PANM 20

Lukáš Vacek; Václav Kučera

Construction of fluxes at junctions for the numerical solution of traffic flow models on networks

In: Jan Chleboun and Pavel Kůs and Petr Přikryl and Miroslav Rozložník and Karel Segeth and Jakub Šístek (eds.): Programs and Algorithms of Numerical Mathematics, Proceedings of Seminar. Hejnice, June 21-26, 2020. Institute of Mathematics CAS, Prague, 2021. pp. 149–158.

Persistent URL: `http://dml.cz/dmlcz/703110`

Programs and Algorithms of Numerical Mathematics 20
J. Chleboun, P. Kůs, P. Přikryl, M. Rozložník, K. Segeth, J. Šístek (Eds.)
Institute of Mathematics of the Czech Academy of Sciences, Prague 2021

CONSTRUCTION OF FLUXES AT JUNCTIONS FOR THE NUMERICAL SOLUTION OF TRAFFIC FLOW MODELS ON NETWORKS

Lukáš Vacek, Václav Kučera

Faculty of Mathematics and Physics, Charles University
Sokolovská 49/83, 186 00, Prague, Czech Republic
lvacek@karlin.mff.cuni.cz, kucera@karlin.mff.cuni.cz

Abstract: We deal with the simulation of traffic flow on networks. On individual roads we use standard macroscopic traffic models. The discontinuous Galerkin method in space and explicit Euler method in time is used for the numerical solution. We apply limiters to keep the density in an admissible interval as well as prevent spurious oscillations in the numerical solution. To solve traffic networks, we construct suitable numerical fluxes at junctions. Numerical experiments are presented.

Keywords: traffic flow, discontinuous Galerkin method, junctions, numerical flux

MSC: 65M60, 76A30, 90B20

1. Introduction

Let us have a road and an arbitrary number of cars. We would like to model the movement of cars on our road. We call this model a traffic flow model. There are two main ways how to describe traffic flow. The first way is the *microscopic model*. Microscopic models describe every car and we can specify the behaviour of every driver and type of car. The basic microscopic models are described by ordinary differential equations. The second approach is the *macroscopic model*. In that case, we view our traffic situation as a continuum and study the density of cars in every point of the road. This model is described by partial differential equations.

Our aim is to numerically solve macroscopic models of traffic flow. Our unknown is density at point x and time t. As we shall see later, the solution can be discontinuous. Due to the need for discontinuous approximation of density, we use the discontinuous Galerkin method. The aim of modelling is understanding traffic dynamics and deriving possible control mechanisms for traffic.

DOI: 10.21136/panm.2020.15

2. Macroscopic traffic flow models

We consider traffic flows on networks, described by macroscopic models, cf. [4, 6]. Here the traffic flow is described by three fundamental quantities – *traffic flow* $Q(x,t)$ which determines the number of cars per second at the position x at time t; *traffic density* $\rho(x,t)$ determines the number of cars per meter at x and t; and the *mean traffic flow velocity* $V(x,t) = Q(x,t)/\rho(x,t)$.

Greenshields described a relation between traffic density and traffic flow in [3]. He realised that traffic flow is a function depending only on traffic density in homogeneous traffic (traffic with no changes in time and space). This implies that even the mean traffic flow velocity depends only on traffic density. The relationship between the traffic density and the mean traffic flow velocity or traffic flow is described by the *fundamental diagram*, cf. [3].

Since the number of cars is conserved, the basic governing equation is a first order hyperbolic partial differential equation, cf. [4]:

$$\frac{\partial}{\partial t}\rho(x,t) + \frac{\partial}{\partial x}\left(\rho(x,t)V(x,t)\right) = 0. \tag{1}$$

Equation (1) must be supplemented by the initial condition

$$\rho(x,0) = \rho_0(x) \quad \text{and} \quad V(x,0) = V_0(x), \qquad x \in \mathbb{R}.$$

and an inflow boundary condition.

We have only one equation for two unknowns. Thus, we need an equation for $V(x,t)$. One possibility is the *Lighthill–Whitham–Richards model* (abbreviated LWR) where we use the *equilibrium velocity* $V_e(\rho)$. There are many different proposals for the equilibrium velocity derived from real traffic data, e.g. Greenshields model takes $V_e(\rho) = v_{\max}\left(1 - \frac{\rho}{\rho_{\max}}\right)$, where $v_{\max}$ is the maximal velocity and $\rho_{\max}$ is the maximal density. The corresponding *equilibrium traffic flow* is $Q_e(\rho) = \rho V_e(\rho)$. Thus we get the following nonlinear first order hyperbolic equation for ρ:

$$\rho_t + \left(\rho V_e(\rho)\right)_x = 0, \qquad x \in \mathbb{R},\ t > 0. \tag{2}$$

2.1. Junctions

Following [2], we consider a complex *network* represented by a directed graph. The graph is a finite collection of directed edges, connected together at vertices. Each vertex has a finite set of incoming and outgoing edges. In our case it is sufficient to study our problem only at one vertex and on its adjacent edges.

On each road (edge) we consider the LWR model, while at junctions (vertices) we consider a *Riemann solver*. At each vertex J, there is a *traffic–distribution matrix* A describing the distribution of traffic among outgoing roads. Let J be a fixed vertex with n incoming and m outgoing edges. Then

$$A = \begin{bmatrix} \alpha_{n+1,1} & \cdots & \alpha_{n+1,n} \\ \vdots & \vdots & \vdots \\ \alpha_{n+m,1} & \cdots & \alpha_{n+m,n} \end{bmatrix}, \tag{3}$$

where for all $i \in \{1,\dots,n\}, j \in \{n+1,\dots,n+m\}$: $\alpha_{j,i} \in [0,1]$ and for all $i \in \{1,\dots,n\}$: $\sum_{j=n+1}^{n+m} \alpha_{j,i} = 1$. The i^{th} column of A describes how traffic from an incoming road I_i distributes to outgoing roads at the junction J. We denote the endpoints of road I_i as a_i, b_i, one of which coincides with J. We introduce the notation of spatial limits $u^{(L)}(a,t) := \lim_{x\to a_-} u(x,t)$ and $u^{(R)}(a,t) := \lim_{x\to a_+} u(x,t)$.

Let $\rho = (\rho_1,\dots,\rho_{n+m})^T$ be a *weak solution at the junction* J, see [2, Definition 5.1.8, page 98], where ρ has bounded variation in space. Then ρ satisfies the *Rankine–Hugoniot condition*, which represents the conservation of cars at the junction:

$$\sum_{i=1}^{n} Q_e(\rho_i^{(L)}(b_i,t)) = \sum_{j=n+1}^{n+m} Q_e(\rho_j^{(R)}(a_j,t)) \tag{4}$$

for almost every $t > 0$ at the junction J, cf. [2, Lemma 5.1.9, page 98].

In [2], the authors define an *admissible weak solution of (2)* related to the matrix A at the junction J as $\rho = (\rho_1,\dots,\rho_{n+m})^T$ satisfying

1) ρ is a weak solution at the junction J such that $\rho_i(\cdot,t)$ is of bounded variation for every $t \geq 0$, i.e. the Rankine–Hugoniot condition holds.

2) $Q_e(\rho_j^{(R)}(a_j,\cdot)) = \sum_{i=1}^{n} \alpha_{j,i} Q_e(\rho_i^{(L)}(b_i,\cdot)),\ \forall j = n+1,\dots,n+m.$

3) $\sum_{i=1}^{n} Q_e(\rho_i^{(L)}(b_i,\cdot))$ is a maximum subject to 1) and 2).

Assumption 1) is the conservation of cars at the junction. Assumption 2) takes into account the prescribed preferences of drivers how the traffic from incoming roads is distributed to outgoing roads according to fixed coefficients. Assumption 3) postulates that drivers choose to maximize the total flux through the junction. In Section 4.1, we define an alternative approach and compare to that of [2].

3. Discontinuous Galerkin method

As an appropriate method for the numerical solution of (2), we choose the *discontinuous Galerkin* (DG) method, which is essentially a combination of finite volume and finite element techniques, cf. [1]. We expect discontinuities, so the finite element method is not suitable. At the same time, DG is a higher order method which approximates smooth solutions better than the finite volume method.

Consider an interval $\Omega = (a,b)$. Let $\mathcal{T}_h$ be a partition of $\overline{\Omega}$ into a finite number of intervals (elements) $K = [a_K,\ b_K]$. We denote the set of all boundary points of all elements by $\mathcal{F}_h$. Let $p \geq 0$ be an integer. We seek the numerical solution in the space of discontinuous piecewise polynomial functions

$$S_h = \{v;\ v|_K \in P^p(K),\ \forall K \in \mathcal{T}_h\},$$

where $P^p(K)$ denotes the space of all polynomials on K of degree at most p. For a function $v \in S_h$ we denote the *jump* in the point s as $[v]_s = v^{(L)}(s) - v^{(R)}(s)$.

We formulate the DG method for the general first order hyperbolic problem

$$\begin{aligned} u_t + f(u)_x &= g, && x \in \Omega,\ t \in (0,T),\\ u &= u_D, && x \in \mathcal{F}_h^D,\ t \in (0,T),\\ u(x,0) &= u_0(x), && x \in \Omega, \end{aligned}$$

where g, u_D and u_0 are given functions and u is our unknown. The Dirichlet boundary condition is prescribed only on the inlet $\mathcal{F}_h^D \subseteq \{a,b\}$, respecting the direction of information propagation (characteristics).

The DG formulation then reads, cf. [1]: Find $u_h : [0,T] \to S_h$ such that

$$\int_\Omega (u_h)_t \varphi \,\mathrm{d}x - \sum_{K\in\mathcal{T}_h} \int_K f(u_h)\varphi_x \,\mathrm{d}x + \sum_{s\in\mathcal{F}_h} H(u_h^{(L)}, u_h^{(R)})\,[\varphi]_s = \int_\Omega g\varphi \,\mathrm{d}x,$$

for all $\varphi \in S_h$. In the boundary terms on $\mathcal{F}_h$ we use the approximation $f(u_h) \approx H(u_h^{(L)}, u_h^{(R)})$, where H is a *numerical flux*. We use the *Godunov flux*, cf. [5]:

$$H(u_h^{(L)}, u_h^{(R)}) = \begin{cases} \min_{u_h^{(L)} \le u \le u_h^{(R)}} f(u), & \text{if } u_h^{(L)} < u_h^{(R)},\\ \max_{u_h^{(R)} \le u \le u_h^{(L)}} f(u), & \text{if } u_h^{(L)} \ge u_h^{(R)}. \end{cases} \tag{5}$$

4. Implementation

For time discretization of the DG method we use the *explicit Euler method*. As a basis for S_h, we use *Legendre polynomials*. We use *Gauss–Legendre quadrature* to evaluate integrals over elements. The implementation is in the C++ language.

Because we calculate physical quantities (density and velocity), the result must be in some interval, e.g. $\rho \in [0, \rho_{\max}]$. Thus, we use *limiters* in each time step to obtain the solution in the admissible interval. Here it is important not to change the total number of cars. Following [5], we also apply limiting to treat spurious oscillations near discontinuities and sharp gradients in the numerical solution.

4.1. Numerical fluxes at junctions

Since we wish to model traffic on networks, the numerical fluxes at junctions must be specified. The basic requirement is that the number of cars at the junctions must be conserved. Moreover, we wish to prescribe the traffic distribution according to the traffic–distribution matrix (3).

At the junction, we consider an incoming road I_i and an outgoing road I_j. If these roads were the only roads at the junction, i.e. if they were directly connected to each other, the (numerical) flux of traffic from I_i to I_j would simply be $H\big(\rho_{hi}^{(L)}(b_i,t), \rho_{hj}^{(R)}(a_j,t)\big)$, where ρ_{hi} and ρ_{hj} are the DG solutions on I_i and I_j, respectively. From the traffic distribution matrix, we know the ratios of the traffic flow distribution to the outgoing roads. Thus, we take the numerical flux $H_j(t)$ at the left point of the outgoing road I_j, i.e. at the junction, at time t as

$$H_j(t) := \sum_{i=1}^{n} \alpha_{j,i} H\big(\rho_{hi}^{(L)}(b_i,t), \rho_{hj}^{(R)}(a_j,t)\big), \tag{6}$$

for $j = n+1, \dots, n+m$. The numerical flux $H_j(t)$ can be viewed as the DG analogue of taking the combined traffic outflow $\sum_{i=1}^{n} \alpha_{j,i} Q_e\big(\rho_i^{(L)}(b_i,t)\big)$ from all incoming roads and prescribing it as the inflow of traffic to the road I_j.

Similarly, we take the numerical flux $H_i(t)$ at the right point of the incoming road I_i, i.e. at the junction, at time t as

$$H_i(t) := \sum_{j=n+1}^{n+m} \alpha_{j,i} H\big(\rho_{hi}^{(L)}(b_i,t), \rho_{hj}^{(R)}(a_j,t)\big), \tag{7}$$

for $i = 1, \dots, n$. Again, this can be viewed as an analogue of the traffic flow $\sum_{j=n+1}^{n+m} \alpha_{j,i} Q_e\big(\rho_j^{(R)}(a_j,t)\big)$ being prescribed as the outflow of traffic from I_i.

It can be shown, that our choice of numerical fluxes conserves the number of cars at junctions, similarly as in (4). However, this choice does not distribute the traffic according to the traffic–distribution matrix (3) exactly, only approximately.

Theorem 1 (Properties of the solution)**.** *Let us use the method described above.*

a) Our solution ρ_{hi}, $i = 1, \dots, n+m$ satisfies the discrete analogue to the Rankine–Hugoniot condition (4)*:*

$$\sum_{i=1}^{n} H_i(t) = \sum_{j=n+1}^{n+m} H_j(t).$$

b) There exists an example such that our solution ρ_{hi}, $i = 1, \dots, n+m$, does not satisfy the property 2) *in Section* 2.1*.*

Proof. a) From the definition of H_i and H_j, we immediately obtain

$$\begin{aligned} \sum_{i=1}^{n} H_i(t) &= \sum_{i=1}^{n} \sum_{j=n+1}^{n+m} \alpha_{j,i} H(\rho_{hi}^{(L)}(b_i,t), \rho_{hj}^{(R)}(a_j,t)) \\ &= \sum_{j=n+1}^{n+m} \sum_{i=1}^{n} \alpha_{j,i} H(\rho_{hi}^{(L)}(b_i,t), \rho_{hj}^{(R)}(a_j,t)) = \sum_{j=n+1}^{n+m} H_j(t). \end{aligned}$$

b) Let us take the situation with one incoming and two outgoing roads. We want to show that $H_2(\cdot) \neq \alpha_{2,1} H_1(\cdot)$ or $H_3(\cdot) \neq \alpha_{3,1} H_1(\cdot)$. Assume the Riemann problem (cf. [2, Definition 4.2.1, page 72]) with $\rho_{h1}^{(L)}(b_1,0) = 0.5$, $\rho_{h2}^{(R)}(a_2,t) = 0.2$,

$\rho_{h3}^{(R)}(a_3,t) = 0.6$, $\alpha_{2,1} = 0.25$ and $\alpha_{3,1} = 0.75$. We use the Greenshields model (with $v_{\max} = \rho_{\max} = 1$) and the Godunov flux (5). Then

$$H_2(0) = \alpha_{2,1} H(\rho_{h1}^{(L)}(b_1,0), \rho_{h2}^{(R)}(a_2,0)) = 0.0625$$

and

$$H_1(0) = \alpha_{2,1} H(\rho_{h1}^{(L)}(b_1,0), \rho_{h2}^{(R)}(a_2,0)) + \alpha_{3,1} H(\rho_{h1}^{(L)}(b_1,0), \rho_{h3}^{(R)}(a_3,0)) = 0.2425.$$

Since $H_2(0) = 0.0625 \neq 0.060625 = \alpha_{2,1} H_1(0)$, we find an example, where the property 2) in Section 2.1 in not satisfied. □

A method how to obtain an admissible solution satisfying properties 1)–3) in Section 2.1 is described in [2] or [7]. As an example, we take a junction with one incoming and two outgoing roads. In [2, 7], maximum possible fluxes are used. If there is a traffic jam in one of the outgoing roads, the maximum possible flow through the junction satisfying the distribution condition 2) is 0, thus the whole junction is blocked by a traffic jam in one of the outgoing roads. On the other hand, the cars in our approach can still go into the second outgoing road according to the traffic-distribution coefficients. So our choice of numerical fluxes corresponds to modelling turning lanes, which allow the cars to separate before the junction according to their preferred turning direction. In our case the junction is not blocked due to a traffic jam on one of the outgoing roads. Since macroscopic models are intended for long (multi-lane) roads with huge numbers of cars, our model makes sense in this situation. The original approach from [2, 7] works for one-lane roads, where splitting of the traffic according to preference is not possible.

Another difference is that we can use all varieties of traffic lights. The model of [2, 7] can use only the so-called full green lights. Our approach gives us an opportunity to change the lights for each direction separately.

An artifact of our model is that sometimes we do not satisfy the traffic–distribution coefficients exactly. This corresponds to the real situation where some cars decide to use another road instead of staying in the traffic jam. For these reasons we interpret the matrix A as a *traffic–preference matrix*. This is also analogous to the DG method, where even for smooth exact solutions, the approximate solution is discontinuous on $\mathcal{F}_h$ with a small error corresponding to small jumps in the solution. Also Dirichlet conditions are not satisfied exactly. Both Dirichlet boundary conditions and continuity are enforced in a weak sense using penalization. Similarly, the traffic distribution is enforced in a weaker sense via the traffic-preference matrix.

5. Numerical results

In this section we present two numerical results on a simple model network. As we mention above, we use the combination of the explicit Euler method and DG method. We can compare our results with the approach in the paper [7] where the authors use the maximum possible fluxes. We calculate the piecewise linear approximations of solutions and we use two Gaussian quadrature points in each element.

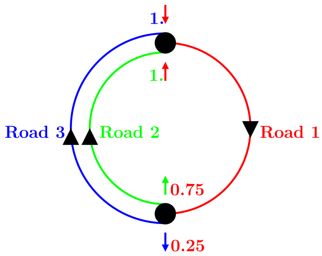


Figure 1: Test network with Road 1, Road 2 and Road 3.

5.1. Simple network

Now we demonstrate how our program computes traffic on networks. We define the simple network from Fig. 1. This network is closed, so we can show the conservation of the total number of cars. We have three roads and two junctions. The length of all roads is 1. At the first junction we have one incoming and two outgoing roads. At the second junction we have the opposite situation. We use a different distribution of cars at the first junction: $\frac{3}{4}$ go from the first road to the second and $\frac{1}{4}$ from the first road to the third. This corresponds to the traffic-preference matrices $A_1 = [0.75, 0.25]^T$ and $A_2 = [1, 1]$.

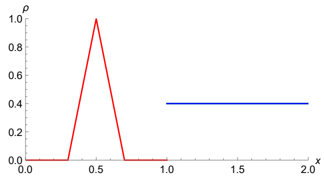
We define different initial conditions for each road. The initial condition for the first road as a piecewise linear "hump" which is defined by

$$\rho_{0,1}(x) = \begin{cases} 5x - 1.5, & x \in [0.3, 0.5], \\ -5x + 3.5, & x \in [0.5, 0.7], \\ 0, & \text{otherwise}, \end{cases}$$

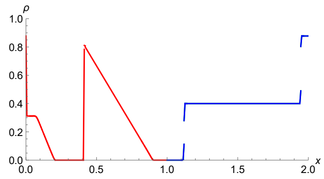
while the second and third road has a constant density of 0.4, cf. Fig. 2a. The total number of cars in the whole network is 1. We use the Greenshields model on all roads. We use the Euler method with the step size $\tau = 10^{-4}$ and the number of elements is $N = 150$ on each road. Since our primary aim is the solution at junctions and resolution of shocks, we choose a very small time step in order to avoid the time error and satisfy the CFL condition with a large margin. Our aim has not been computational costs yet, we chose such τ and N which give us stability in a wide range of examples.

We can see the results in Fig. 2. Road 1 distributes the traffic density between the other roads. We have too many cars at the second junction, where we have two incoming roads. Thus, we create a traffic congestion on Road 2 and Road 3. We can observe the transport and the distribution of the jump from the first road through the junction in Fig. 2g and Fig. 2h. The result converges to a stationary solution. The traffic density in Fig. 2i is close to the stationary solution. The amount of cars is conserved.

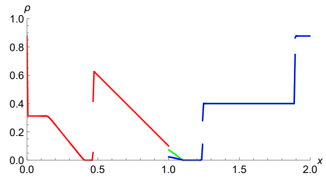
We note that our program can compute traffic on bigger networks and we are not limited by the number of incoming or outgoing roads at junctions.



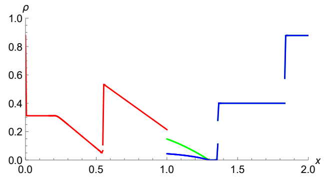
(a) $t = 0$.



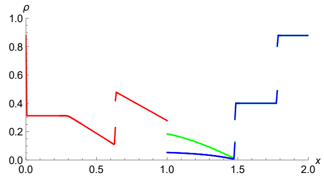
(b) $t = 0.2$.



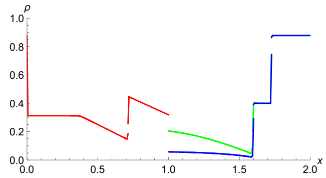
(c) $t = 0.4$.



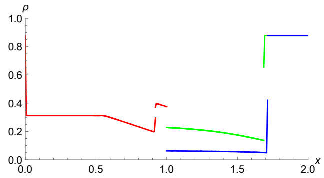
(d) $t = 0.6$.



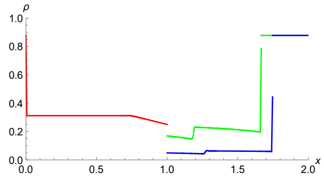
(e) $t = 0.8$.



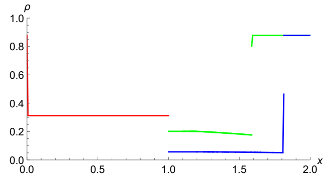
(f) $t = 1$.



(g) $t = 1.5$.



(h) $t = 2$.



(i) $t = 3$.

Figure 2: Traffic density on network from Fig. 1 – Road 1, Road 2 and Road 3.

5.2. Simple network – comparison with the maximum possible fluxes

We consider the same network as in Section 5.1 with different initial conditions:

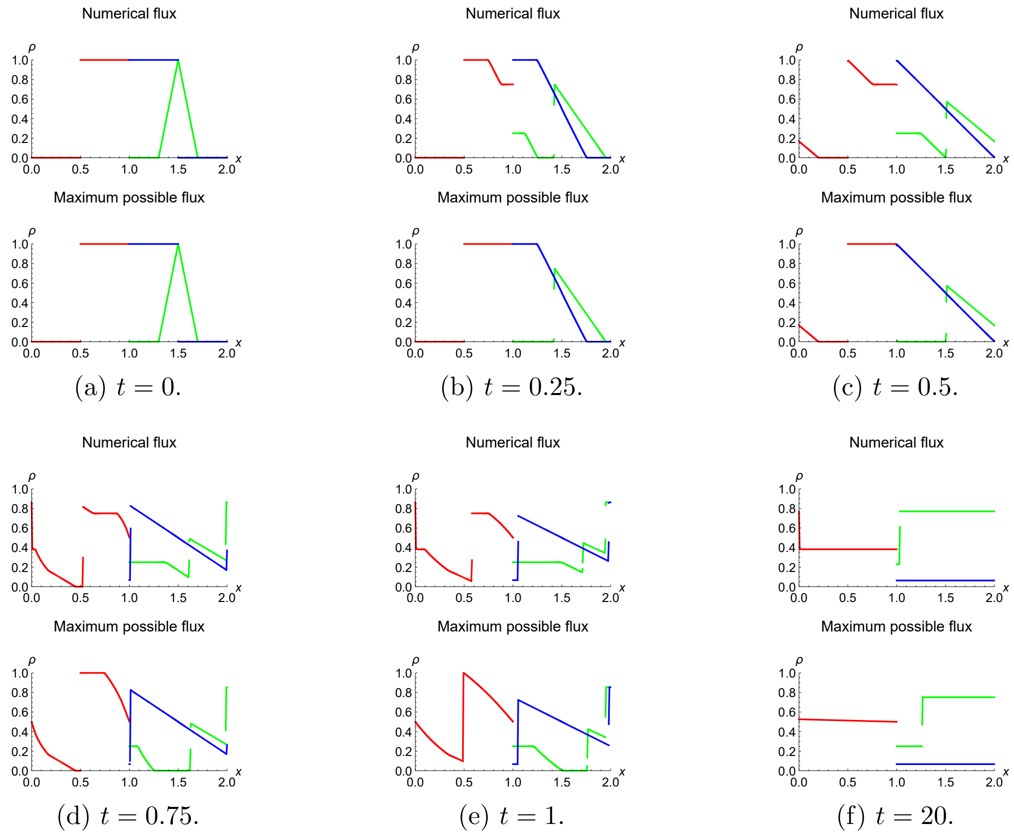
$$\widetilde{\rho}_{0,1}(x) = \begin{cases} 0, & x \in [0, 0.5], \\ 1, & x \in [0.5, 1], \end{cases} \quad \widetilde{\rho}_{0,2}(x) = \rho_{0,1}(x) \quad \widetilde{\rho}_{0,3}(x) = \begin{cases} 1, & x \in [0, 0.5], \\ 0, & x \in [0.5, 1], \end{cases}$$

where $\widetilde{\rho}_{0,i}$ is the initial condition on road number i.

We compare our approach with that of [7] which uses the maximum possible flux. In both approaches we use the Godunov flux and the explicit Euler method. A right of way parameter q must be prescribed for the junction with two incoming roads in the case of the maximum possible flux, cf. [2, Section 5.2.2]. We use $q = 0.5$, so the roads are equal. In our approach, we do not have a defined right of way (in the sense of yielding rules at main or side roads), so the roads are equal as well.

We can see the comparison in Figure 3. Our approach is in the top row while the approach using the maximum possible flux is in the bottom row. We point out the different behavior in both junctions.

First, we notice the first junction with one incoming and two outgoing roads, i.e. $x = 1$ in the figures. As we mention in Section 4.1, the maximum possible flux through the junction at the time $t \in [0, 0.5]$ is zero because one of the outgoing roads (Road 3) reaches the maximal traffic density, cf. Figure 3b and 3c. Our approach has nonzero traffic flow through this junction at the time $t \in [0, 0.5]$ because



(a) $t = 0$. (b) $t = 0.25$. (c) $t = 0.5$.

(d) $t = 0.75$. (e) $t = 1$. (f) $t = 20$.

Figure 3: Comparison of network with Road 1, Road 2 and Road 3

the numerical flux is nonzero between Road 1 and Road 2 allowing the cars to go from Road 1 to Road 2. For times $t > 0.5$, the maximal traffic density is not attained on Road 3 and the traffic flow is nonzero through the junction in both cases, cf. Figure 3d, 3e and 3f. If we compare both approaches, we see completely different results on Roads 1 and 2 while the results on Road 3 are almost identical.

Now we focus on the second junction with two incoming and one outgoing road, i.e. $x = 0$ and $x = 2$ in the figures. At first glance, there is no difference between the two approaches. Let's compare $\rho_1^{(R)}(0, 1)$, i.e. the limit from the right of traffic density on the outgoing Road 1 at $x = 0$ and $t = 1$. Our approach gives us $\rho_1^{(R)}(0, 1) \approx 0.4$ while the approach using the maximum possible flux gives us $\rho_1^{(R)}(0, 1) \approx 0.5$, which is the maximal traffic flow. The reason for this difference is that we do not have a defined right of way in our approach. Road 2 and Road 3 push too many cars into the junction congesting it slightly. The approach using the maximum possible flux takes into account the whole situation and selects the best solution for both roads. From a real point of view, this approach could be viewed as simulating the behavior of communicating autonomous vehicles which optimize the traffic situation globally, while our approach could be interpreted as simulating the behavior of human drivers without the right of way. We note that both approaches converge to stationary solutions which are not identical, see Figure 3f.

We would like to implement right of way into our approach and introduce it in future work.

6. Conclusion

We have demonstrated the numerical solution of macroscopic traffic flow models using the discontinuous Galerkin method. For the approximation in time we choose explicit Euler methods. For traffic networks, we construct special numerical fluxes at the junctions. The use of DG methods on networks is not standard. We have described the differences between our approach and the paper [7] by Čanić, Piccoli, Qiu and Ren, where the maximum possible flow at the junction is used.

Acknowledgements

The work of L. Vacek is supported by the Charles University, project GA UK No. 1114119. The work of V. Kučera is supported by the Czech Science Foundation, project No. 20–01074S.

References

[1] Dolejší, V. and Feistauer, M.: *Discontinuous Galerkin method – analysis and applications to compressible flow.* Springer, Heidelberg, 2015.

[2] Garavello, M. and Piccoli, B.: *Traffic flow on networks*, vol. 1. American Institute of Mathematical Sciences (AIMS), Springfield, MO, 2006.

[3] Greenshields, B. D.: A study of traffic capacity. Highway Research Board **14** (1935), 448–477.

[4] Kachroo, P. and Sastry, S.: *Traffic flow theory: mathematical framework.* University of California Berkeley. In preparation, available online: https://www.scribd.com/doc/316334815/Traffic-Flow-Theory, accessed: 2020-12-15.

[5] Shu, C. W.: Discontinuous Galerkin methods: general approach and stability. Numerical Solutions of Partial Differential Equations **201** (2009).

[6] van Wageningen-Kessels, F., van Lint, H., Vuik, K., and Hoogendoorn, S.: Genealogy of traffic flow models. EURO Journal on Transportation and Logistics **4** (2015), 445–473.

[7] Čanić, S., Piccoli, B., Qiu, J., and Ren, T.: Runge-Kutta discontinuous Galerkin method for traffic flow model on networks. Journal of Scientific Computing **63** (2014), 31.