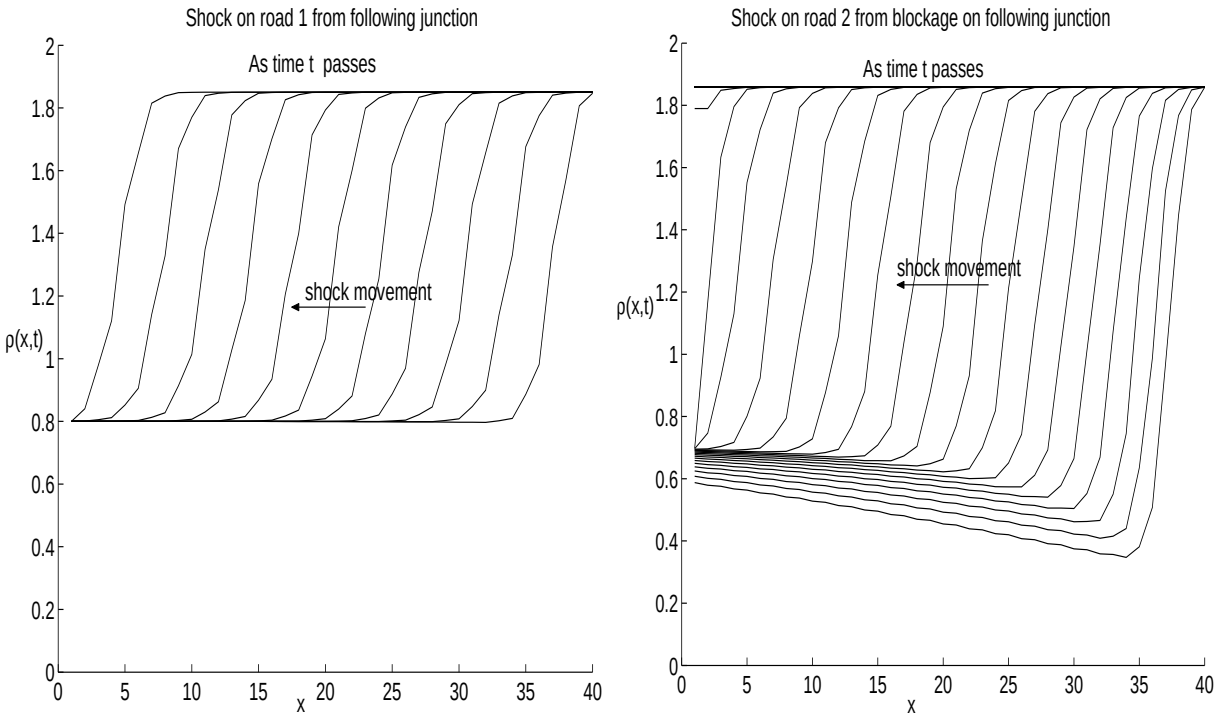

Figure 4.1.9: Movement of backward moving shock wave on road 1 (left plot) and road 2 (right plot) towards inflow of the sample network.

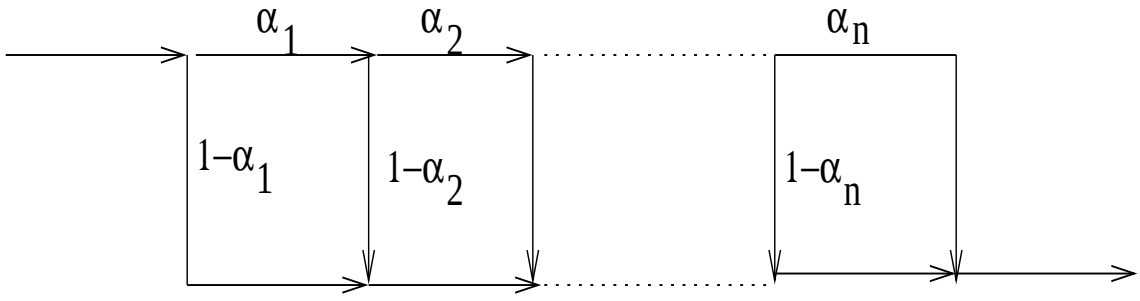

Figure 4.1.10: Layout of a scalable network.

4.1.4 Results of Adjoint Gradient

The network as shown in figure 4.1.6 is considered. The gradient of the cost functional $\mathcal{J}_t$ given by (2.1.4) of (modified) ODE model (1.3.28) can be evaluated in two ways: Either by using the adjoint calculus of section 2.2 or by using a finite difference approximation for $\nabla\mathcal{J}_t$.

$$\nabla\mathcal{J}_t = \frac{\mathcal{J}_t(\alpha + \Delta\alpha) - \mathcal{J}_t(\alpha)}{\Delta\alpha}$$

The components of finite difference approximations of gradient $\nabla\mathcal{J}_t$ with respect to each control α_1 and α_2 are computed using a one–sided difference with $\Delta\alpha_i = 10^{-1}$. On the other hand the gradient $\nabla\mathcal{J}_t$ is computed using adjoint calculus where adjoint and gradient equations are solved as discussed in section 2.2. The absolute difference between the finite difference approximations and adjoint computations of $\nabla\mathcal{J}_t$ are plotted in figure 4.1.11. The gradient differs in order of $O(\Delta\alpha_i)$ and vanishes at the minimum point $(\alpha_1, \alpha_2) = (0.5, 0)$.

Of course, it is advantageous to use the adjoint calculus instead of finite difference approximations for gradients. This is because the adjoint calculus yields all derivatives after a single computation of (2.2.13) and (2.2.14); whereas for finite difference approximations one needs to compute (1.3.28) atleast twice for each control α_i.

4.1.5 Results of Bound Constrained Optimization

The numerical solution of an optimal control problem (2.1.5) is considered. This problem is a nonlinear, bound constrained optimization problem. Using the adjoint calculus of section 2.2 gradients for the cost functional $\nabla\mathcal{J}_t$ are at hand. This allows the application of general optimization methods for bound constrained problems. Different descent methods are considered for solving the optimization problem (2.1.5).

One of the method used for the comparison is the simple steepest descent supplemented with projection onto feasible set in order to incorporate the bound constraints as presented in section 2.4.1 (in chapter 2). Here the step–width is chosen according to the Armijo–rule. The second method considered is the full BFGS method which is a Quasi–Newton method. This method uses second order information by determining approximations to the Hessian of $\mathcal{J}_t$ using the rank-two BFGS update formula [51, 93] discussed briefly in section 2.4. This general BFGS is combined with an active set strategy to determine active variables at bounds as in [10, 21, 51]. It also uses the Armijo step–width rule satisfying the sufficient decrease condition discussed in section 2.4. Especially for large networks it is not possible to store an approximation of the full Hessian. Therefore, the third method used is a limited memory BFGS method. Here, the state-of-art L–BFGS–B code is applied [15, 101]. This code implements a projected gradient method with a limited memory BFGS update and performs well compared with the other optimization methods, see [15] and references therein. Other numerical optimization methods, for example: SQP trust–region [25] methods, augmented Lagrangian methods or penalty methods [9] can

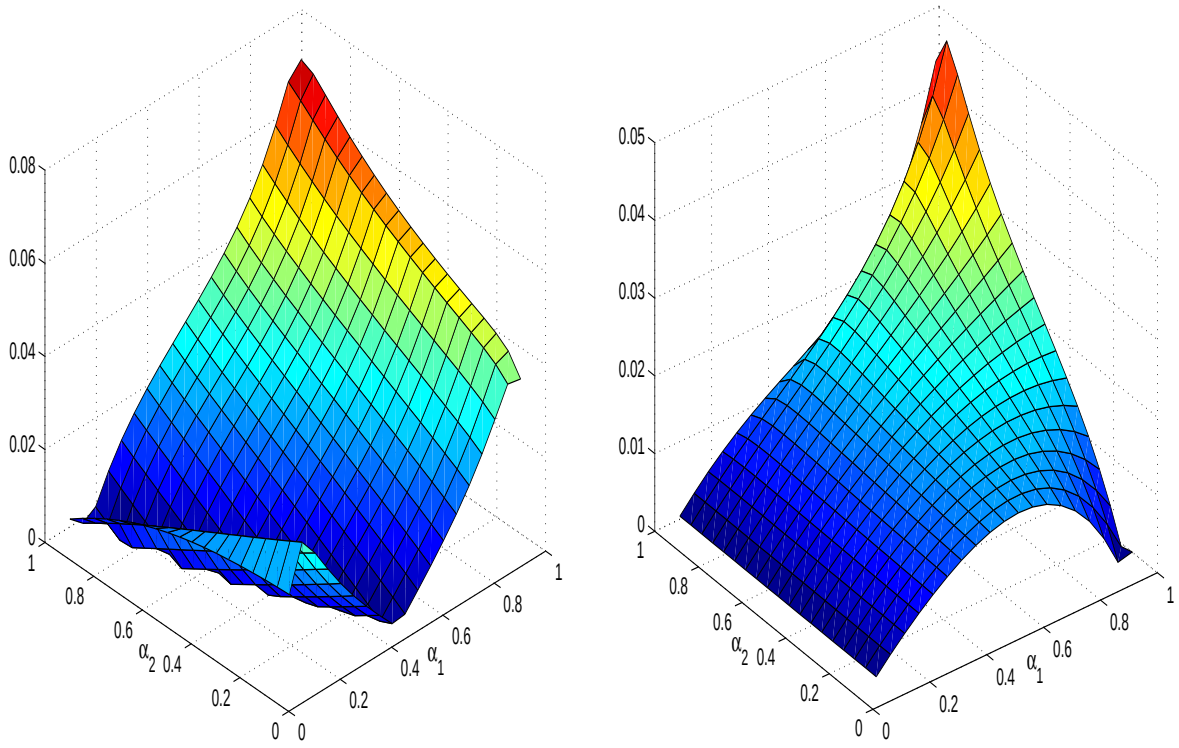

Figure 4.1.11: Absolute difference of gradient $\nabla \mathcal{J}_t$ computed using adjoint calculus and finite difference approximation with respect to α_1 and α_2.

also be applied in this context but are not considered here.

For the comparison of the performance of the above discussed optimization methods, the academic network of figure 4.1.10 is considered. The reason for choosing this network instead of a more realistic one is that the optimal controls α_i are known [44]. For a constant inflow $f_1(\rho_0)$, sufficiently large T and $L_j = b_j - a_j = 1$, the optimal controls are $\alpha = (\frac{1}{2}, 1, \ldots, 1)$, i.e., the flow is such that all roads connecting top and bottom row remain empty. Therefore, one can plot for each iteration step of a numerical optimization method the L^2-norm of the residual. The iteration history of residuals is reported for a network with 20 controls and 61 streets in figure 4.1.12. Linear convergence of projected steepest descent is expected and observed in figure 4.1.12. In case of BGFS methods, due to the lack of exact second–order information of the problem, one cannot expect quadratic convergence results, but superlinear convergence can be expected and is seen figure 4.1.12. The L-BFGS-B method outperforms full BFGS due to a more successful strategy in identifying the active set.

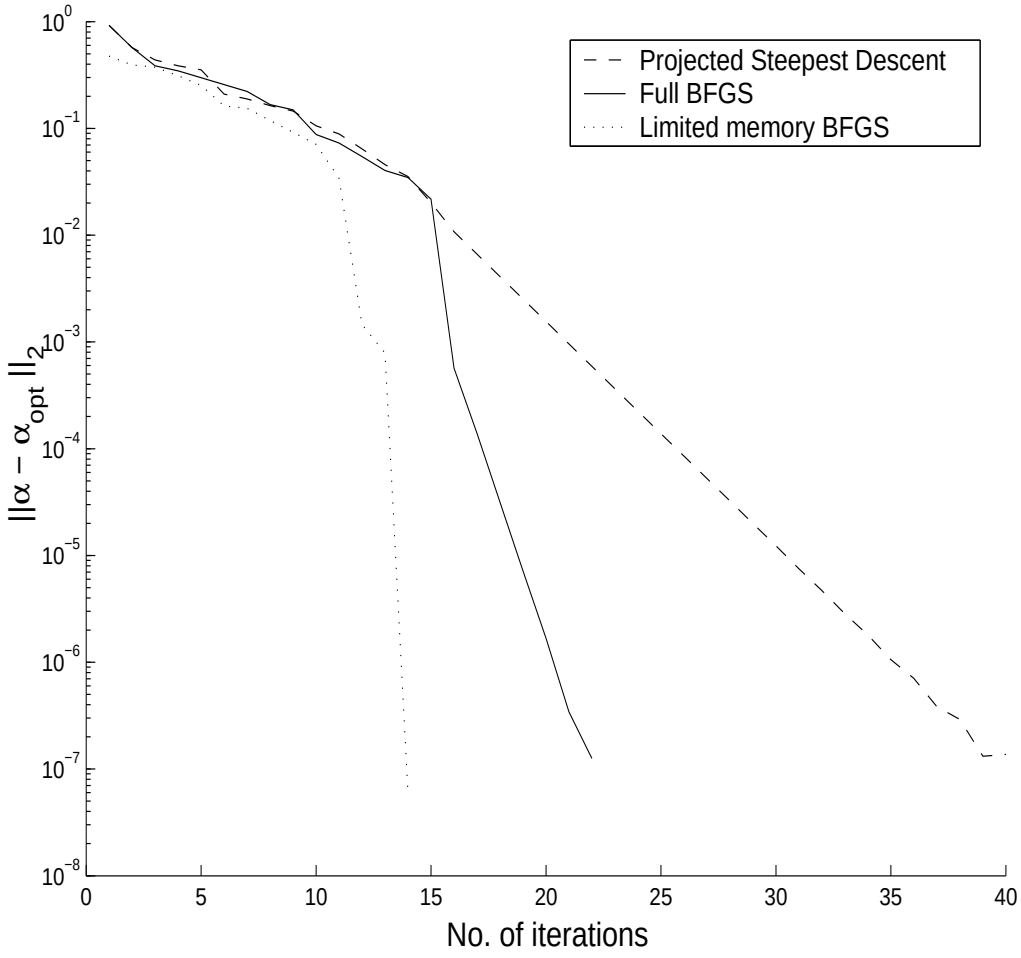

Figure 4.1.12: Convergence result: L^2-norm of the residual of the optimal values of the bound constrained methods vs. number of iterations for a road network with 20 junctions where traffic can be controlled.

4.2 Results of Exact Penalty Methods

In this section the optimization results are presented to optimize the traffic flow through the network shown in figure 3.6.5 governed by the RSA model introduced in section 1.5. The optimization problem to be solved is a nonlinear bound and equality constrained (2.3.32). This constrained optimization problem is transformed into a bound constrained problem (3.4.12) using the concept of exact penalization and smoothing as discussed in chapter 3. The constrained model problem (2.3.32) is solved approximately by solving the bound constrained problem (3.4.12) in the framework of the algorithm (3.5.13) proposed in chapter 3.

The numerical results are reported using two different solvers for the solution of subproblems (3.4.12) in the algorithm (3.5.13). More precisely, numerical results are presented for the extension of bound constrained solvers, L-BFGS-B [15, 101] and PL2

[30, 94]. The test–network is given in the figure 3.6.5 and where roads are labelled as $j \in E := \{1, \ldots, n_E\}$. To compare the performance of the algorithm, networks of different sizes as well as different functions $\tau^j(\cdot)$ appearing in (2.3.31) are considered. The Neural network smoothing kernel p_1 defined in chapter 3 is used otherwise stated.

Parameters of the model problem are: The time horizon for the optimization is $T = 10^3$, bound constraints imposed on fluxes are $0 \leq q_i \leq 1.0$ and an incoming flux q_0 on the road $j = 1$ is $q_0 = 0.75$.

Parameters of the algorithm (3.5.13) are specified as follows. The initial values of parameters are, β^0 is given by (3.6.28), $\alpha^0 = 10$ and $q^0 = \frac{1}{2}(1, \ldots, 1)$. Parameters η_i are set as $\eta_1 = 1$, $\eta_2 = \eta_3 = 10$. The execution of the algorithm terminates if the infeasibility of constraints is less than $tol := 10^{-6}$. In case of bound constrained solvers L-BFGS-B and PL2,

q^{k+1} is accepted as a solution to the subproblem (3.4.12),

if the relative decrease in Ψ is less than 10^{-8}
or
if the l_2-norm of the projected gradient is less than 10^{-10}.

For the remaining parameters the default settings of solvers are used. Results for the problem (2.3.32) with τ^e given by (2.3.29) are presented. To compare the numerically obtained optimal solution the optimal value used is [44],

$$q_{opt} = (q^1, \ldots, q^{n_E}) = (q_0, \frac{q_0}{2}, \frac{q_0}{2}, 0, \frac{q_0}{2}, \frac{q_0}{2}, 0, \ldots, \frac{q_0}{2}, \frac{q_0}{2}, 0, \frac{q_0}{2}, \frac{q_0}{2}, q_0) \tag{4.2.1}$$

4.2.1 Results based on Initial Estimates of Penalty Parameters

For the numerical tests β^0 is set as the estimate in (3.6.28). Figures (4.2.13) and (4.2.14) show the convergence history of the algorithm (3.5.13), where L-BFGS-B and PL2 methods are used to solve the subproblem (3.4.12) for networks of different sizes $n_E \in \{13, 31, 61\}$. The characteristic of residuals is the staircase-like shape of the iteration history. The steep descent in the residual corresponds to an increase of the smoothing parameter after successfully solving (3.4.12). In the history of the L-BFGS-B method in the figure (4.2.13) slight increases of the residual are observed. This increase corresponds to the increase of the penalty parameter and the re-initialization of the smoothing parameter, which can be observed in the table 4.2.2. The PL2 algorithm performs better than the L-BFGS-B method in terms of total number of iterations. One reason for the efficient performance of PL2 over L-BFGS-B is that former uses the exact second order information by evaluating the Hessian matrix, whereas the later uses BFGS approximation to the Hessian. In figure (4.2.14) the fast convergence of algorithm with the PL2 as solver for subproblem inspite of increasing the size of the network is observed.

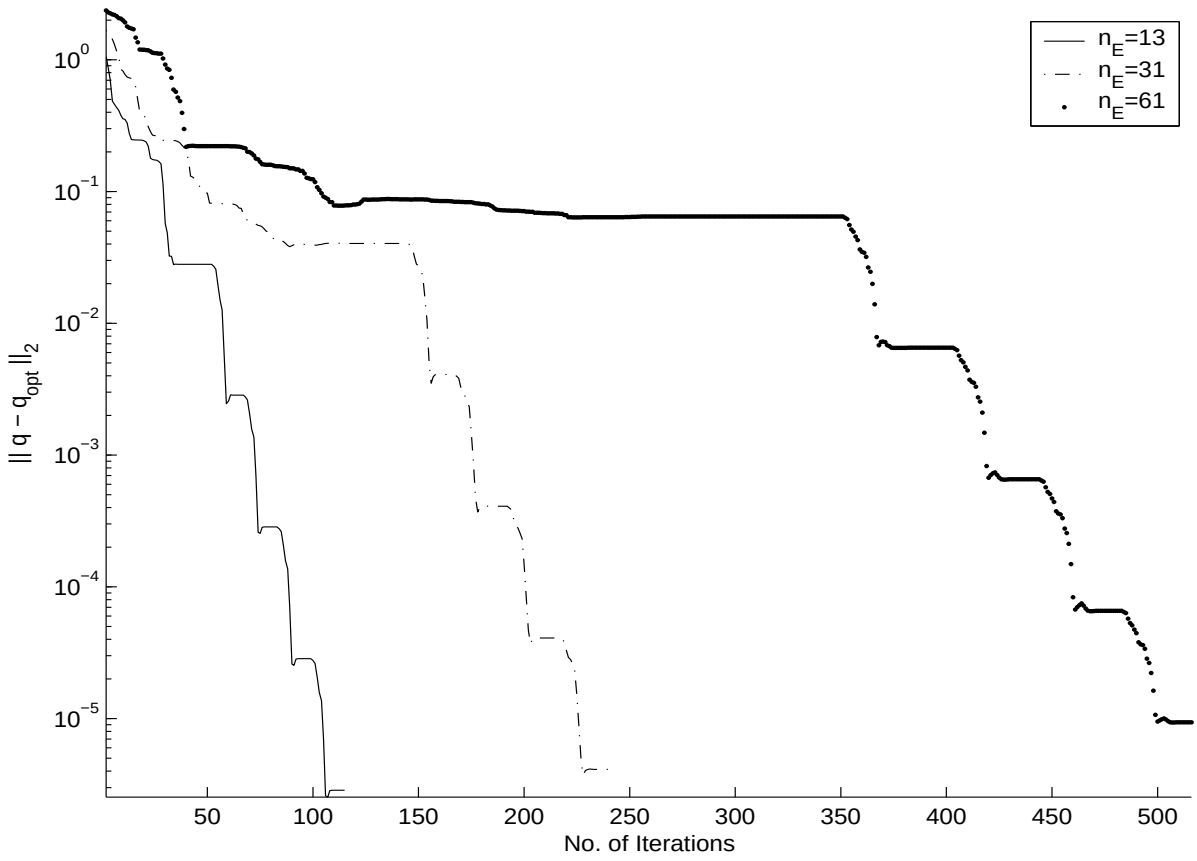

Figure 4.2.13: Iteration history of residuals of optimal values obtained from the execution of algorithm (3.5.13) together with the L-BFGS-B solver for subproblems (3.4.12). The initial value of smoothing parameter α^0 and penalty parameter β^0 are $\alpha^0 = 10$ and β^0 given by (3.6.28) in the algorithm, respectively.

Tables 4.2.2 and 4.2.3 list the current smoothing parameter, the penalty parameter and the number of iterations to solve the subproblem in step 2 of algorithm (3.5.13). One can observe that the penalty parameter remains unchanged and only the smoothing parameter is updated when PL2 is used as a subproblem solver. The computation time for the algorithm with PL2 as solver are tabulated for mid-size networks in the table 4.2.4.

In the figure 4.2.15 the convergence of the cost functional $\mathcal{J}$ in (2.3.32) of original problem and the cost functional Ψ of smoothed problem in (3.4.12) to their optimal values are plotted. It is seen that both cost functionals converge to the identical value.

4.2.2 Arbitrary Choice of Penalty Parameters

In order to study the performance of the algorithm for arbitrary initial choice of the penalty parameter, a second test is performed. Parameters are initially set as $\alpha^0 = \beta^0 = 10$ in algorithm (3.5.13) on a network of size $n_E = 13$. The corresponding residuals with

Table 4.2.2: Smoothing parameter α^k, penalty parameter β^k and number of iterations necessary to solve the subproblem (3.4.12) for algorithm (3.5.13) with L-BFGS-B as solver. The dimension of the problem is $n_E = 61$,$\alpha^0 = 10$ and β^0 given by (3.6.28).

α^k	β^k	Iterations
10	1166604.8163	360
100	1166604.8163	48
1000	1166604.8163	1
10	11666048.163	2
100	11666048.163	37
1000	11666048.163	1
10000	11666048.163	1
10	116660481.63	1
100	116660481.63	36
1000	116660481.63	1
10000	116660481.63	1
100000	116660481.63	1
10	1166604816.3	1
100	1166604816.3	30
1000	1166604816.3	1
10000	1166604816.3	1
100000	1166604816.3	1

Table 4.2.3: Smoothing parameter α^k, penalty parameter β^k and number of iterations necessary to solve the subproblem (3.4.12) for algorithm (3.5.13) with PL2 as solver. The dimension of the problem is $n_E = 61$, $\alpha^0 = 10$ and β^0 given by (3.6.28).

α^k	β^k	Iterations
10	1166604.8163	51
100	1166604.8163	33
1000	1166604.8163	33
10000	1166604.8163	33
100000	1166604.8163	33

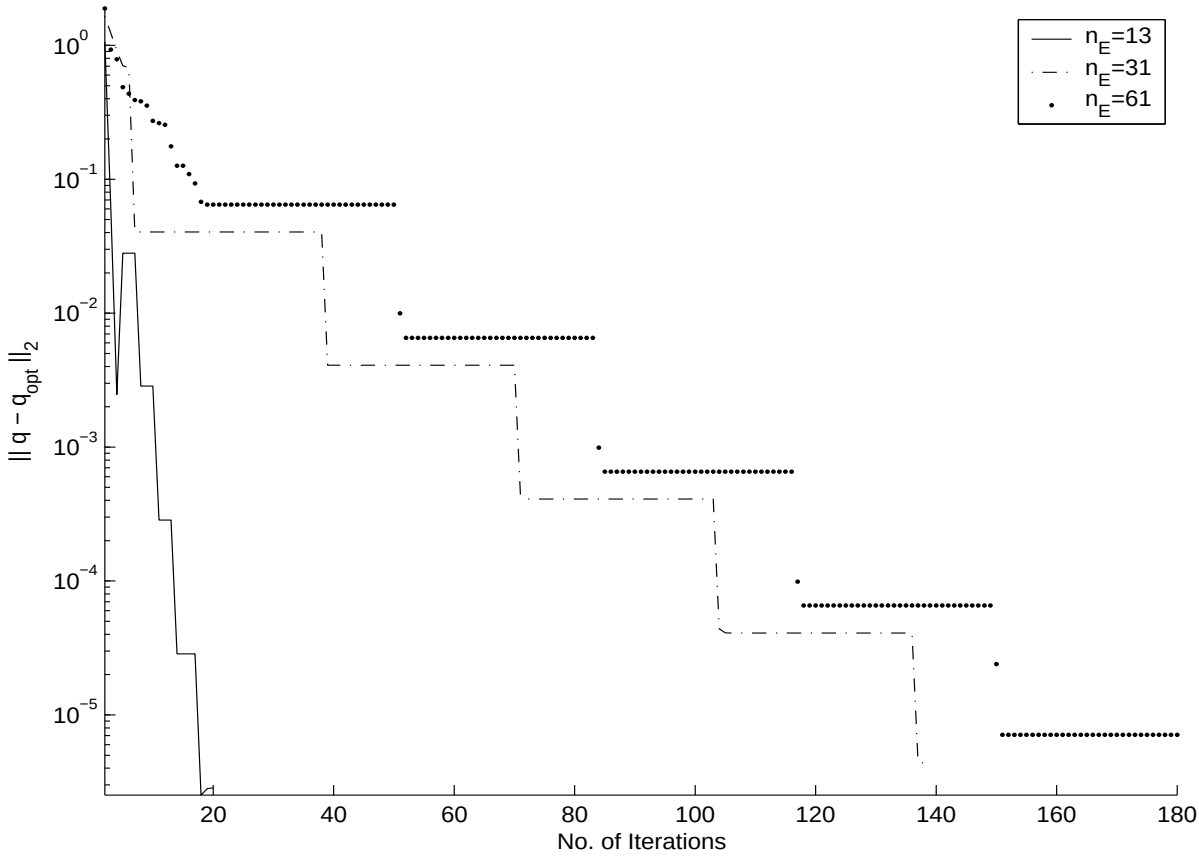

Figure 4.2.14: Iteration history of residuals of optimal values obtained from the execution of algorithm (3.5.13) together with the PL2 solver for subproblem (3.4.12). The initial value of smoothing parameter α^0 and penalty parameter β^0 are $\alpha^0 = 10$ and β^0 given by (3.6.28) in the algorithm, respectively.

respect to L-BFGS-B and PL2 as a solver for the bound constrained subproblem are plotted in the figure 4.2.16. As expected independent of the solver for the subproblem an increase in the value of penalty parameter is observed, see tables 4.2.5 and 4.2.6. The peaks in the residual correspond to an increase of the penalty parameter, re-initialization of the smoothing parameter and the restart in step 3b of the algorithm (3.5.13).

The tables 4.2.5 and 4.2.6 list the current smoothing parameter α^k, the current penalty parameter β^k and the number of iterations necessary to solve the subproblem (3.4.12) in step 2 of algorithm (3.5.13). An iteration count of zero for the solution of the subproblem implies that the previous iterate q^{k-1} is also a solution for the new problem with increased smoothing or penalty parameter.

As observed in the previous test, one can note that the PL2 generates solutions closer to the exact minimum which leads to less number of updations of β^k compared to L-BFGS-B. The total number of iterations (iterations per subproblem $\times$ number of solved subprob-

Table 4.2.4: Computation time on mid size networks for solving the constrained problem using the algorithm (3.5.13) together with PL2 as solver for the subproblem.

Number of Roads	Computation time
46	0.15 seconds
151	7.7 seconds
301	11.3 seconds
1501	11.8 minutes

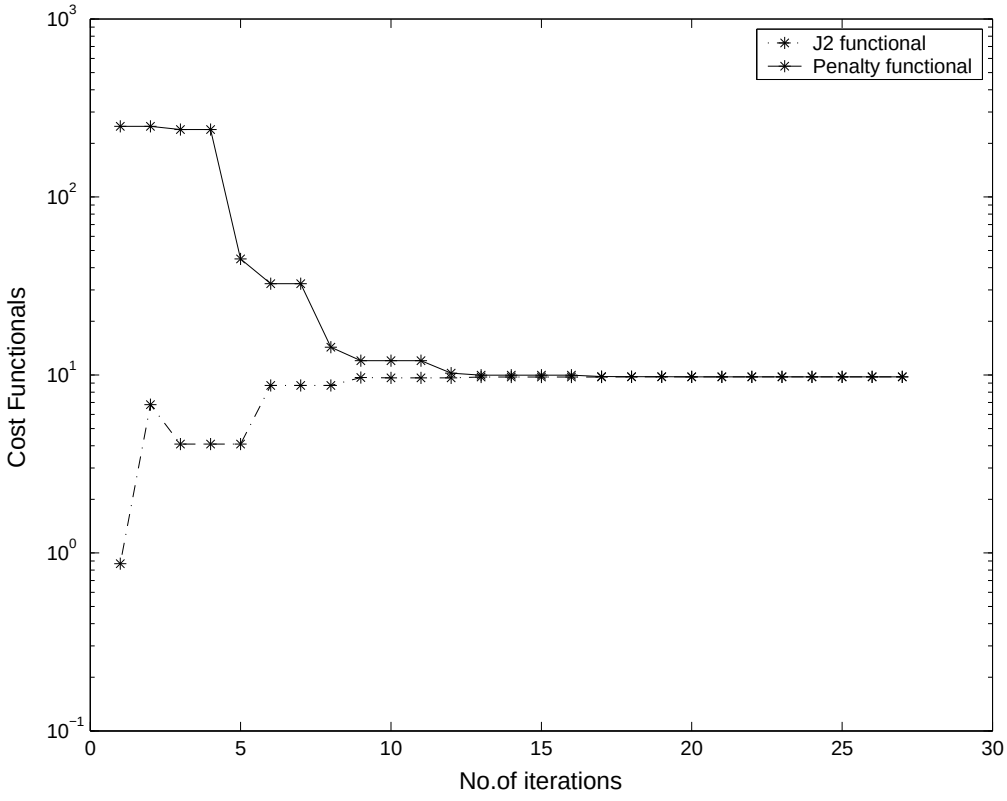

Figure 4.2.15: Convergence history of cost functionals of the original constrained problem $\mathcal{J}$ and approximated bound constrained problem Ψ to the optimal value.

Table 4.2.5: Smoothing parameter α^k, penalty parameter β^k and number of iterations necessary to solve the subproblem (3.4.12) for algorithm (3.5.13) with L-BFGS-B as solver. The dimension of the problem is $n_E = 13$ and $\alpha^0 = \beta^0 = 10$.

α^k	β^k	Iterations
10	10	1
10	100	1
10	1000	22
10	10000	42
10	100000	52
100	100000	13
1000	100000	1
10	1000000	4
100	1000000	13
1000	1000000	1
10000	1000000	1
10	10000000	0
100	10000000	12
1000	10000000	1
10000	10000000	1
100000	10000000	1
10	100000000	0
100	100000000	11
1000	100000000	1
10000	100000000	1
100000	100000000	1

Table 4.2.6: Smoothing parameter α^k, penalty parameter β^k and number of iterations necessary to solve the subproblem (3.4.12) for algorithm (3.5.13) with PL2 as solver. The dimension of the problem is $n_E = 13$ and $\alpha^0 = \beta^0 = 10$.

α^k	β^k	Iterations
10	10	2
10	100	3
10	1000	21
10	10000	26
10	100000	28
100	100000	24
1000	100000	17
10000	100000	25
100000	100000	22

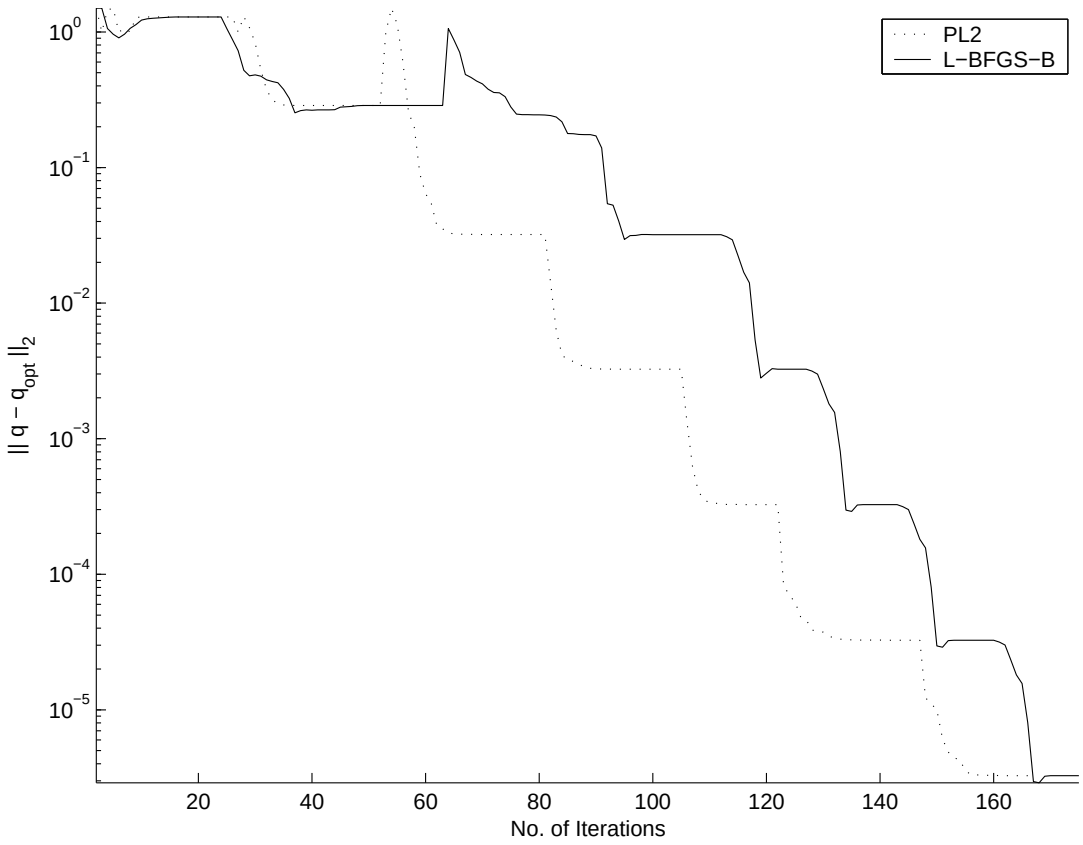

Figure 4.2.16: Comparison of iteration history of algorithm (3.5.13) combined with PL2 and L-BFGS-B respectively. Initial values are $\alpha^0 = \beta^0 = 10$ and size of the problem is $n_E = 13$.

lems) is comparable for both solvers, i.e., 168 (PL2) and 173 (L-BFGS-B). The method L-BFGS-B as subproblem solver needs half of the iterations per subproblem compared to the PL2. On the other hand, the generated approximations q^k using L-BFGS-B have larger residual and this leads to an additional increase in the penalty parameter and more number of subproblems need to be solved. It can be inferred here that the PL2 performs better even for arbitrary choice of initial penalty parameter.

Now, results are presented on solving the unconstrained subproblems (3.4.12) with the function τ being set as $\tau(q) = q$, in (2.3.31). The initial values of parameters are chosen as $\beta^0 = \alpha^0 = 10$ and $n_E = 13$. All other parameters are fixed as defined above in the section 4.2. In the figure (4.2.17) the residual and the iteration history of unconstrained optimization problem are plotted. Similar to observations from the figure (4.2.14), we observe that the PL2 method generates approximations q^k closer to the optimal solution. This leads to less number of updations of the penalty parameter (Step 3b), but additional iterations for the solution of the subproblem in Step 2 are needed. The comparison is tabulated in the table (4.2.7).

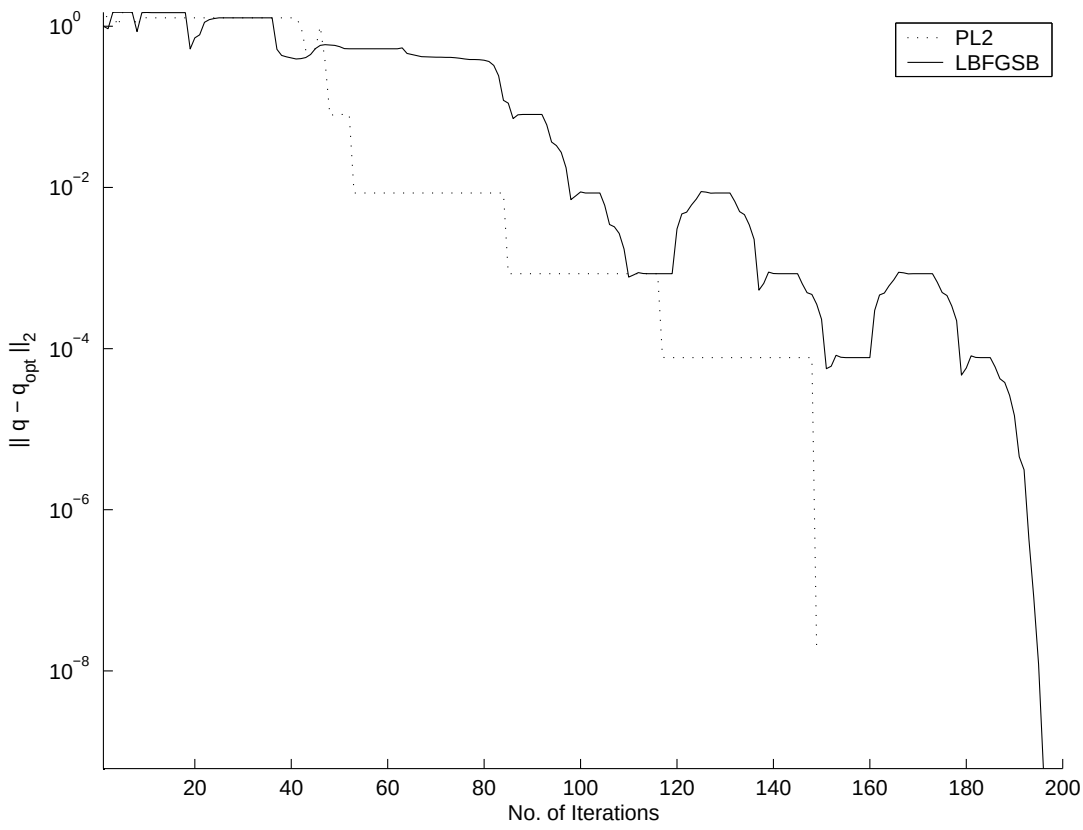

Figure 4.2.17: Iteration history of algorithm (3.5.13) for PL2 and L-BFGS-B respectively. for unconstrained subproblems, $\tau(q) = q$ and $n_E = 13$.

4.2.3 Different Smoothing of the Exact l_1-Penalty Function

The behavior of different smoothing approximations defined in section 3.4 is investigated. Results are presented corresponding to all four smoothing kernels, i.e., for all $p_k, k = 1, 2, 3, 4$. The smoothing kernel is not changed during the execution of the algorithm. The same is true for values of parameters of the algorithm (3.5.13). Now parameters η_i are given by $\eta_1 = 1, \eta_2 = 5$ and $\eta_3 = 10$, respectively. The other settings are kept same as in section 4.2. The initial smoothing parameter is $\alpha^0 = 10$ and the choice of the initial penalty parameter is discussed below. In step 2 of the algorithm (3.5.13) the PL2 is applied as a subproblem solver.

To illustrate the difference in the overall performance of the algorithm (3.5.13) results are persented for two choices of the initial penalty parameter:

1. β^0 is set to $\beta^0 = 10$

2. The estimate for β^0 given by (3.6.28) is used.

The numerical results are as follows:

Table 4.2.7: Smoothing and penalty parameters and number of iterations necessary to solve $\min \Psi(q;\alpha,\beta)$ for algorithm (3.5.13) combined with PL2 (left table) and L-BFGS-B (right table). Size of the problem is $n_E = 13$ and $\tau(q) = q$.

α^k	β^k	Iterations
10	10	3
10	100	4
10	1000	34
10	10000	4
10	100000	6
100	100000	32
1000	100000	32
10000	100000	32
100000	100000	4

α^k	β^k	Iterations
10	10	6
10	100	11
10	1000	18
10	10000	26
10	100000	30
100	100000	12
1000	100000	14
10000	100000	1
10	1000000	12
100	1000000	14
1000	1000000	13
10000	1000000	1
100000	1000000	1
10	10000000	13
100	10000000	12
1000	10000000	12
10000	10000000	1
100000	10000000	1

Example 1:

A network with $n_E = 31$ roads is considered and $\beta^0 = 10$. Tables 4.2.8 and 4.2.9 are presented corresponding to the smoothing kernels p_1, p_2 and p_3, p_4, respectively. In each table the current smoothing parameter α^k, the current penalty parameter β^k and the number of iterations needed to solve the subproblem (3.4.12) in Step 2 of algorithm (3.5.13) are listed. These tables show that the number of update in steps 3a and 3b respectively, depend on the choice of smoothing kernel.

Example 2:

For fixed $n_E = 31$ and fixed smoothing kernel p_k the convergence history of algorithm (3.5.13) is presented. Residuals versus number of iterations are plotted for different smoothing kernels in the same figure 4.2.18 in order to perform easier comparison. The top part of figure 4.2.18 corresponds to the tables given in Example 1, i.e., using $\beta^0 = 10$ and in the bottom plot β^0 is given by (3.6.28).

As before a step like shape of the optimization history for any smoothing kernel and initial penalty parameter is observed. Each step downwards is related to an update of the smoothing parameter α^k (step 3a). The sharp peaks in the residual plots are related

Table 4.2.8: Example 1: (Left table): Using Neural Network smoothing kernel p_1 and (Right table): Chen-Harker-Kanzow-Smale smoothing kernel p_2.

α^k	β^k	Iterations
10	10	2
10	100	3
10	1000	6
10	10000	37
10	100000	38
50	100000	33
250	100000	33
1250	100000	33
6250	100000	33
31250	100000	32
156250	100000	33
781250	100000	3
10	1000000	32
50	1000000	4
250	1000000	32
1250	1000000	32
6250	1000000	32
31250	1000000	33
156250	1000000	5
781250	1000000	4

α^k	β^k	Iterations
10	10	2
10	100	3
10	1000	35
10	10000	39
10	100000	41
50	100000	35
250	100000	35
1250	100000	35
6250	100000	34
31250	100000	34
156250	100000	34
781250	100000	33
10	1000000	33
50	1000000	34
250	1000000	34
1250	1000000	34
6250	1000000	33
31250	1000000	34
156250	1000000	33
781250	1000000	4

to the update of the penalty parameter β^k (step 3b). The plateau shaped areas (steps) correspond to iterations performed by the inner (subproblem) optimization in step 2. Of course, the length of the steps is related to the tolerances for the subproblem optimization.

Example 3:

The setting of parameters is same as in Example 2, except that now results are presented for networks of the size $n_E = 7$ (small scale, figure 4.2.19) and $n_E = 61$ (medium scale, figure 4.2.20). Again, the convergence history of all four smoothing kernels is plotted in the same figure. The observations coincide with those from example 2.

The different smooth approximations p_k show a qualitatively similar convergence behavior. However, starting the algorithm with smoothing kernel p_2 nearly triples the total number of iterations compared with p_3 or p_4. This is still true for smaller and larger networks, see example 3. When using the estimate (3.6.28) as initial penalty parameter β^0, less differences are observed in the overall performance. Still, the smoothing kernel p_2 performs worse than other kernels and this can also be observed in examples 2 & 3.

Table 4.2.9: Example 1: (Left table): Using Pinar-Zenios smoothing kernel p_3 and (Right table): Zang smoothing kernel p_4

α^k	β^k	Iterations
10	10	2
10	100	4
10	1000	39
10	10000	9
10	100000	13
50	100000	3
250	100000	31
1250	100000	33
6250	100000	33
31250	100000	33
156250	100000	5
781250	100000	4

α^k	β^k	Iterations
10	10	3
10	100	6
10	1000	8
10	10000	9
10	100000	13
50	100000	31
250	100000	33
1250	100000	33
6250	100000	33
31250	100000	5
156250	100000	32
781250	100000	4

4.3 Summary

In this chapter numerical results on two different models for the traffic flow in the road networks are presented. The first part of the chapter presents the results of the ODE model. This model exhibits similar qualitative properties with respect to the PDE model. It is found that the ODE model is cheaper in terms of computation time. The numerical computation of gradient using adjoint equations shows gradient vanishes at the optimum point. The optimization problems governed by the ODE model are solved using the first and second order gradient based methods. The optimization results are presented on sample networks and return optimal allowable traffic flux distribution factor. It is found that second order optimization methods show q-superlinear convergence.

The second part of the chapter includes optimization results of a nonlinear, equality and box constrained problem defining the traffic flow in the network. The traffic flow is governed by the RSA model. This constrained optimization problem is solved by exact penalization and smoothing method using the given estimates of the initial penalty parameter. The test is performed on a strip shaped network of different sizes. In addtion, the performance of the different smoothing kernels to the non-differentiable exact $l-$ penalty function is compared. In order to solve the bound constrained subproblem, two different solvers returning second order minimum point, L-BFGS-B and PL2 are used in the algoritm. It is found that the algorithm converges faster with PL2 as a solver. Also results support the claim that the initial estimate (3.6.28) is a reasonable estimate for the threshold of the penalty parameter corresponding to the model problem.

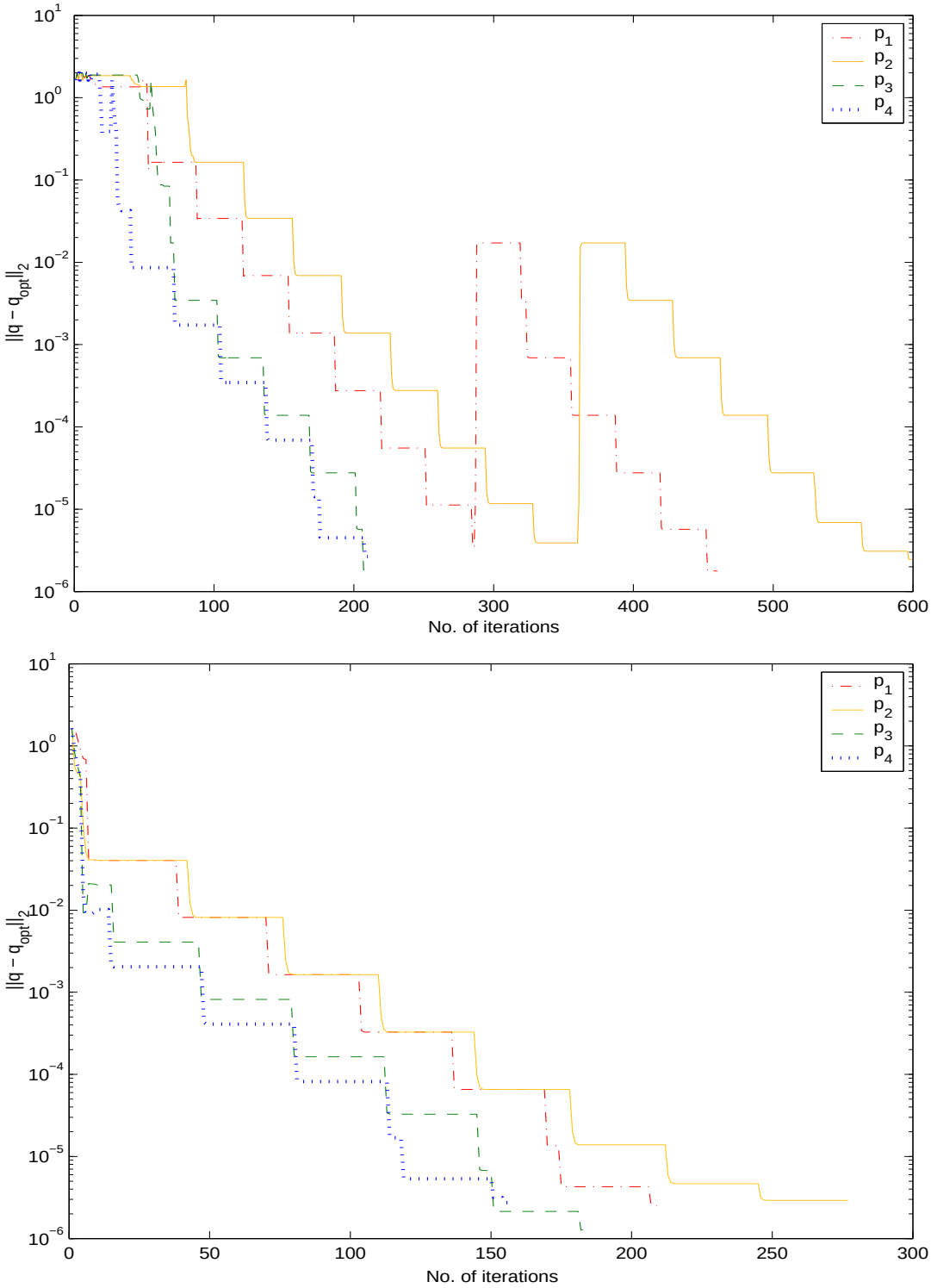

Figure 4.2.18: Example 2, $n_E = 31$, top part $\beta^{(0)} = 10$ and bottom part $\beta^{(0)}$ given by (3.6.28). The labeling of the smoothing kernels is according to Section 3.4. Iteration history of residuals of optimal values obtained from the execution of algorithm (3.5.13) together with the PL2 solver for subproblem (3.4.12).

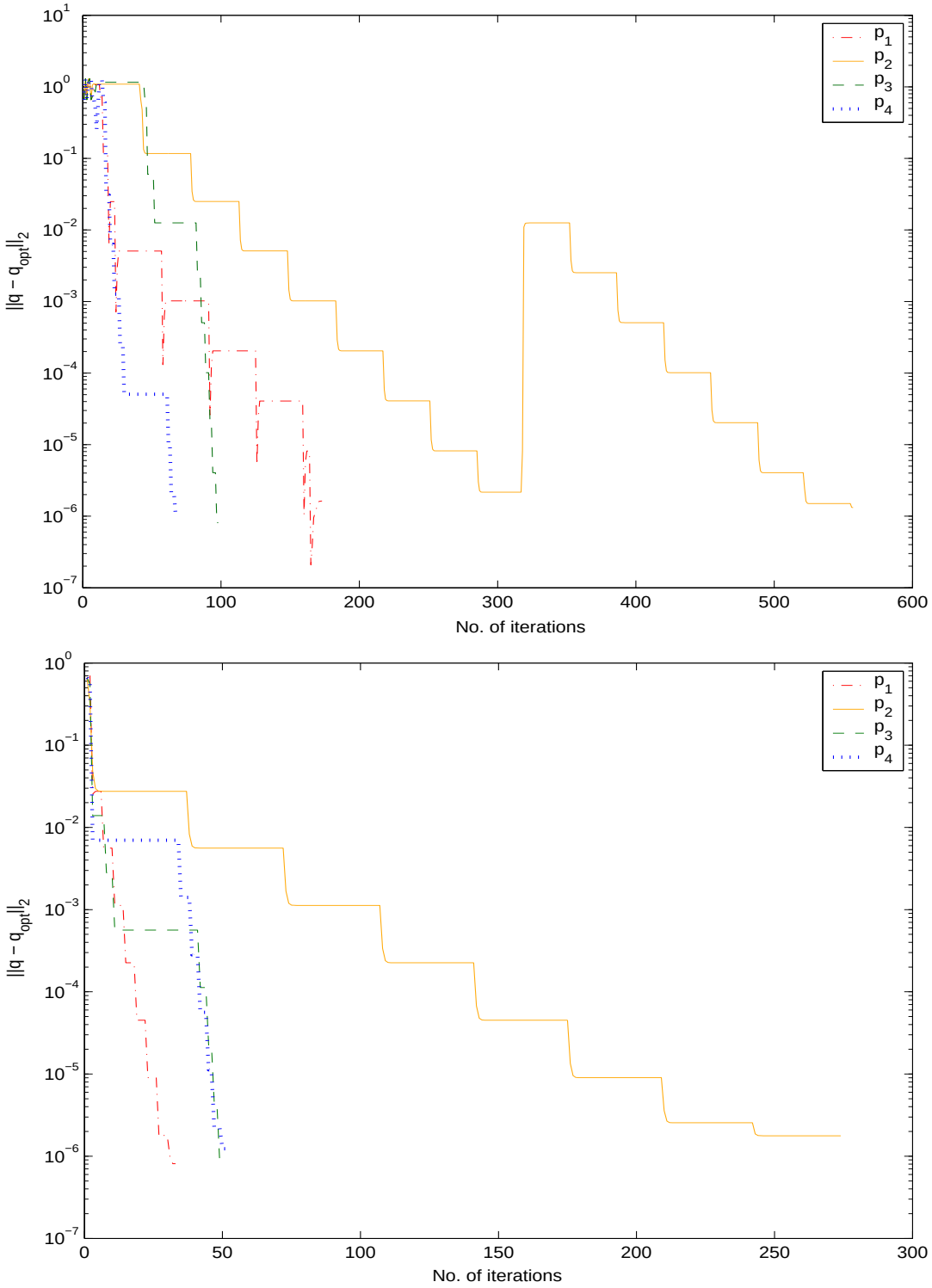

Figure 4.2.19: Example 3, $n = 7$, top part $\beta^{(0)} = 10$ and bottom part $\beta^{(0)}$ given by (3.6.28). The labeling of the smoothing kernels is according to Section 3.4. Iteration history of residuals of optimal values obtained from the execution of algorithm (3.5.13) together with the PL2 solver for subproblem (3.4.12).

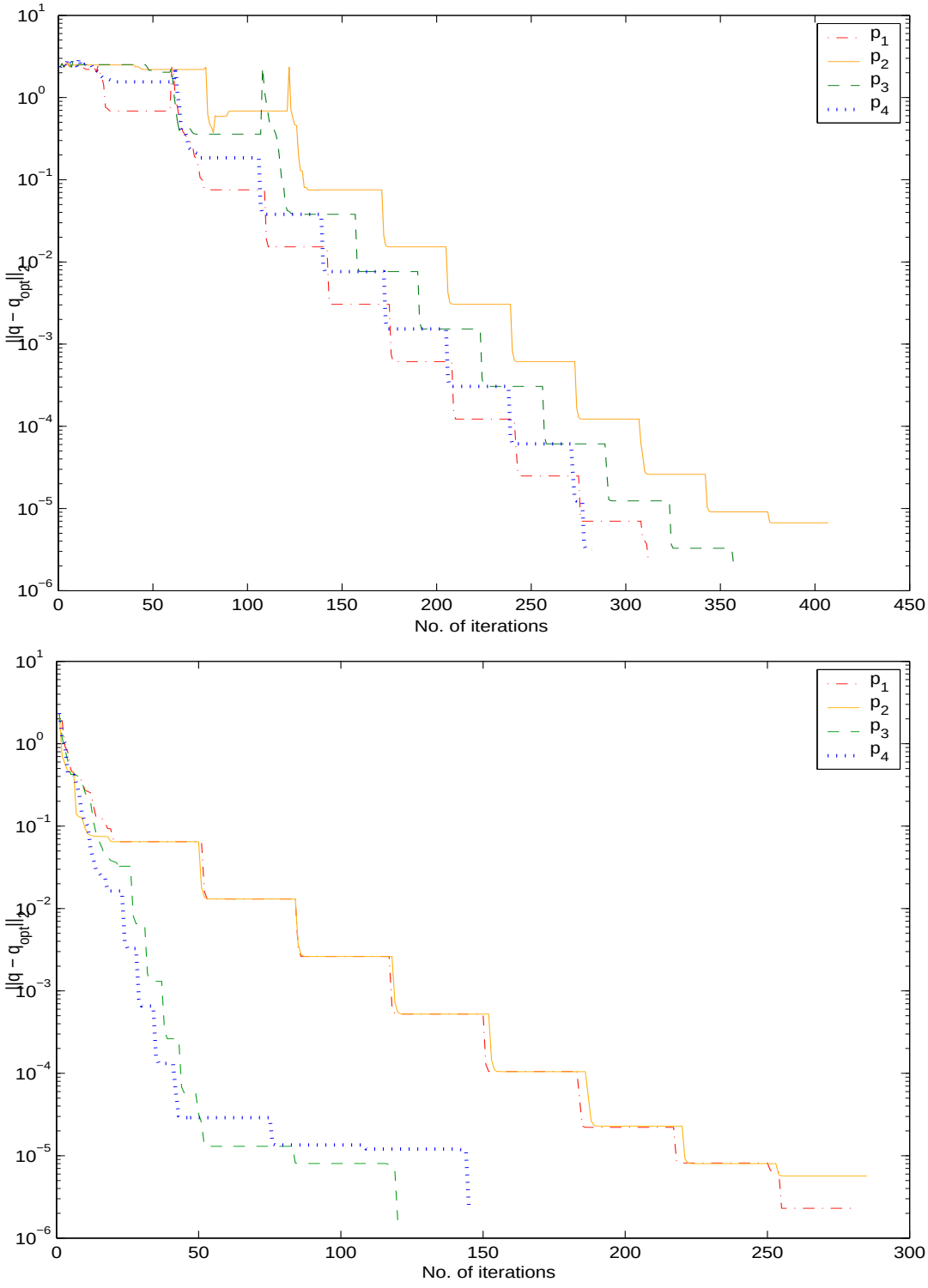

Figure 4.2.20: $n = 61$, Above: $\beta^{(0)} = 10$ and Below: $\beta^{(0)}$ given by (3.6.28). The labeling of the smoothing kernels is according to Section 3.4 Iteration history of residuals of optimal values obtained from the execution of algorithm (3.5.13) together with the PL2 solver for subproblem (3.4.12).

Chapter 5

Domain Decomposition for Conservation Laws

This chapter explains a domain decomposition method to solve problems with spatially discontinuous flux functions involved in one dimensional scalar conservation laws. In case of vehicular traffic flow, the number of lanes may change due to weather conditions or constructions and also, free flow velocity leads to a spatially varying discontinuous flux function.

5.1 Introduction

Conservation laws with discontinuous flux function occur in many physical applications, for example two phase flow in porous media [33], sedimentation phenomena [29] and vehicular traffic flows [75, 99]. These conservation laws are defined by nonlinear partial differential equations of the form,

$$\begin{array}{rcl} \partial_t u + \partial_x f(k(x), u) & = & 0, \quad (x,t) \in \mathbb{R} \times \mathbb{R}^+ \\ u(x,0) = u_0(x) & & \end{array} \tag{5.1.1}$$

where $u(x,t)$ is the scalar unknown function to be determined and $f(k(x),u)$ is a flux function of the form, $f(k(x),u) = k(x)f(u)$. The coefficient function $k(x)$ may have discontinuity due to the physical conditions or properties depending upon the problem under study. It is known in general that independent of smoothness of the $k(x)$ and of the initial data $u(x,0)$ discontinuities will develop in the solution $u(x,t)$ of (5.1.1).

The development of efficient solution procedures for such problems is challenging, since in many applications its solution involves abrupt changes in flow variables at discontinuities. Standard numerical methods often perform badly on conservation laws with discontinuous flux function due to the presence of critical regions of high spatial activity such as shocks, boundary layers, wavefronts, etc. The present solution either oscillates wildly in the vicinity of such regions if the mesh is too coarse, or fronts are smeared if too much dissipation is added to control the oscillations. Nevertheless, theoretical results

have been presented for example in [33, 61, 96]. Klingenberg & Risebro and Gimse & Risebro have transformed the scalar conservation law (5.1.1 with discontinuous coefficient $k(x)$ in x) into a system of 2×2 equations which is not strictly hyperbolic [33, 61]. The idea behind this transformation is that the behavior of $u(x,t)$ at discontinuities of $k(x)$ can be more easily analyzed. Also, existence and uniqueness of weak solutions has been proved for this class of flux functions satisfying a wave entropy condition in addition to the Rankine-Hugeniot condition of conservation law (5.1.1). This method involves 2×2 Riemann solvers which are complicated to implement. Another approach in [96] Towers has presented a scalar finite-difference scheme based on Godunov or Engquist-Osher numerical flux. The algorithm uses a scalar Riemann solver under the restriction that the flux function is concave. Also convergence, monotonicity of the corresponding numerical scheme are proved by using singular mapping approach. This scheme is also hard to implement as it involves discretization of $k(x)$ by a staggered grid with respect to $u(x,t)$.

Recently, there has been a lot of interest in the numerical treatment of discontinuous flux functions for hyperbolic equations, see for example [1, 50, 74, 92]. In [50] Karlsen et al. have derived a relaxation scheme which is a 2×2 semilinear system of hyperbolic equations with a relaxation term containing discontinuous flux function and proved the convergence of this scheme. In addition to this, in [92] Seaid has developed characteristic based relaxation method which is independent of CFL condition used for time steps.

Herein, an alternative view is presented to solve problems (5.1.1) by introducing a domain decomposition procedure [63, 69, 88]. Our approach is inspired from the recent work on partial differential equation on network geometries, see for example [4, 22, 45]. Therefore the coupling conditions for subdomains are similar to the coupling conditions at vertices of the network model. The spatial domain is decomposed into subdomains at the location of the discontinuity of the flux function. For each subdomain the associated conservation law having a continuous flux function is solved numerically using a nonoscillatory relaxation scheme.

Relaxation methods for hyperbolic equations of conservation laws have initially been introduced by Jin & Xin in [49]. The method consist of transforming the original conservation law by a semilinear hyperbolic system with linear characteristic variables and a relaxation source term. Main advantages of relaxation methods are the semilinear construction of the approximating system and a special time implicit-explicit splitting for the relaxation term. The first advantage allows us to solve the system numerically without introducing Riemann solvers and the second is to avoid the solution of nonlinear system of algebraic equations. Relaxation schemes are the combination of a non-oscillatory upwind space discretization and a TVD implicit-explicit time integration of the resulting semi-discrete system, see for instance [5, 6, 49, 91, 92].

In section 5.2 a Domain Decomposition Method (DDM) is introduced and explained. An algorithm based on the DDM to perform numerical computations is presented in section 5.3. To examine the performance of the method numerical results are presented on test

examples related to bottleneck situations in the traffic flow and two-phase flow in fluids in section 5.4.

5.2 A Domain Decomposition Method

The scalar conservation law where the flux function depends on space variable x in addition to u is

$$\begin{array}{rcl} \partial_t u + \partial_x f\,(x,u) & = & 0, \quad (x,t) \in \mathbb{R} \times \mathbb{R}^+ \\ u(x,0) & = & u_0(x), \quad x \in \mathbb{R} \end{array} \tag{5.2.2}$$

where the flux function $f(x,u)$ is piecewise differentiable and possibly has a finite number of discontinuities located at $\{x_1, \ldots, x_M\}$. Assume that on each spatial region $[x_m, x_{m+1})$ the function $(x,u) \longrightarrow f(x,u)$ is independent of the spatial variable x and strictly concave with single maximum. Assumptions on the flux function are motivated by examples on the traffic flow models where at certain location point x_m, a construction site or a lane reduction starts and continues until the next location point x_{m+1} on the road. In the traffic flow terminology, these discontinuity points are known by junctions and a traffic road can contain more than one junction conducting to the notion of networks. The restriction to this (special) class of flux functions allows for a short presentation of the main ideas. However, the whole method can be applied to far more general flux functions and later on numerical results on general discontinuous flux function are presented. An example of a flux function under consideration is depicted in figure 5.2.1.

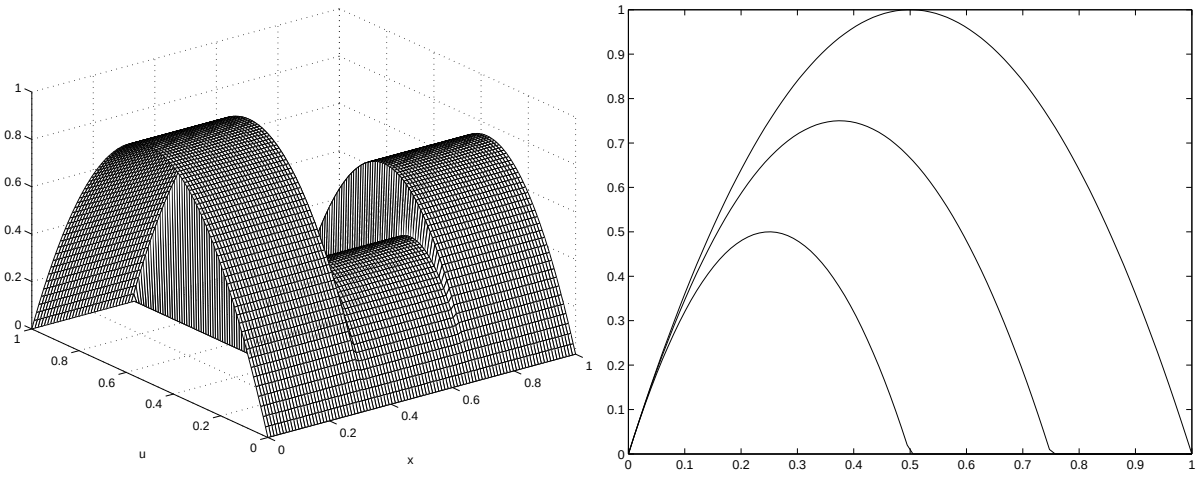

Figure 5.2.1: Flux function $f(x,u)$ with two discontinuities located at $x_1 = 1/3$ and $x_2 = 2/3$ being piecewise concave (right plot) and its interpretation as a network with three independent flux functions (left plot) where arcs of the network are connected at x_i.

The basic idea of this approach is to apply a domain decomposition method to the problem (5.2.2). Thus, the spatial domain is decomposed into non-overlapping subdomains $\Omega_m =$

$[x_m, x_{m+1})$ such that

$$\bigcup_{m=0}^{M} \Omega_m = \mathbb{R}, \qquad \Omega_k \cap \Omega_m = \emptyset, \quad \text{for} \quad k \neq m.$$

Hence, the problem (5.2.2) reduces to a sequence of M scalar conservation laws

$$\begin{aligned} \partial_t u_m + \partial_x f_m(u_m) &= 0, \qquad (x,t) \in \Omega_m \times \mathbb{R}^+, \\ u_m(x,0) &= u_0(x), \quad x \in \Omega_m, \end{aligned} \tag{5.2.3}$$

where the flux function $f_m(u)$ is defined by $f_m(u) = f(x,u)$ for $x \in \Omega_m$. By assumption the flux function in (5.2.3) is a continuous function. Note that the subdomains Ω_1 and Ω_M are extended to $-\infty$ and $+\infty$, respectively. Boundary conditions necessary for the numerical solution of (5.2.3) on each subdomain Ω_m are given by the coupling conditions given below.

Once solutions u_m of the conservation laws (5.2.3) are obtained, the solution of the original problem (5.2.2) is recovered by prolongation as

$$u(x,t) = u_m(x,t), \qquad (x,t) \in \Omega_m \times \mathbb{R}^+.$$

This approach has the advantage that each single conservation law is well-posed and can be solved by standard numerical methods designed for conservation laws with continuous flux functions. A remark is that the domain decomposition procedure can also be seen as a network problem with a particular simple geometry and different flux functions f_m on each road, see for example [45, 22]. Coupling conditions are derived, as in reference [45], from the weak formulation of decomposed problems (5.2.3) as follows:

Consider a set of test functions $\{\phi_m\}_{m=1}^{M}$ with $\phi_m : \Omega_m \times [0, +\infty] \longrightarrow \mathbb{R}$ having compact support in Ω_m and, for all $t \in \mathbb{R}^+$, satisfying the following conditions

$$\begin{aligned} \phi_m(x_m,t) &= \phi_{m+1}(x_{m+1},t), \\ \partial_x \phi_m(x_m,t) &= \partial_x \phi_{m+1}(x_{m+1},t), \qquad \forall\, m = 1,\ldots,M-1. \end{aligned} \tag{5.2.4}$$

Let u be a weak solution to the problem (5.2.2) and let ψ be an arbitrary test function, $\psi \in C_0^\infty(\mathbb{R} \times \mathbb{R}^+)$. It is easy to verify that functions ϕ_m are defined as the restriction of ψ on Ω_m, *i.e.*

$$\phi_m(x,t) = \psi(x,t), \qquad (x,t) \in \Omega_m \times \mathbb{R}^+,$$

satisfy conditions (5.2.4). Similarly, the restricted solution $u_m := u\big|_{\Omega_m \times \mathbb{R}^+}$ solves the conservation law (5.2.3) and the weak formulation for (5.2.2) yields

$$\int_0^\infty \int_{\mathbb{R}} \partial_t \psi u + \partial_x \psi f(x,u) dx dt + \int_{\mathbb{R}} \psi(x,0) u_0(x) dx = \sum_{m=1}^{M} \left(\int_0^\infty \int_{\Omega_m} \partial_t \phi_m u_m + \partial_x \phi_m f_m(u_m) dx + \int_{\Omega_m} \phi_m u_0(x) dx \right) = 0. \tag{5.2.5}$$

It is required that (5.2.5) holds for all sets of test functions $\{\phi_m\}_{m=1}^M$ with the property (5.2.4) and if each u_m is sufficiently regular, then the Rankine-Hugenoit jump conditions hold for all $t \in \mathbb{R}^+$:

$$f_m\left(u_m\left(x_m^-, t\right)\right) = f_{m+1}\left(u_{m+1}(x_m^+, t)\right), \qquad m = 1, \ldots, M-1. \tag{5.2.6}$$

Note that conditions (5.2.6) are the appropriate coupling conditions which will be used to define boundary values for (5.2.3). For convenience, a brief derivation is given below using a different motivation and notation than those presented in [45].

For each single subdomain Ω_m, let us consider the Riemann problem associated with the problem (5.2.3) at the boundary points $\hat{x}$ of Ω_m and defined as

$$\begin{aligned} \partial_t u_m + \partial_x f_m(u_m) &= 0, \qquad (x,t) \in \Omega_m \times \mathbb{R}^+, \\ u_m(x,0) &= \begin{cases} u^-, & \text{if} \quad x < \hat{x}, \\ u^+, & \text{if} \quad x > \hat{x}. \end{cases} \end{aligned} \tag{5.2.7}$$

The Riemann data is defined for $t = 0$ depending on the left boundary $\hat{x} = x_m$ or the right boundary $\hat{x} = x_{m+1}$ as

$$\begin{cases} u^- = u_0(x), & \text{if} \quad \hat{x} = x_{m+1}, \\ u^+ = u_0(x), & \text{if} \quad \hat{x} = x_m. \end{cases} \tag{5.2.8}$$

An (entropy) solution of (5.2.5) can be constructed such that all generated waves have locally non-positive $\hat{x} = x_{m+1}$ and non-negative $\hat{x} = x_m$ speed, respectively. Moreover, all solutions u_m satisfy the condition (5.2.6) for $x = x_m$ and $x = x_{m+1}$. The previous discussion holds true for all kinds of flux function. For simplicity of the presentation now only concave flux functions $f_m : \; u \longrightarrow f_m(u)$ are considered. Then, assumptions on the speed of waves can be reformulated more explicitly by using the notion of demand and supply functions [64]. In the sequel, it is assumed that the concave flux functions f_m additionally satisfy $f_m(0) = 0$ and $f_m(u_{m,\max}) = 0$ for some value $u_{m,\max} > 0$, and the unique single maximum is reached at $u = \sigma_m$. Then, the demand function $d_m : \; u \longrightarrow d_m(u)$ corresponds to the non-decreasing part of f_m *i.e.*,

$$d_m(u) := \begin{cases} f_m(u), & \text{if} \quad u \leq \sigma_m, \\ f_m(\sigma_m), & \text{otherwise.} \end{cases}$$

Analogously, the supply function $s_m : \ u \longrightarrow s_m(u)$ corresponds to the non-increasing part of the flux function f_m *i.e.*,

$$s_m(u) := \begin{cases} f_m(\sigma_m), & \text{if} \quad u \leq \sigma_m, \\ f_m(u), & \text{otherwise.} \end{cases}$$

Using demand/supply functions one can easily characterize the Riemann data $u^{\pm}$ in (5.2.8), which gives rise to waves of positive or negative speeds. For example in context of traffic flow, fundamental diagram defines flux function which is a concave function of density only. It retains a maximum (the capacity) at the critical density as explained in chapter 1. When traffic density is less than the critical density, it is in under-critical region and in over-critical region if more than the critical density. Here the traffic demand function is the flow rate when traffic density is in under-critical region or capacity if it is in over-critical region. The traffic supply function is the capacity when traffic density is in under critical or the flow rate when is in over-critical region. In the figure 5.2.2 demand and supply functions are plotted according to the above definition and corresponding to a concave flux function.

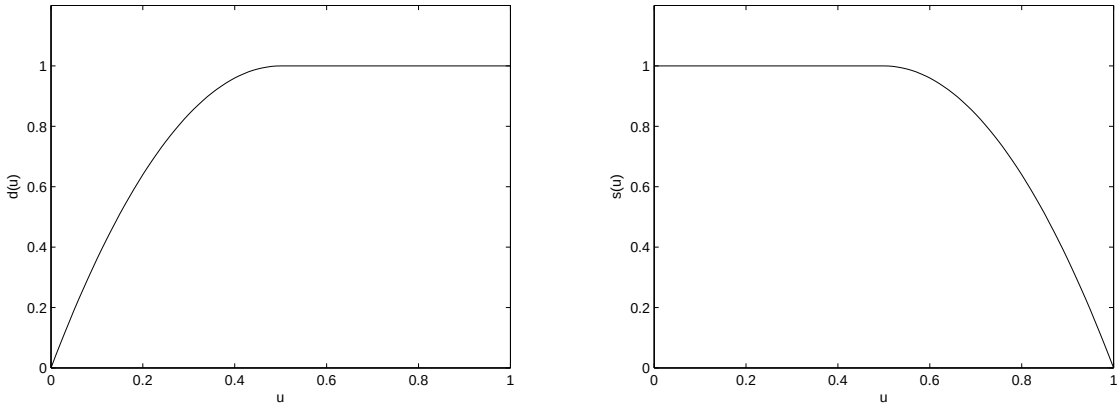

Figure 5.2.2: The demand function (left plot) and the supply function (right plot) for concave function defined in 5.2.1

Remark 5.2.1. *The concept of demand and supply functions can be extended to functions which are neither convex nor concave, see for example [28]. Then, the similar statements as in Propositions 4 and 5 are much more involved.*

Proposition 4 *Consider the problem (5.2.7) with a flux function satisfying the above assumptions and let the constant initial data u^- be given. Assume a flux function q is given such that*

$$0 \leq q < d_m(u^-).$$

Then, there exists a unique state u^+ such that the Riemann problem (5.2.7) admits only waves of negative speed. In addition, if $q = d_m(u^-)$ then set $u^+ = u^-$ and the solution to (5.2.7) is constant. □

Proof: This proof is standard and the two possible cases are:

(i) If $u^- < \sigma_m$ then $d(u_m) = f_m(u_m)$. Thus, by concavity of the flux function, there exists a point $\tau(u^-) > \sigma_m$ such that $f_m(\tau(u^-)) = f_m(u^-)$. Now, since f_m is continuous and strictly concave, there exists a unique state $u^+ \in [\tau(u^-), u_{m,\max}]$ such that $f_m(u^+) = q$. Moreover, the solution to (5.2.7) with data u^- and u^+ is a shock wave of negative speed.

(ii) If $u^- > \sigma_m$, then $d(u_m) = f_m(\sigma_m)$. Thus, there exists a unique state $u^+ \in [\sigma_m, u_{m,\max}]$ such that $f_m(u^+) = q$. Moreover, the solution to (5.2.7) is a rarefaction wave of negative speed if $u^+ < u^-$ and a shock wave otherwise. ■

Proposition 5 *Consider the problem (5.2.7) with a flux function satisfying the above assumptions and let the constant initial data u^+ be given. Assume a flux q is given such that*

$$0 \leq q < s_m(u^+).$$

Then, there exists a unique state u^- such that the Riemann problem (5.2.7) admits only waves of positive speed. In addition, if $q = s_m(u^+)$ then set $u^- = u^+$ and the solution to (5.2.7) is constant. ■

The proof is similar to the one of Proposition 4. By virtue of Proposition 4 and Proposition 5 a solution to (5.2.5) and (5.2.2) satisfying the coupling conditions (5.2.6) can be described.

For brevity, let us assume that the flux function $f(x,u)$ in (5.2.2) has a single discontinuity located at the point $x = x_m$. Hence, the proposed domain decomposition method leads to two problems (5.2.3) which are to be solved in subdomains Ω_{m-1} and Ω_m. Let us also assume that (at least locally) the constant initial data $u_0(x) = u_k^0$ for $x \in \Omega_k$, $k \in \{m-1, m\}$ and with possibly different flux functions $f_{m-1} \neq f_m$. Then, the theorem below describes the admissible solutions u_{m-1} and u_m satisfying (5.2.6) and (5.2.3).

Theorem 5.1 *Consider problems (5.2.3) in subdomains Ω_{m-1} and Ω_m with constant initial data $u_0(x) = u_{m-1}^0$ and $u_0(x) = u_m^0$ for $x \in \Omega_{m-1}$ and $x \in \Omega_m$, respectively. Then, there exists unique solutions $u_{m-1}(x,t)$ and $u_m(x,t)$ of the Riemann problems (5.2.7) and (5.2.8) at the junction with the following properties:*

(i) The solution $u_k(x,t)$ is a weak solution of the network problem (5.2.3) for $k \in \{m-1, m\}$ (and also to (5.2.5)). Furthermore, the condition (5.2.6) is satisfied.

(ii) The flux value $u_m(x_m^-, t)$ is maximal at the interface. □

Proof: Let $q := \min(d_{m-1}(u_{m-1}^0), s_m(u_m^0))$ be the flux at the interface $x = x_m$. Then, due to Preposition 4, there exists an admissible state $\bar{u}_{m-1}$ such that $f_{m-1}(\bar{u}_{m-1}) = q$. Furthermore, the solution to the Riemann problem (5.2.7) with initial data $u^- = u_{m-1}^0$ and $u^+ = \bar{u}_{m-1}$ is either constant or it consists of waves of negative speed only. Therefore, $u_{m-1}(x_m^-, t) = q$ for all $t \in \mathbb{R}^+$.

Similarly, due to Proposition 5, there exists an admissible state $\bar{u}_m$, such that $f_m(\bar{u}_m) = q$ and the solution to the Riemann problem (5.2.7) with initial data $u^- = \bar{u}_m$ and $u^+ = u_m^0$ is either constant or consists of waves with positive speed. Therefore, $u_m(x_m^+, t) = q = u_{m-1}(x_m^-, t)$. Since q is defined as minimum of supply and demand functions, the flux is maximal at the interface $x = x_m$.

■

Note that the theorem 5.1 describes admissible boundary values for a numerical procedure. A piecewise constant approximation u_m^n of u_m on Ω_m has to be supplemented with boundary conditions for example at $x = x_m$. The correct values to be prescribed at $x = x_m$ are given by the states $u_m^{n+1} = \bar{u}_m$ of (5.1). Similarly, the approximation u_{m-1}^n is supplemented with the boundary conditions at $x = x_m$ given by $\bar{u}_{m-1}$. If there is more than one discontinuity in the flux function $f(x, u)$, then the above has to be repeated at each single discontinuity point. This holds for first order methods only and is implemented in the numerical approach below.

Remark 5.2.2. *Obviously, if the flux function $f(x, u)$ is independent of x, the above discussion yields the entropy solution for a standard Riemann problem with initial data given by*

$$u(x,t) = \begin{cases} u^-, & \text{if} \quad x < x_m, \\ u^+, & \text{if} \quad x > x_m. \end{cases}$$

5.3 Domain Decomposition Algorithm

For completeness, the domain decomposition algorithm for solving the equation (5.2.2) is presented. It consists of the following steps:

Given x_{min}, x_{max} and a point of discontinuity of flux function, x_c.

1. Decompose the spatial domain into a finite sequence of subdomains $\Omega_m = [x_m, x_{m+1})$, $m = 1, \ldots, M$, using the location points x_m of the discontinuity in the flux function.
2. Initialize vectors u_m on each subdomain Ω_m to the initial value u_0 , respectively.
3. For $m = 1, \ldots, M$:
   - **i.** Reconstruct the boundary values at x_m and x_{m+1} according to the coupling conditions (5.2.4).
   - **ii.** Solve the sub-equations of conservation laws with the continuous flux functions in Ω_m using the numerical procedure described in [92] and briefly described in Appendix B.
4. Update the numerical solution u of (5.2.2) using the sub-solution u_m in Ω_m.

Note that, steps 3 and 4 have to be carried out at each time step in the computational process. One should make a point that the DDM algorithm proposed in the previous section 5.2 differs from the canonical DDM methods used for elliptic partial differential equation. In fact in DDM no iterations are required for resolving interfaces.

5.4 Results and Numerical Examples

In this section the performance of the proposed domain decomposition method is investigated numerically. To this end the method is applied to several test examples on conservation laws with discontinuous flux functions showing that the method appears to be promising. In order to compare the new method, the relaxation schemes presented in [92] are considered. Here DDM, RELAX1 and RELAX2 stand for the domain decomposition method which is introduced in section 5.2, the standard first order relaxation schemes and the second order relaxation schemes presented in [92], respectively and briefly reviewed in Appendix B. In all computations the CFL number is fixed to 0.75 and time steps Δt are calculated according to the condition given below,

$$\text{CFL} = \max_i \lambda_i \frac{\Delta t}{\Delta x} \leq 1 \tag{5.4.9}$$

where λ_i are characteristic speeds of the conservation law.

5.4.1 Accuracy Test Example

This example consists of equations (5.2.2) where the flux function is defined as

$$f(x,u) = k(x)u(1-u), \qquad k(x) = \begin{cases} 2, & \text{if} \quad 0 \leq x \leq 2.5, \\ \frac{25-2x}{10}, & \text{if} \quad 2.5 < x < 7.5, \\ 1, & \text{if} \quad 7.5 \leq x \leq 10, \end{cases} \tag{5.4.10}$$

with an initial condition given by

$$u_0(x) = \begin{cases} 0.9, & \text{if} \quad 0 \leq x \leq 2.5, \\ \frac{1+\sqrt{0.28}}{2}, & \text{if} \quad 2.5 < x \leq 10. \end{cases} \tag{5.4.11}$$

It is easy to verify that the problem (5.4.10)-(5.4.11) has an exact steady-state solution defined as

$$u_\infty(x) = \begin{cases} 0.9, & \text{if} \quad 0 \leq x \leq 2.5, \\ \dfrac{1}{2} + \dfrac{\sqrt{k(x)^2 - 0.72k(x)}}{2k(x)}, & \text{if} \quad 2.5 < x < 7.5, \\ \frac{1+\sqrt{0.28}}{2}, & \text{if} \quad 7.5 \leq x \leq 10. \end{cases}$$

The exact steady-state solution can be used to quantify results obtained by the DDM method in terms of error norms. According to the proposed DDM method, the flux function (5.4.10) results in three subproblems to be solved with continuous flux functions. The approximate solution is computed at $t = 10$. The L^∞- and L^1-error norms considered here are defined as

$$\max_{1 \leq i \leq N} |u_i - u_\infty(x_i)| \qquad \text{and} \qquad \sum_{i=1}^{N} |u_i - u_\infty(x_i)| \, \Delta x, \tag{5.4.12}$$

Table 5.4.1: Error-norms for the accuracy test example.

N	L^∞-error	L^1-error
100	1.79e-3	4.25e-3
200	9.03e-4	2.08e-3
400	4.52e-4	1.03e-3
800	2.26e-4	5.12e-4
1600	1.13e-4	2.56e-4

respectively. Here, u_i and $u_\infty(x_i)$ are the computed and the exact steady-state solutions at gridpoint x_i respectively, where N denotes the number of gridpoints used in the spatial discretization. Table 5.4.1 lists the obtained error-norms using different values of N. It is clear that increasing the number of gridpoints in the spatial domain turns into a decrease of error-norms in the computed solution.

In the figure 5.4.3 the computed solution is plotted using $N = 200$. To perform a comparison the results obtained using RELAX1 and RELAX2 are included. As seen in figure 5.4.3, there is no significant difference in resolutions of DDM and RELAX2 schemes. However, the RELAX1 scheme failed to accurately resolve the steady-state solution. The DDM method has performed well for this test example.

5.4.2 Traffic Flow Example

In this example the performance of the proposed DDM method is addressed for traffic flow models. By taking into account the nature of vehicular roads, these models offer a realistic one-dimensional conservation law with discontinuous coefficients. As explained earlier in chapter 1 the well-known Lighthill-Whitham and Richards model [68] for traffic flow in conservation form is studied with $u(x,t) = a(x)\rho(x,t)$ and $a(x)$ and $\rho(t,x)$ are the number of lanes and the density, respectively. The flux function is

$$f(x,u) = \frac{v(x)}{v_{max}} u(1-u), \tag{5.4.13}$$

where $v(x)$ is the free flow velocity at the point x and $v_{max} = \max_{x\in\mathbb{R}} v(x)$ is the maximum speed. Here, the accuracy of DDM method for a bottleneck situation in traffic flow is examined.

Consider a road of length $L = 10\ Km$ with an initial density $\rho(x,0) = 0.2\ veh/Km$. The coefficients $a(x)$ and $v(x)$ are discontinuous functions given by

$$a(x) = \begin{cases} 4, & \text{if} \quad x < 3\ Km, \\ 2, & \text{if} \quad x \geq 3\ Km, \end{cases}, \qquad v(x) = \begin{cases} 1, & \text{if} \quad x < 3\ Km, \\ 0.6, & \text{if} \quad x \geq 3\ Km. \end{cases} \tag{5.4.14}$$

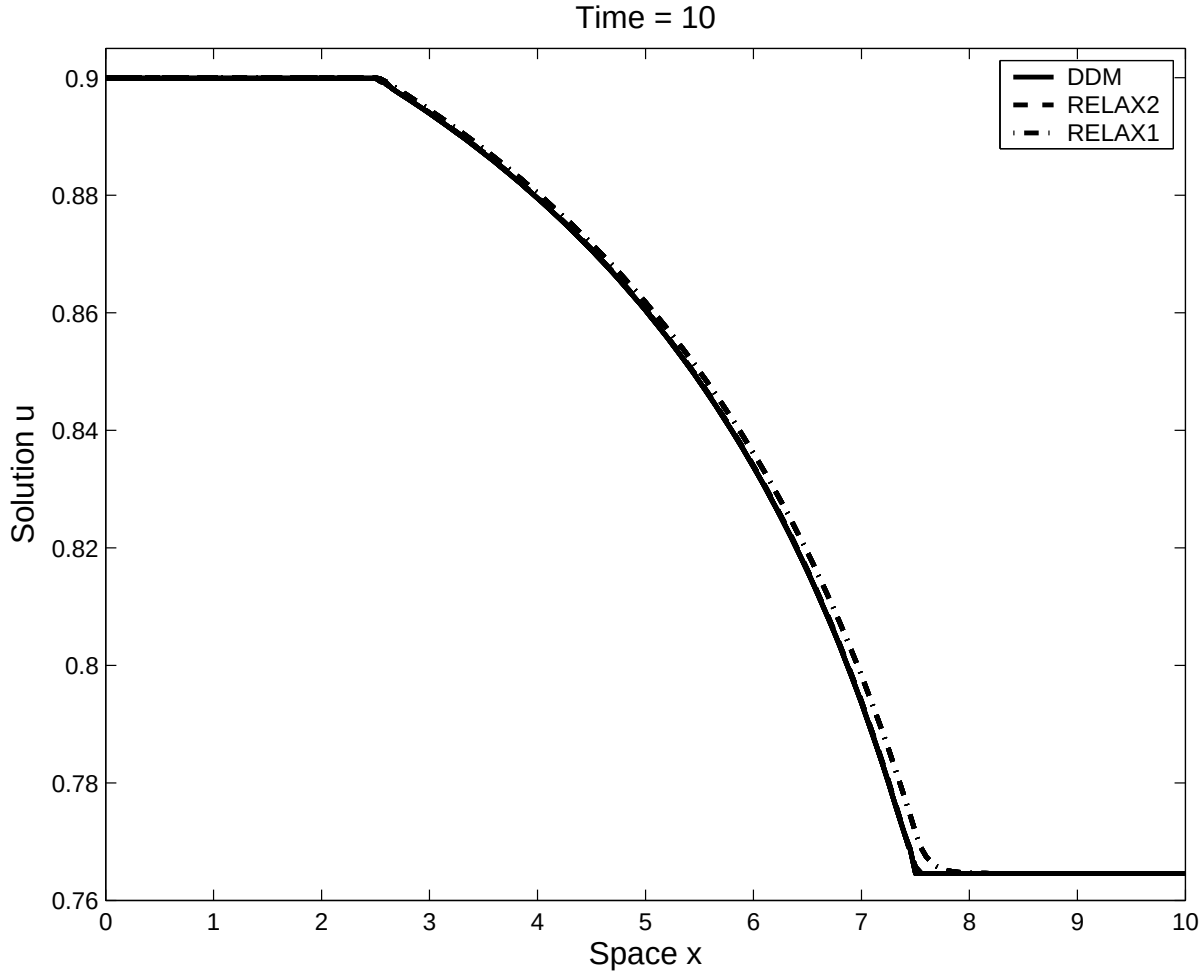

Figure 5.4.3: Results obtained for the accuracy test example.

The DDM method applied to equations (5.4.13)-(5.4.14) yields two subproblems to be solved in roads $[0, 3)$ and $[3, 10]$ with continuous flux functions. In the computation, the total road is divided into 100 gridpoints and the duration of the simulation is 900 s. In the figure 5.4.4 results are displayed using dimensionless space x/L and dimensionless time tv_{max}/L. As expected, the RELAX1 scheme introduces extensive numerical dissipation in the computed solution, the RELAX2 scheme performs better but still diffusive effects are presented in its resolution. This numerical dissipation is substantially eliminated in the DDM results and the high accuracy of DDM method over the other relaxation schemes is clearly demonstrated.

5.4.3 Two-Phase Flow Example

As already mentioned in Remark 5.2.1., the proposed domain decomposition method can also be applied if the flux function on each subdomain is neither convex nor concave as shown in figure 5.4.5. The standard example of such situation is the Buckley-Leverett flux

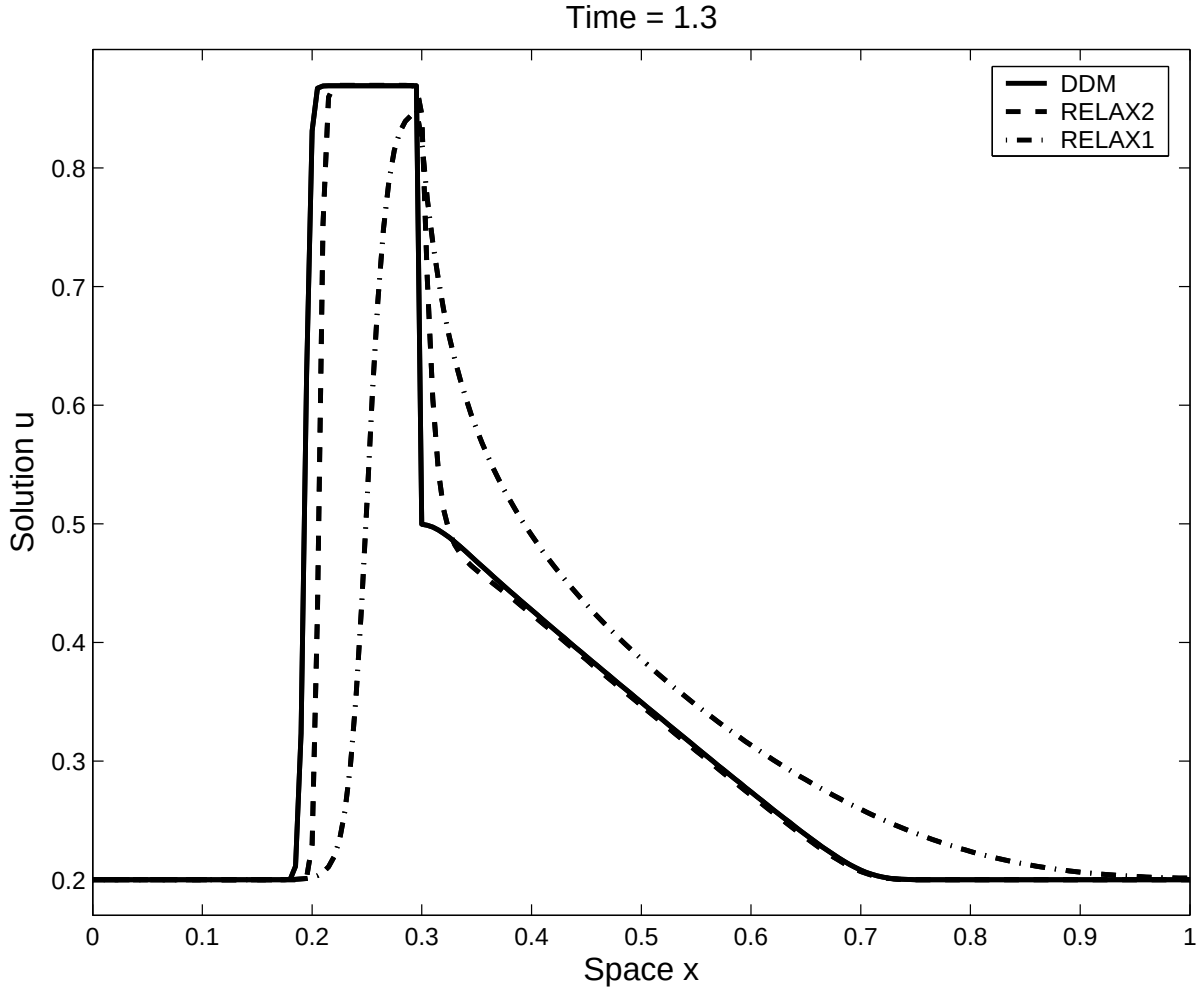

Figure 5.4.4: Results obtained for the example on traffic flow model.

function. The Buckley-Leverett equation serves as one of the simple model of two-phase flow in porous media [67]. Here, the governing equations are given by (5.2.2) with

$$f(k(x),u) = \frac{u^2}{u^2 + k(x)(1-u)^2}, \qquad k(x) = \begin{cases} 50, & \text{if} \quad 0 \le x < 0.5, \\ 5, & \text{if} \quad 0.5 \le x \le 1. \end{cases} \tag{5.4.15}$$

The initial condition is

$$u_0(x) = \begin{cases} 0, & \text{if} \quad 0 \le x \le 1 - \frac{1}{\sqrt{2}}, \\ 1, & \text{if} \quad 1 - \frac{1}{\sqrt{2}} \le x \le 1. \end{cases}$$

Again, the DDM procedure decomposes this test example in two subproblems with continuous flux functions to be solved in the sub-intervals $[0, 0.5)$ and $[0.5, 1]$. Here, the spatial domain is discretized into 200 gridpoints and computed results are presented at time $t = 0.2$.

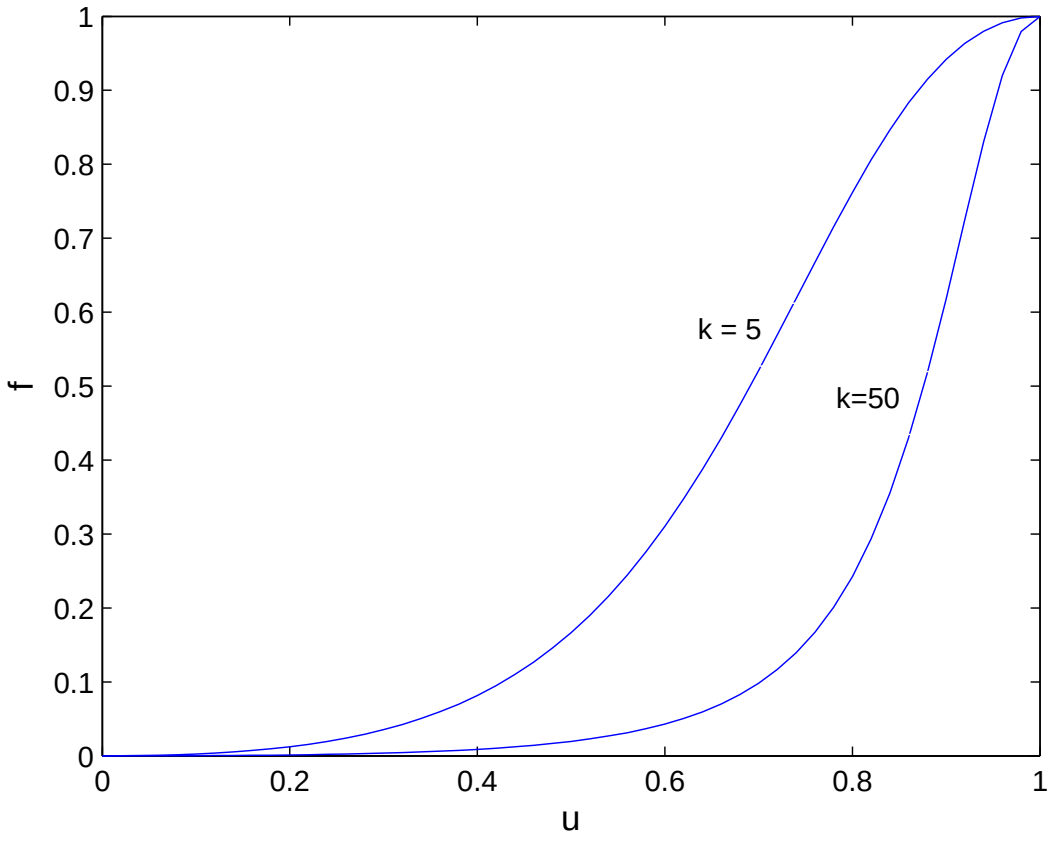

Figure 5.4.5: Buckley-Leverett flux function.

Figure 5.4.6 shows, results computed using DDM, RELAX1 and RELAX2 schemes. As can be seen in the figure 5.4.6, the results obtained by the DDM method are more accurate than those obtained using the other relaxation schemes. Results again show very good performance of the DDM method for this test example too.

5.5 Summary

In this chapter, a class of domain decomposition method for numerical solution of conservation laws with discontinuous flux functions is constructed and studied. This method is effective even when the flux function presents more than one discontinuity within the same spatial domain. The main idea is to transform the problem at hand into a finite sequence of conservation laws with continuous flux functions, which are easy to solve using conventional discretization. The coupling conditions for the boundary interfaces are established. The performance of the proposed method has been illustrated in several test examples on conservation laws with discontinuous flux functions. In order to feel the effectiveness of DDM approach comparisons to the second-order relaxations scheme has also been presented. It is found that the domain decomposition method is clearly

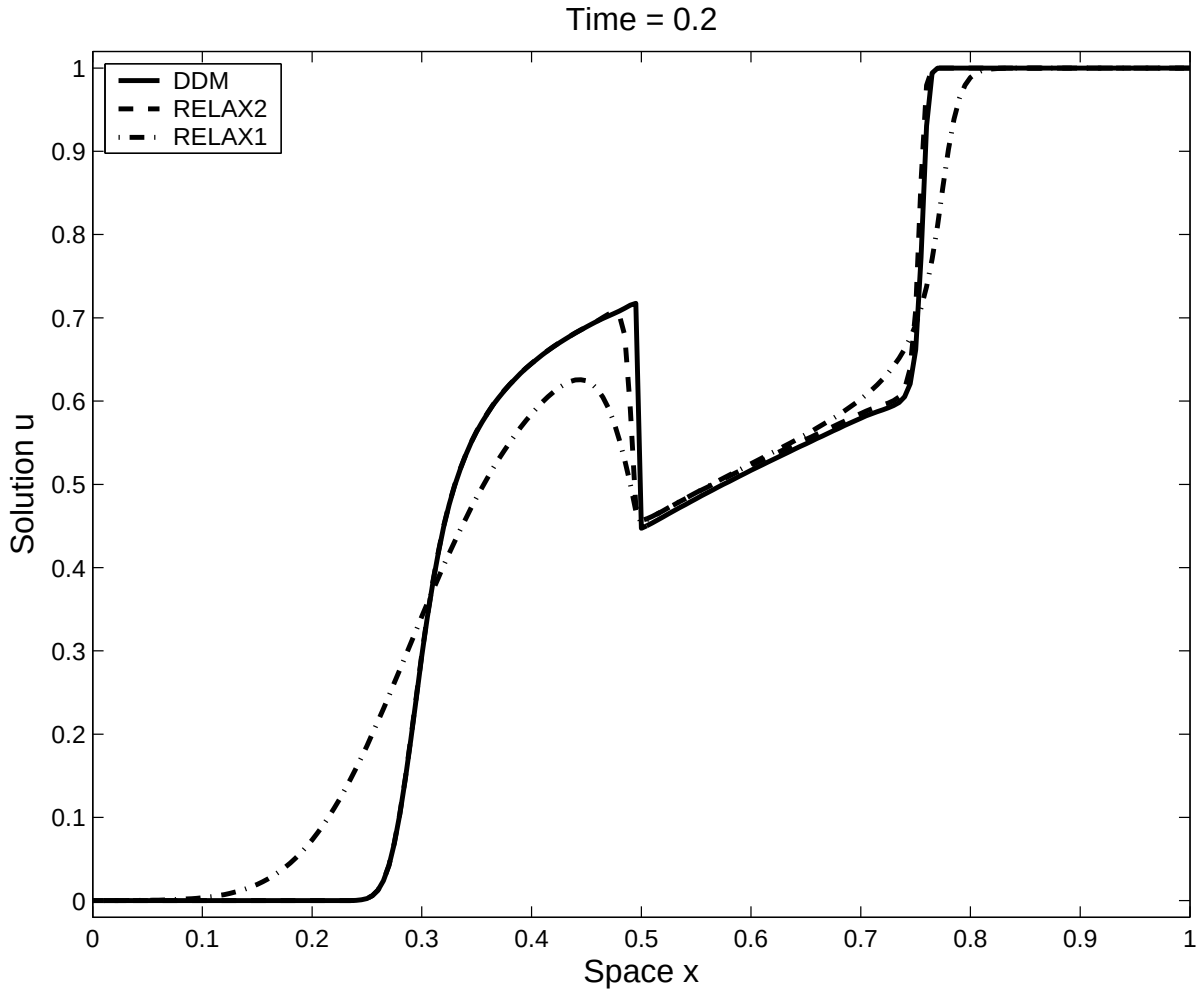

Figure 5.4.6: Results obtained for the example on two-phase flow model.

superior to other schemes. The obtained results demonstrate good shock resolution with high accuracy in smooth regions and without any nonphysical oscillations near the shock areas. For system of conservation laws, the principle of domain decomposition techniques can still be used, however, a different construction of coupling conditions at interfaces is needed.

Chapter 6

Summary and Outlook

The whole work is summarized and possible future research perspectives of the present work are given.

6.1 Summary

Starting from a macroscopic fluid-dynamic traffic flow model on networks, a model based on ordinary differential equations (ODE) is derived. In the network junctions with a total of three roads are considered. The flow is controlled in the network at dispersing junctions by controls α. This describes how traffic can be distributed according to fixed ratios at the junction. The ODE model enjoys similar qualitative properties as PDE model in interesting cases, free and congested flow. Particularly in the latter case, backward moving waves appear in the ODE model resembling a major feature of the PDE model. The derivation of adjoint equations to compute gradient of the cost functional is given for arbitrary networks. Numerical results for sample networks which are partly of academic interest are presented and discussed. The corresponding optimal control problem is computable at reasonable cost for mid-size networks. Gradient based projected steepest descent method is a natural choice to obtain optimal controls, since explicit formulas for the gradient of the cost functional are available and linear convergence is obtained. Also projected Quasi–Newton methods are used as a solver for nonlinear, bound constrained optimization problem and obtained super–linear convergence. In the comparison the expected convergence behavior is seen and due to the academic network the known optimal values are obtained.

In addition the simplified algebraic model [44] is reduced to a model defined in terms of flux only. The obtained optimization problem corresponding to this model is a nonlinear, equality and bound constrained problem. The PL2 penalty algorithm is used successfully to solve this, but still there are numerical and theoretical questions on the update of penalty and smoothing parameters. In addition different smooth approximations to the exact non-differentiable penalty function are introduced and compared in order to efficiently use the bound constrained optimization methods. It is found that the proposed initial estimate of the penalty parameter is a good approximation of the defined threshold

value for the model problem.

Later conservation laws with discontinuous flux functions are considered. A real world example is a bottleneck situation on a road. A domain decomposition method has been proposed to solve this conservation law. The effectiveness of the method is compared with the second order relaxation schemes and it is found that this method shows higher accuracy over relaxation schemes. The performance of the method is very attractive since this does not involve solution of Riemann problems but it requires special front tracking techniques.

6.2 Open Questions & Possible Extensions

Real data has been recorded and available on many highways. It is needed to adapt the ODE model for traffic flow on networks, in order to perform real-time simulations and optimization of flow. Another possibility left to check is the behavior of the ODE model by considering different flux functions as listed in [45].

The performance of the penalty based optimization algorithm can be enhanced by estimation of the penalty parameter and by adaptively choosing the update rule depending on the structure of the network [48]. The penalty based optimization method has to be tested further on other sample problems and has to be compared with the other well known methods to evaluate its benefit.

The domain decomposition method can be extended to situations involving time varying discontinuities. In addition, the same ideas can also be extended to two-dimensional equations of conservation laws with discontinuous flux functions

Appendix A: Continuous Adjoint Equations

A general system of continuous adjoint equation is derived, when the state equations are given as a system of differential equations,

$$\left.\begin{array}{lll} \dot{y}(t) & = & F(y(t),u(t)), \quad \text{for } t \in (0,T) \\ y(0) & \text{given}, & \end{array}\right\} \tag{A.1.1}$$

where $y(t) \in \mathbb{R}^m$ denotes the state variable and $u(t) \in \mathbb{R}^n$ is the control variable.

Let the cost functional be f:

$$\mathcal{F} = \mathcal{F}(y,u) = \int_0^T \mathcal{F}(y(t),u(t))dt \tag{A.1.2}$$

and consider the optimization problem,

$$\begin{array}{c} \min\limits_{u} \mathcal{F}(y,u) \\ \text{subject to} \quad (A.1.1) \end{array}$$

Let us assume that $\mathcal{F}$ and F are continuously differentiable functions. One needs to compute $\nabla_u \mathcal{F}$ that is,

$$\nabla_u \mathcal{F} = \int_0^T \mathcal{F}_y(y(t),u(t)).y_u(t)dt + \int_0^T \mathcal{F}_u(y(t),u(t)).1dt, \tag{A.1.3}$$

which involves the additional unknown term $y_u(t) = \nabla_u y(t)$. To get rid of this term the following is done:

First multiply equation (A.1.1) by a function $p(t) \in \mathbb{R}^m$ which is a adjoint state variable, one obtains,

$$p(t).\dot{y}(t) = p(t).F(y(t),u(t)) \tag{A.1.4}$$

Integrating from 0 to T with respect to t and using integration by parts on left hand side term, one obtains,

$$p(t).y(t)|_{t=0}^{t=T} - \int_0^T y(t).\dot{p}(t)dt = \int_0^T p(t).F(y(t),u(t))dt \tag{A.1.5}$$

On rewriting equation (A.1.5),

$$p(T)y(T) - p(0)y(0) - \int_0^T \dot{p}(t)y(t)dt = \int_0^T p(t)F(y(t),u(t))dt \tag{A.1.6}$$

Differentiating (A.1.6) with respect to control variable u on both sides one gets,

$$-p(T).y_u(T) + \int_0^T \dot{p}(t).y_u(t)dt + \int_0^T p(t)F_y(y(t),u(t)).y_u(t)dt + \int_0^T p(t)F_u(y(t),u(t))dt = 0 \tag{A.1.7}$$

On Adding (A.1.3) and (A.1.7) one gets,

$$\nabla_u\mathcal{F} = \int_0^T \mathcal{F}_y(y(t),u(t)).y_u(t)dt + \int_0^T \mathcal{F}_u(y(t),u(t)).1dt - p(T).y_u(T) + \int_0^T \dot{p}(t).y_u(t)dt + \int_0^T p(t)F_y(y(t),u(t)).y_u(t)dt + \int_0^T p(t)F_u(y(t),u(t))dt \tag{A.1.8}$$

By collecting similar terms in the above expression and rewriting the equation:

$$\nabla_u\mathcal{F} = \int_0^T \left(\mathcal{F}_y(y(t),u(t)) + \dot{p}(t) + p(t)F_y(y(t),u(t))\right).y_u(t)dt - p(T).y_u(T) + \int_0^T \mathcal{F}_u(y(t),u(t)).1dt + \int_0^T p(t).F_u(y(t),u(t))dt \tag{A.1.9}$$

Equating coefficients of $y_u(t)$ to zero one gets

$$\begin{aligned} p(T) &= 0 \\ \mathcal{F}_y(y(t),u(t)) + \dot{p}(t) + p(t)F_y(y(t),u(t)) &= 0 \end{aligned}$$

or

$$\left.\begin{aligned} p(T) &= 0 \\ -\dot{p}(t) &= \mathcal{F}_y(y(t),u(t)) + p(t)F_y(y(t),u(t)) \end{aligned}\right\} \tag{A.1.10}$$

Hence gradient of f can be written in the following form:

$$\nabla_u\mathcal{F} = \int_0^T \mathcal{F}_u(y(t),u(t))dt + \int_0^T p(t)F_u(y(t),u(t))dt \tag{A.1.11}$$

On introducing the following notations,

$$\nabla_u F = H(t), \quad \nabla_u \mathcal{F} = h(t), \quad \nabla_y F = G(t), \quad \nabla_y \mathcal{F} = \gamma(t)$$

Equations (A.1.10) and (A.1.11) can be rewritten as

$$\left.\begin{array}{l} p(T) = 0 \\ -\dot{p}(t) = p(t)G(t) + \gamma(t) \end{array}\right\} \tag{A.1.12}$$

$$\nabla_u \mathcal{F} = \int_0^T \Big(h(t) + p(t)H(t) \Big)\, dt \tag{A.1.13}$$

It is assumed also that $p(t)$ is continuously differentiable function and finally the following is obtained,

$$\nabla_u \mathcal{F} = h(t) + p(t)H(t) \tag{A.1.14}$$

From (A.1.12) the adjoint variable $p(t)$ is obtained and substituted in (A.1.14). However, to solve this system numerically one needs an approximation of the differential equation e.g. by discretization. One ends up with the gradient of the cost functional.

Appendix B: A Relaxation Approach for Scalar Conservation Laws

To numerically solve the conservation laws (5.2.3) the well established relaxation methods are considered [5, 49]. Briefly some implementation details are presented. The method below can also be applied to discontinuous flux functions without any change, see [92]. In the chapter 5 this approach is referred to as RELAX2 and will be used as a comparison with the introduced domain decomposition.

Let us consider a single equation in conservation laws (5.2.3)

$$\begin{aligned} \partial_t U + \partial_x F(U) &= 0, && (x,t) \in \Omega \times \mathbb{R}_+, \\ U(\hat{x},t) &= \hat{u}(t), && t \in \mathbb{R}_+, \\ U(x,0) &= u_0(x), && x \in \Omega, \end{aligned} \tag{B.1.1}$$

where U, F, Ω and $\hat{x}$ refer respectively, to any variable u_m, flux function f_m, subdomain Ω_m and junction point x_m in (5.2.3) with $m = 1, \ldots, M$. A relaxation approximation of equations (B.1.1) reads

$$\begin{aligned} \partial_t U + \partial_x V &= 0, \\ \partial_t V + \lambda \partial_x U &= -\frac{1}{\varepsilon}\Big(V - F(U)\Big), \\ U(\hat{x},t) = \hat{u}(t), &\qquad V(\hat{x},t) = F\left(\hat{u}(t)\right), \\ U(x,0) = u_0(x), &\qquad V(x,0) = F\left(u_0(x)\right), \end{aligned} \tag{B.1.2}$$

where ε is the relaxation rate and V is a relaxation variable expected to converge to $F(U)$ as ε approaches to zero. The parameter λ is the characteristic speed selected based on the *sub-characteristic* condition [49],

$$\frac{|\partial_U F(U)|}{\sqrt{\lambda}} \leq 1. \tag{B.1.3}$$

The main advantage of numerically solving the relaxation system (B.1.2) over the original conservation law (B.1.1) lies in the special structure of the linear characteristic fields and localized lower order terms. Indeed, the linear hyperbolic nature of (B.1.2) makes it possible to approximate its solution easily by underresolved stable numerical discretization that uses neither Riemann solvers spatially nor nonlinear system of algebraic equations

solvers temporally.

As in may kinetic models, the relaxation system (B.1.2) can be reformulated in diagonalizable form

$$\begin{aligned} \partial_t \mathcal{F} + \sqrt{\lambda}\partial_x \mathcal{F} &= -\frac{1}{\varepsilon}\left(\mathcal{F} - \hat{\mathcal{F}}\right), \\ \partial_t \mathcal{G} - \sqrt{\lambda}\partial_x \mathcal{G} &= -\frac{1}{\varepsilon}\left(\mathcal{G} - \hat{\mathcal{G}}\right), \end{aligned} \tag{B.1.4}$$

where the kinetic variables (Riemann invariants) $\mathcal{F}$ and $\mathcal{G}$ are defined as

$$U = \mathcal{F} + \mathcal{G}, \qquad V = \sqrt{\lambda}(\mathcal{F} - \mathcal{G}). \tag{B.1.5}$$

The local equilibrium functions (Maxwellians) $\hat{\mathcal{F}}$ and $\hat{\mathcal{G}}$ given by

$$\hat{\mathcal{F}} = \frac{U}{2} + \frac{F(U)}{2\sqrt{\lambda}}, \qquad \hat{\mathcal{G}} = \frac{U}{2} - \frac{F(U)}{2\sqrt{\lambda}}.$$

Note that systems (B.1.4) and (B.1.2) are equivalent such that a discretization of each one of them lead essentially to discretization of the other. For spatial discretization consider control volumes $[x_{i-1/2}, x_{i+1/2}]$ with uniform dimension $\Delta x = x_{i+1/2} - x_{i-1/2}$. Integrating (B.1.2) with respect to x over the control volume and keeping the time t continuous one obtain the following semi-discrete system

$$\begin{aligned} \frac{dU_i}{dt} + \mathcal{D}_x V_i &= 0, \\ \frac{dV_i}{dt} + \lambda_i, \mathcal{D}_x U_i &= -\frac{1}{\varepsilon}\left(V_i - F(U_i)\right), \end{aligned} \tag{B.1.6}$$

where $\Psi_i(t)$, $\Psi = U$ or $\Psi = V$, is the space average of a generic solution Ψ in the cell $[x_{i-1/2}, x_{i+1/2}]$ at time t,

$$\Psi_i(t) = \frac{1}{\Delta x}\int_{x_{i-1/2}}^{x_{i+1/2}} \Psi(x,t)dx.$$

The difference quotient in (B.1.6) is defined as

$$\mathcal{D}_x \Psi_i = \frac{\Psi_{i+1/2} - \Psi_{i-1/2}}{\Delta x},$$

where $\Psi_{i\pm1/2} = \Psi(x_{i\pm1/2}, t)$ are the numerical fluxes at $x = x_{i\pm1/2}$. Since $\mathcal{F}$ and $\mathcal{G}$ travel along constant characteristics with speed $+\lambda$ and $-\lambda$, respectively, upwind reconstructions can be easily applied to them. For example, a first-order upwind scheme yields

$$\mathcal{F}_{i+1/2} = \mathcal{F}_i, \qquad \mathcal{G}_{i+1/2} = \mathcal{G}_{i+1}. \tag{B.1.7}$$

A second order discretization can be reconstructed by incorporating limiters in (B.1.7). Using the Sweby's notation, a second order reconstruction reads

$$\mathcal{F}_{i+1/2} = \mathcal{F}_i + \frac{1}{2}\Delta x \sigma_i^+, \qquad \mathcal{G}_{i+1/2} = \mathcal{G}_{i+1} + \frac{1}{2}\Delta x \sigma_{i+1}^-, \tag{B.1.8}$$

where σ_i^+ and σ_i^- are the slope of $\mathcal{F}$ and $\mathcal{G}$ on the cell $[x_{i-1/2}, x_{i+1/2}]$, respectively. For $\Psi^+ = \mathcal{F}$ and $\Psi^- = \mathcal{G}$, the slopes $\sigma^\pm$ are defined by

$$\sigma_i^\pm = \frac{1}{\Delta x}\left(\Psi_{i+1}^\pm - \Psi_i^\pm\right)\Phi(\theta_i^\pm), \qquad \theta_i^\pm = \frac{\Psi_i^\pm - \Psi_{i-1}^\pm}{\Psi_{i+1}^\pm - \Psi_i^\pm},$$

with $\Phi(\theta)$ defines the van Leer's slope limiter function

$$\Phi(\theta) = \frac{|\theta| + \theta}{1 + |\theta|}.$$

Once $\mathcal{F}_{i+1/2}$ and $\mathcal{G}_{i+1/2}$ are reconstructed using (B.1.7) or (B.1.8), the numerical fluxes $U_{i+1/2}$ and $V_{i+1/2}$ in the relaxation system (B.1.6) are obtained from (B.1.5) as

$$U_{i+1/2} = \mathcal{F}_{i+1/2} + \mathcal{G}_{i+1/2} \qquad \text{and} \qquad V_{i+1/2} = \sqrt{\lambda_i}(\mathcal{F}_{i+1/2} - \mathcal{G}_{i+1/2}).$$

Note that higher order discretization are also applicable. For instance, a third-order reconstruction for numerical fluxes in (B.1.6) based on essentially non-oscillatory method has been studied in [5, 6, 92]. Recently a fifth-order reconstruction has been developed in [5] combining relaxation method with a weighted essentially non-oscillatory reconstruction.

Most relaxation schemes integrate the equation (B.1.6) in time using the implicit-explicit (IMEX) methods, compare [49, 5, 92] among others. In fact, the special structure of the nonlinear terms in (B.1.6) makes it trivial to evolve the flux terms explicitly and the source term implicitly. At the limit ($\varepsilon \longrightarrow 0$), the IMEX methods for (B.1.2) reduced to an explicit time integration of the original conservation law (B.1.1) based on the explicit scheme in IMEX methods. For simplicity of presentation, only the relaxed scheme ($\varepsilon = 0$) are formulated. Therefore, with Δt being the time step and Ψ^n denoting the approximate solution of a generic function Ψ at $t = n\Delta t$, the implementation of a second order relaxed scheme to solve (B.1.6) can be carried out in the following two steps:

$$\begin{aligned} U_i^* &= U_i^n - \Delta t \mathcal{D}_x V_i^n\Big|_{V_i^n = F(U_i^n)}, \\ U_i^{**} &= U_i^* - \Delta t \mathcal{D}_x V_i^*\Big|_{V_i^* = F(U_i^*)}, \\ U_i^{n+1} &= \frac{1}{2}(U_i^n + U_i^{**}). \end{aligned} \tag{B.1.9}$$

It is worth remarking that, since the advective part in (B.1.6) is treated explicitly, the time stepping (B.1.9) is conditionally stable such as the time step Δt has to satisfy the CFL condition

$$\text{CFL} = \max_i \lambda_i \frac{\Delta t}{\Delta x} \leq 1, \tag{B.1.10}$$

Remark B.1.3. *In general, the characteristic speeds λ_i can be chosen large enough such that the sub-characteristic condition (B.1.3) is satisfied. However, the numerical diffusion increases with their values and for accuracy reasons λ_i should be chosen as small as possible.*

Bibliography

[1] Adimurthi, J. Jaffre, G. D. V. Gowda, *Godunov-type methods for conservation laws with flux function discontinuous in space*, SIAM J. Numer. Anal., 42(1):179-208, (2004).

[2] L. Armijo, *Minimization of functions having Lipschitz continuous first partial derivatives*, Pacific J. Math., 16:1-3, (1966).

[3] A. Aw, M. Rascle, *Resurrection of second order models of traffic flow?*, SIAM J. Appl. Math., 60:916-938, (2000).

[4] M. K. Banda, M. Herty, A. Klar, *Coupling conditions for gas networks governed by the isothermal Euler equations*, Networks and Heterogeneous Media, 1(1):41-56, (2006).

[5] M. K. Banda, M. Seaïd, *Higher order relaxation schemes for hyperbolic systems of conservation laws*, J. Numer. Math., 13:171–196, (2005).

[6] M. K. Banda, M. Seaïd, *Relaxation WENO schemes for multi-dimensional hyperbolic systems of conservation laws*, preprint.

[7] N. Bellomo, M. Delitala, V. Coscia, *On the mathematical theory of vehicular traffic flow I. Fluid dynamic and kinetic modelling*, Math. Mod. Meth. Appl. Sci., 12(12):1801-1843, (2002).

[8] D. P. Bertsekas, *Necessary and sufficient conditions for a penalty method to be exact*, Math. Progamming, 9(1):87-99, (1975).

[9] D. P. Bertsekas, *Constrained optimization and Lagrange multiplier methods*, Academic press, (1982).

[10] D. P. Bertsekas, *Projected Newton methods for optimization problems with simple constraints*, SIAM J. Control Opt., 20(2):221-246, (1982).

[11] J. F. Bonnans, J. C. Gilbert, C. Lemarechal, C. A. Sagastizabal, *Numerical optimization - theoretical and practical aspects*, Springer (2002).

[12] A. Bressan, *Hyperbolic systems of conservation laws*, Oxford University Press, Oxford, UK, (2000).

[13] J. V. Burke, *An exact penalization viewpoint of constrained optimization*, SIAM J. Control Opt., 29:968-998, (1991).

[14] R. H. Byrd, J. Nocedal, R. Schnabel, *Representations of Quasi-Newton matrices and their use in limited memory methods*, Math. Programming, 63:129-156, (1994).

[15] R. H. Bryd, P. Lu, J. Nocedal and C. Zhu, *A limited memory algorithm for bound constrained optimization*, SIAM J. Sci. Comput., 16:1190-1208, (1995).

[16] R. E. Chandler, R. Herman, E. W. Montroll, *Traffic dynamics: Studies in car following*, Oper. Res., 6(2):165-184, (1958).

[17] C. Charalambous, *A lower bound for the controlling parameters of the exact penalty functions*, Math. Programming, 15:278-290, (1978).

[18] C. Chen, O. L. Mangasarian, *Smoothing methods for convex inequalities and linear complementarity problems*, Math. Programming, 71:51-69, (1995).

[19] C. Chen, O. L. Mangasarian, *A class of smoothing functions for nonlinear and mixed complementarity problems*, Comput. Optim. Appl., 5:97-138, (1996).

[20] B. Chen, P. T. Harker, *A non-interior point continuation method for linear complementarity problems*, SIAM J. Matrix Anal. Appl., 14(4):1168-1190, (1993).

[21] E. Chong, S. H Zȧk, *An introduction to optimization*, Wiley-Interscience Series in Discrete Mathematics and Optimization, (1996).

[22] G. M. Coclite, M. Garavello, B. Piccoli, *Traffic flow on a road network*, SIAM J. Math. Anal., 36(6):1862-1886, (2002).

[23] T. F. Coleman, A. R. Conn, *Second order conditions for an exact penalty function*, Math. Prog., 19:178-185, (1980).

[24] A. R. Conn, *Constrained optimization using a nondifferentiable penalty function*, SIAM. J. Numer. Anal., 10(4):760-784, (1973).

[25] A. R. Conn, N. I. M. Gould, P. L. Toint, *Trust-Region Methods*, MPS/SIAM Series on Optimization, SIAM Philadelphia, (2000).

[26] A. R. Conn, J. W. Sinclair, *Quadratic programming via a nondifferentiable penalty function*, University of Waterloo, Report CORR 75/15.

[27] C.F. Daganzo, *Requiem for second order fluid approximations of traffic flow*, Transp. Res. B, 29:277-286, (1995).

[28] C. M. Dafermos, *Polygonal approximations of solutions of the initial value problem for a conservation law*, J. Math. Anal. Appl., 38:33-41, (1972).

[29] S. Diehl, *A conservation law with point source and discontinuous flux function*, SIAM J. Math. Anal., 56:388-419, (1996).

[30] R. Felkel, *On a bound constrainted optimization technique using second order information*, TU Darmstadt, PhD Thesis, (1999).

[31] A. Fügenschuh, M. Herty, A. Klar, A. Martin, *Combinatorial and continuous models for the optimization of traffic flow networks*, to appear in SIAM J. Opt., (2006).

[32] D. C. Gazis, R. Herman, R. Rothery, *Nonlinear follow the leader models of traffic flow*, Oper. Res., 9(4):545-567, (1961).

[33] T. Gimse and N. H. Risebro, *Solution to the Cauchy problem for a conservation law with a discontinuous flux function*, SIAM J. Math. Anal., 23:635-648, (1992).

[34] G. Golub, C. Van Loan, *Matrix computations*, 3rd ed., Johns Hopkins University press Baltimore, MD, (1996).

[35] H. Greenberg, *An analysis on traffic flow*, Oper. Res., 7:79-85 (1959).

[36] M. Günther, A. Klar, T. Materne, R. Wegener, *An explicitly solvable kinetic model for vehicular and associated macroscopic equations*, Math. Comp. Modelling, 35:591-606, (2002).

[37] S. P. Han, O. L. Mangasarian, *Exact penalty function in nonlinear programming*, Math. Programming, 17:251-269, (1979).

[38] D. Helbing, *Improved fluid dynamic model for vehicular traffic*, Phys. Rev. E, 51:3164-3171, (1995).

[39] ——, *Gas-kinetic derivation of Navier Stokes like traffic equation*, Phys. Rev. E, 53:2366-2381, (1996).

[40] ——, *Verkehrsdynamik*, Springer-Verlag, Berlin, Heidelberg, New York, (1997).

[41] D. Helbing, A. Greiner, *Modeling and simulation of multilane traffic flow*, Phys. Rev. E, 55(5):5498-5508, (1997).

[42] M. Herty, *Mathematics of traffic flow networks.*, PhD Thesis, TU Darmstadt, (2004).

[43] M. Herty, A. Klar, *Modelling, simulation and optimization of traffic flow networks*, SIAM J. Sci. Comp., 25(3):1066-1087, (2003).

[44] M. Herty, A. Klar, *Simplified dynamics and optimization of large scale traffic networks*, Math. Mod. Meth. Appl. Sci., 14(4):579-601, (2004).

[45] H. Holden, N. Risebro, *A mathematical model of traffic flow on a network of unidirectional road*, SIAM J. Math. Anal., 26(4):999-101, (1995).

[46] R. Illner, A. Klar, M. Materne, *Vlasov-Fokker-Planck models for multilane traffic flow*, Comm. Math. Sci., 1:1-12, (2003).

[47] O. Jahn, R. Möhring, A. S. Schulz, *Optimal routing in traffic flows with length restrictions in networks with congestions*, in Operations Research Proceedings, Berlin, (2000).

[48] Silvia M. H. Janesch, L. T. Santos, *Exact penalty methods with constrained subproblems*, Investigacion Operativa, 7(1-2):55-65, (1997).

[49] S. Jin, Z. Xin, *The relaxation schemes for systems of conservation laws in arbitrary space dimensions*, Comm. Pure Appl. Math., 48:235-277, (1995).

[50] K. H. Karlsen, C. Klingenberg, N. H. Risebro, *A relaxation scheme for conservation laws with discontinuous coefficient*, Math. Comp., 73:1235-1259, (2003).

[51] C.T. Kelley, *Iterative methods for optimization*, SIAM Philadelphia, (1999).

[52] B. S. Kerner, P. Konhäuser, *Cluster effect in initially homogeneous traffic flow*, Phys. Rev. E, 48(4):2335-2338, (1993).

[53] A. Klar, M. Guenther, T. Materne, R. Wegener, *Multivalued fundamental diagrams and stop and go waves for continuum traffic flow equations*, SIAM J. Appl. Math. 64(2):468-483, (2003).

[54] A. Klar, R. Illner, H. Lange, A. Unterreiter, R. Wegener, *A Kinetic model for vehicular traffic: Existence of stationary solutions*, J. Math. Anal. Appl., 237:622-643, (1999).

[55] A. Aw, A. Klar, T. Materne, M. Rascle, *Derivation of continuum traffic flow models from microscopic follow-the-leaders models*, SIAM, J. Appl. Math., 63(1): 259-278, (2002).

[56] A. Klar, R, Kühne and R. Wegener, *Mathematical models for vehicular traffic*, Surv. Math. Ind., 6:215-239, (1995).

[57] A. Klar, R. Wegener, *Enskog-like kinetic models for vehicular traffic*, J. Stat. Phys., 87:91-114, (1997).

[58] A. Klar, R. Wegener, *A hierarchy of models for multilane vehicular traffic I: Modeling*, SIAM J. Appl. Math., 59(3):983-1001, (1999).

[59] A. Klar, R. Wegener, *A hierarchy of models for multilane vehicular traffic I: Numerical investigations*, SIAM J. Appl. Math., 59(3):983-1001, (1999).

[60] A. Klar, R. Wegener, *Kinetic derivation of macroscopic anticipation models for vehicular traffic*, SIAM J. Appl. Math., 60:1749-1766, (2000).

[61] C. Klingenberg, N. H. Risebro, *Convex conservation laws with discontinuous coefficients – existence, uniqueness and asymptotic behaviour*, Comm. Partial Differential Equations, 20:1959-1990, (1995).

[62] R. Kühne, *Macroscopic freeway model for dense traffic*, in 9th Int. Symp. on Transportation and Traffic Theory, N. Vollmuller, ed., (1984).

[63] J. E. Lagnese, G. Leugering, *Domain decomposition methods in optimal control of partial differential equations*, volume 148 of International Series of Numerical Mathematics, Birkhäuser Verlag, Basel, (2004).

[64] J. P. Lebacque, *Les Mmodeles Macroscopiques du Traffic*, Annales des Ponts., 67:24-45, (1993).

[65] J. Lebacque, M. Khoshyaran, *First order macroscopic traffic flow models for networks in the context of dynamic assignment*, in Transportation Planning-State of the Art, M. Patriksson and K. A. P. M. Labbe, eds., (2002).

[66] Y. J. Lee, O. L. Mangasarin, *SSVM: A smooth support vector machine for classfication*, COAP, 20:5-22, (2001).

[67] R.J. LeVeque, *Numerical methods for conservation laws*, Birkhäuser-Verlag, (1990).

[68] M.J. Lighthill, G.B. Whitham, *On kinematic waves: II. A theory of traffic flow on long crowded roads*, Proc. Roy. Soc. London, Ser. A, 229:317-345, (1955).

[69] G. Lube, L. Müller, and F. C. Otto, *A non-overlapping domain decomposition method for the advection-diffusion problem*, Computing, 64(1):49-68, (2000).

[70] K. Madsen, H. B. Nielsen, *A finite smoothing algorithm for linear l_1 estimation*, SIAM J. Opt., 3(2):223-235, (1993).

[71] A. May, *Traffic Flow Fundamentals*, Prentice Hall, Englewood Cliffs, NJ, (1990).

[72] D. Q. Mayne, N. Maratos, *A first order exact penalty function algorithm for equality constrained optimization problems*, Math. Programming, 16:303-324, (1979).

[73] D.Q. Mayne, E. Polak, *A superlinearly convergent algorithm for constrained optimization problems*, Math. Programming Stud., 16 (1979).

[74] S. Mishra, *Convergence of upwind finite difference schemes for a scalar consveration law with indefinite discontinuities in the flux function*, SIAM J. Numer. Anal., 43(2):559-577, (2005).

[75] S. Mochon, *An analysis of the traffic on highways with changing surface conditions*, Math. Model., 9(1):1-11, (1987).

[76] J.J. More and G. Toraldo, *On the resolution of large quadratic programming problems with bound constraints*, SIAM J. Opt., 1:93-113, (1991).

[77] T. Nagatani, *Spreading of traffic jam in a traffic flow model*, J. Phys. Soc. Japan, 62(4):1085-1088, (1993).

[78] K. Nagel, M. Schreckenberg, *A cellular automaton model for freeway traffic*, J. Phys. I France, 2(12):2221-2229, (1992).

[79] B. N. Parlett, *The symmetric eigenvalue problem*, SIAM Classics in Applied Mathematics, (1997).

[80] S. L. Paveri-Fontana, *On Boltzmann like treatments for traffic flow*, Transp. Res. 9:225-235, (1975).

[81] H. Payne, *FREFLO: A macroscopic simulation model for freeway traffic*, Transp. Res. Rec., 722:68-75, (1979).

[82] G. Di. Pillo, L. Grippo, *A new class of augmented Lagrangians in nonlinear programming*, SIAM J. Control Opt., 17(5):618-628 (1979).

[83] G. DI. Pillo, L. Grippo, *On the exactness of a class of nondifferentiable penalty functions*, J. Optim. Theo. Appl., 57(3):399-410, (1988).

[84] G. Di. Pillo, L. Grippo, *Exact penalty functions in constrained optimization*, SIAM J. Control Opt., 27(6):1333-1360, (1989).

[85] M. C. Pinar, S. A. Zenios, *On smoothing exact penalty functions for convex constrained optimization*, SIAM J. Opt., 4(3):486-511, (1994).

[86] I. Prigogine, F. C. Andrews, *A Boltzmann like approach for traffic flow*, Oper. Res., 8(6):789-797, (1960).

[87] I. Prigogine, *A Boltzmann like approach to the statistical theory of traffic flow*, In R. Herman, editor, Theory of traffic flow, (1961).

[88] A. Quarteroni, A. Valli, *Domain decomposition methods for partial differential equations*, Oxford University Press, Oxford, (1999).

[89] P. I. Richards, *Shock waves on the highway*, Oper. Res., 4:42-51, (1956).

[90] W. Rudin, *Principles of mathematical analysis*, McGraw-Hill, New York, (1953).

[91] M. Seaïd, *Non-Oscillatory relaxation methods for the shallow water equations in one and two space dimensions*, Int. J. Num. Meth. Fluids, 46:457-484, (2004).

[92] M. Seaïd, *Stable numerical methods for conservation laws with discontinuous flux function*, Appl. Math. Comput., 175:383-400, (2006).

[93] P. Spellucci, *Numerische verfahren der nichtlinearen optimierung*, Birkhäuser Verlag, Basel, Boston, Berlin, (1993).

[94] P. Spellucci, *Solving QP problems by penalization and smoothening*, Preprint (2002).

[95] PL2 is avaible at *http://plato.la.asu.edu/topics/problems/nlounres.html*

[96] J. D. Towers, *Convergence of a difference scheme for conservation laws with discontinuous flux*, SIAM J. Math. Anal., 38:681-698, (2000).

[97] R. Wegener, A. Klar, *A kinetic model for vehicular traffic derived from a stochastic microscopic model*, Transp. Theory Stat. Phys., 25(7):785-798, (1996).

[98] I. Zang, *A smoothing-out technique for min-max optimization*, Math. Programming, 19:61-77, (1980).

[99] P. Zhang and R. X. Liu, *Generalization of Runge-Kutta discontinuous Galerkin method to LWR traffic flow model with inhomogeneous road conditions*, Numer. Methods for PDEs, 21:80-88, (2005).

[100] W. I. Zhangwill, *Nonlinear programming*, Prentice-Hall, (1969).

[101] C. Zhu, R. Byrd, J. Lu, J. Nocedal, *L-BFGS-B: Fortran subroutines for large scale bound constrained optimization*, Tech. Report, NAM-11 EECS Dapartment Nortwestern University, (1994).

Curriculum Vitae

28 August 1978	Born in Assam (Tezpur), India.
1983 – 1993	Primary and Senior School, India
1993 – 1995	Senior Secondary School, India
1995 – 1998	B.Sc.(Mathematics,Physics,Computer-Science), India
1998 – 2000	M.Sc. (Mathematics), Osmania University, India.
2001 – 2003	M.Tech. (Industrial Mathematics), IIT Madras, India.
May 2003 – 2006	PhD (Department of Mathematics), TU-Kaiserslautern, Germany.