Proceedings of ALGORITMY 2020
pp. 41–50

TRAFFIC FLOW MODEL ON NETWORKS USING NUMERICAL FLUXES AT THE JUNCTIONS*

LUKÁŠ VACEK† AND VÁCLAV KUČERA‡

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

Key words. Traffic flow, discontinuous Galerkin method, junctions, numerical flux

AMS subject classifications. 65M60, 76A30

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 (ODEs). 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 (PDEs).

Our aim is to numerically solve macroscopic models of traffic flow. Our unknown is density at point x and at 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.

2. Macroscopic traffic flow models.

We consider traffic flow on networks, described by macroscopic models, cf. [1, 2]. 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

*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.

†Faculty of Mathematics and Physics, Charles University, Prague, Czech Republic (lvacek@karlin.mff.cuni.cz).

‡Faculty of Mathematics and Physics, Charles University, Prague, Czech Republic.

hyperbolic partial differential equation, cf. [2]:

$$\frac{\partial}{\partial t}\rho(x,t) + \frac{\partial}{\partial x}\left(\rho(x,t)V(x,t)\right) = 0. \tag{2.1}$$

Equation (2.1) must be supplemented by the initial condition

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

and the 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 the 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 equations equation for ρ:

$$\rho_t + \left(\rho V_e(\rho)\right)_x = 0, \qquad x \in \mathbb{R},\ t > 0. \tag{2.2}$$

2.1. Junctions. Following [4], we study 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. 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

$$A = \begin{bmatrix} \alpha_{n+1,1} & \cdots & \alpha_{n+1,n} \\ \vdots & \vdots & \vdots \\ \alpha_{n+m,1} & \cdots & \alpha_{n+m,n} \end{bmatrix}, \tag{2.3}$$

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.

Let $\rho = (\rho_1,\dots,\rho_{n+m})^T$ be a *weak solution at the junction* J, see [4, Definition 5.1.8, page 98], such that each $x \to \rho_i(x,t)$ has bounded variation. Then ρ satisfies the *Rankine–Hugoniot condition*, which represents the conservation of cars at the junction:

$$\sum_{i=1}^{n} Q_e(\rho_i(b_{i-},t)) = \sum_{j=n+1}^{n+m} Q_e(\rho_j(a_{j+},t)) \tag{2.4}$$

for almost every $t > 0$ at the junction J, where $\rho_j(a_{j+},t) := \lim_{(x\to a_j+)} \rho_j(x,t)$ and $\rho_i(b_{i-},t) := \lim_{(x\to b_i-)} \rho_i(x,t)$, cf. [4, Lemma 5.1.9, page 98].

Finally, $\rho = (\rho_1,\dots,\rho_{n+m})^T$ is called an *admissible weak solution of (2.2)* related to the matrix A at the junction J if the following properties hold:

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) $Q_e(\rho_j(a_{j+},\cdot)) = \sum_{i=1}^{n} \alpha_{j,i} Q_e(\rho_i(b_{i-},\cdot)),\ \forall j = n+1,\dots,n+m.$

3) $\sum_{i=1}^{n} Q_e(\rho_i(b_{i-},\cdot))$ 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) describes the behaviour that drivers choose so as to maximize the total flux through the junction.

3. Discontinuous Galerkin method. As an appropriate method for the numerical solution of (2.2), we choose the *discontinuous Galerkin* (DG) method, which is essentially a combination of finite volume and finite element techniques, cf. [5]. We consider a 1D domain $\Omega = (a, b)$. Let $\mathcal{T}_h$ be a partition of $\overline{\Omega}$ into a finite number of closed intervals (elements) $[a_K,\ b_K]$. 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 use the notation: $v^{(L)}(x) = \lim_{y\to x-} v(y)$, $v^{(R)}(x) = \lim_{y\to x+} v(y)$ and $[v]_x = v^{(L)}(x) - v^{(R)}(x)$.

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

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

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. [5]: Find $u_h : [0,T] \to S_h$ such that

$$\int_\Omega (u_h)_t \varphi \,\mathrm{d}x - \sum_{K\in\mathcal{T}_h} \int_K f(u_h)\varphi' \,\mathrm{d}x + \sum_{x\in\mathcal{F}_h} H(u_h^{(L)}, u_h^{(R)})\,[\varphi]_x = \int_\Omega g\varphi \,\mathrm{d}x,$$

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 *Lax–Friedrichs flux*, cf. [5]: We define $\alpha = \max_{u\in(u_h^{(L)},u_h^{(R)})} |f'(u)|$. Then we calculate the numerical flux as

$$H(u_h^{(L)}, u_h^{(R)}) = \frac{1}{2}\left(f(u_h^{(L)}) + f(u_h^{(R)}) - \alpha(u_h^{(R)} - u_h^{(L)})\right). \tag{3.1}$$

In practice, we approximate α by calculating $|f'(u)|$ at the points $u_h^{(L)}$, $u_h^{(R)}$ and $\frac{1}{2}(u_h^{(L)} + u_h^{(R)})$ and we take the maximal value.

4. Implementation. For time discretization of the DG method we use *explicit Euler method*. As a basis for S_h, we use *Legendre polynomials* and 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), we know that the result must be in some interval, e.g. $[0, \rho_{\max}]$. 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. For a piecewise linear approximation of ρ in LWR models, we find each element K for which there exists $x \in [a_K, b_K]$ such that $\rho(x) \notin [\rho_{\min}, \rho_{\max}]$. If the average density on element K is in the admissible interval, we decrease the slope of our solution so that the modified density lies in $[\rho_{\min}, \rho_{\max}]$. If the average density on element K is not in the admissible interval $[\rho_{\min}, \rho_{\max}]$ we decrease the time step. Following [6], 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 (2.3). The number of cars which inflow or outflow through the junction is given by the traffic flow Q_e. More precisely, the traffic flow from incoming road I_i, $i = 1, \ldots, n$, at time t is given by $Q_e\left(\rho_i(b_{i-}, t)\right)$. Due to the traffic–distribution matrix, we know the ratio of the traffic flow distribution between the outgoing roads. Thus, the traffic flow to the outgoing road I_j, $j = n+1, \ldots, n+m$, at time t is given by $Q_e\left(\rho_j(a_{j+}, t)\right) = \sum_{i=1}^{n} \alpha_{j,i} Q_e\left(\rho_i(b_{i-}, t)\right)$. Since the traffic flow at the boundary of an element is represented by the numerical flux, we take the numerical flux $H_j(t)$ at the left point of the outgoing road I_j, i.e. the point at the junction, at time t as

$$H_j(t) := \sum_{i=1}^{n} \alpha_{j,i} H(\rho_{hi}(b_{i-}, t), \rho_{hj}(a_{j+}, t)),$$

for $j = n+1, \ldots, n+m$, where ρ_{hi} is the DG solution on the i^{th} road. The numerical flux $H_j(t)$ approximates the traffic flow $Q_e\left(\rho_j(a_{j+}, t)\right)$. Similarly, we take the numerical flux $H_i(t)$ at time t at the right point of the incoming road I_i, i.e. at the junction point, as

$$H_i(t) := \sum_{j=n+1}^{n+m} \alpha_{j,i} H(\rho_{hi}(b_{i-}, t), \rho_{hj}(a_{j+}, t)),$$

where $i = 1, \ldots, n$. Then $H_i(t)$ approximates the traffic flow $Q_e\left(\rho_i(b_{i-}, t)\right)$.

It can be shown, that our choice of numerical fluxes conserves the number of cars at junctions. However, this choice does not distribute the traffic according to the traffic–distribution matrix (2.3) exactly, only approximately.

THEOREM 4.1 (Properties of the solution). *Let us use the method described above.*

a) Our solution ρ_{hi}, $i = 1, \ldots, n+m$ satisfies the Rankine-Hugoniot condition (2.4).

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

Proof. a) In our case, we want to show

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

From the definition of H_i and H_j, we immediately obtain

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

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 that $\rho_{h1}(b_{1-},0) = 0.5$, $\rho_{h2}(a_{2+},t) = 0.2$, $\rho_{h3}(a_{3+},t) = 0$, $\alpha_{2,1} = 0.75$ and $\alpha_{3,1} = 0.25$. We use the Greenshields model (with $v_{\max} = \rho_{\max} = 1$) and the Lax-Friedrichs flux (3.1). Then

$$H_2(0) = \alpha_{2,1}H(\rho_{h1}(b_{1-},0),\rho_{h2}(a_{2+},0)) = 0.22125$$

and

$$H_1(0) = \alpha_{2,1}H(\rho_{h1}(b_{1-},0),\rho_{h2}(a_{2+},0)) + \alpha_{3,1}H(\rho_{h1}(b_{1-},0),\rho_{h3}(a_{3+},0)) = 0.315.$$

Since $H_2(0) = 0.2212 \neq 0.23625 = \alpha_{2,1}H_1(0)$, 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 [4] or [7]. As an example, we take a junction with one incoming and two outgoing roads. In [4, 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 is 0. 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 the macroscopic models are aimed for long (multi–lane) roads with huge number of cars, our model makes sense in this situation. The original approach from [4, 7] is aimed 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 [4, 7] can use only the full green lights. Our approach gives us an opportunity to change the lights for each direction separately.

An artefact of our model is that 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. The problem is when there is no traffic jam. Since we do not control the traffic–distribution exactly, we do not satisfy it exactly. For this reason we interpret the matrix A as a *traffic–preference matrix*. Now the element $\alpha_{j,i}$ is the preference that the cars want to go from the incoming road I_i to outgoing road I_j.

5. Numerical results. In this section we present our program and numerical results. As we mention above, we use the combination of the explicit Euler and DG methods. We show the result of calculation on a bottleneck and on networks.

5.1. Bottleneck. First we demonstrate results for a single road with a bottleneck, cf. Fig. 5.1. In the sector 1 and 4 we have maximal velocity $v_{\max,1} = 1.3$

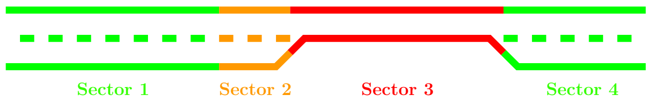
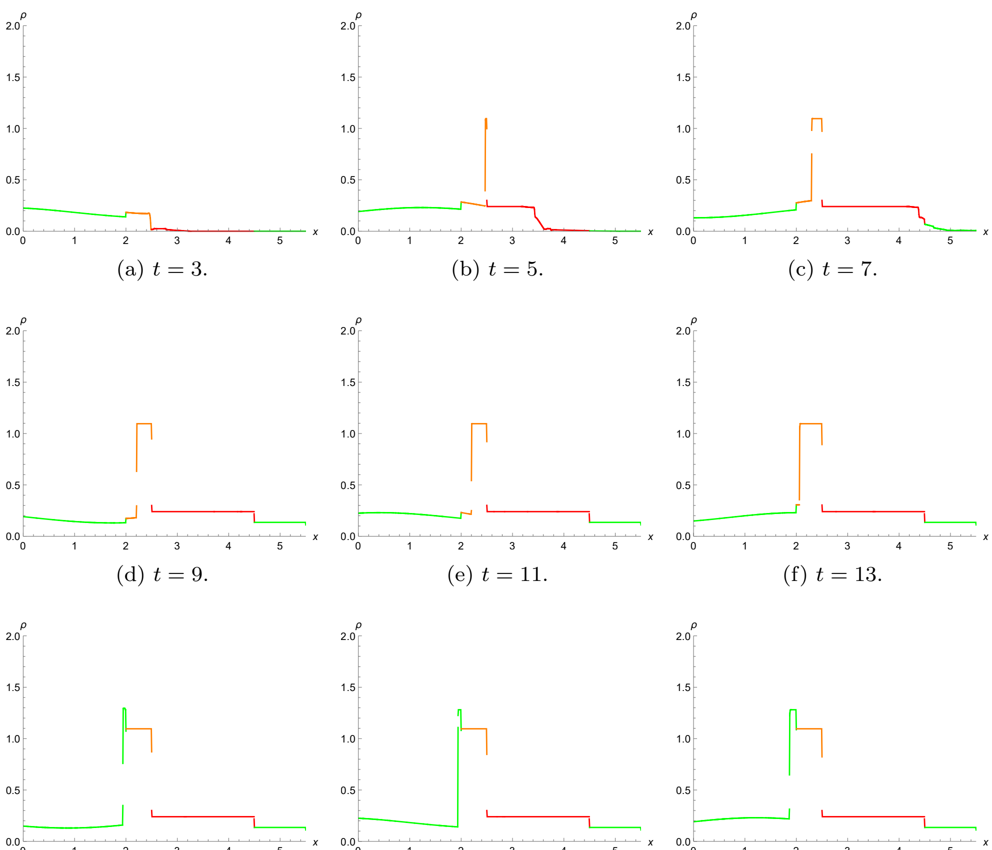


Fig. 5.1: Test road with a bottleneck in Sector 3.

and maximal density $\rho_{\max,1} = 2$, which corresponds to two lanes and a 130 km/h speed limit. The length of the first sector is $L_1 = 2$ (kilometres) and the length of the fourth sector is $L_4 = 1$. Sector 2 is a short sector with length $L_2 = 0.5$ and with decreased maximal velocity $v_{\max,2} = 1$ and maximal density $\rho_{\max,2} = 2$. Sector 3 is the bottleneck, where the maximal density is $\rho_{\max,3} = 1$, which corresponds to one line. The maximal velocity is $v_{\max,3} = 0.8$ and the length of this sector is $L_3 = 2$.

The cars go from left to right. The boundary condition on the left is $\rho(0,t) = \frac{1}{20}\sin\left(\frac{2\pi t}{7} - \frac{\pi}{2}\right) + 0.18$ to simulate time–varying traffic. The initial condition is an empty road. We use the Greenshields model. The time–step size is $\tau = 10^{-4}$ and the



(a) $t = 3$. (b) $t = 5$. (c) $t = 7$.

(d) $t = 9$. (e) $t = 11$. (f) $t = 13$.

(g) $t = 15$. (h) $t = 17$. (i) $t = 19$.

Fig. 5.2: Bottleneck – density on Sector 1, Sector 2, Sector 3 and Sector 4.

length of each element is $h = \frac{1}{150}$.

In Fig. 5.2 we can observe the emergence of a traffic congestion between Sector 2 and Sector 3. The traffic congestion spreads backwards to Sector 1 and becomes longer or shorter depending on the boundary influx. Because $\rho(x,t) < \rho_{\max,i}$ for all x, t and all sectors, the cars in the traffic congestion are still moving.

We can observe the relationship between maximal velocity and traffic density depending on the presence of traffic congestion in Sector 1 and Sector 2. Without traffic congestion, the density in Sector 1 (with higher maximal velocity) is lower than the density in Sector 2, cf. Fig. 5.2a. Conversely, with traffic congestion, the density in Sector 1 is higher than density in Sector 2, cf. Fig. 5.2i. This behaviour is due to the same traffic flow in both sectors.

5.2. Simple network. Now we demonstrate how our program computes traffic on networks. Thus, we define the simple network from Fig. 5.3. 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 road 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

$$\rho_0(x) = \begin{cases} 5x - 1.5, & x \in [0.3, 0.5], \\ -5x + 3.5, & x \in [0.5, 0.7], \\ 0, & \text{otherwise}, \end{cases}$$

while the second and third road has a constant density of 0.4, cf. Fig. 5.4a. 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 $N = 100$ on each road.

We can see the results in Fig. 5.4. 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 transporting and the distribution of the jump from the first road through the junction in Fig. 5.4g and Fig. 5.4h. The result converges to a stationary solution. The traffic density in Fig. 5.4i is close to the stationary solution. The amount of cars is conserved.

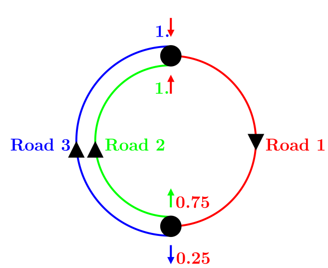
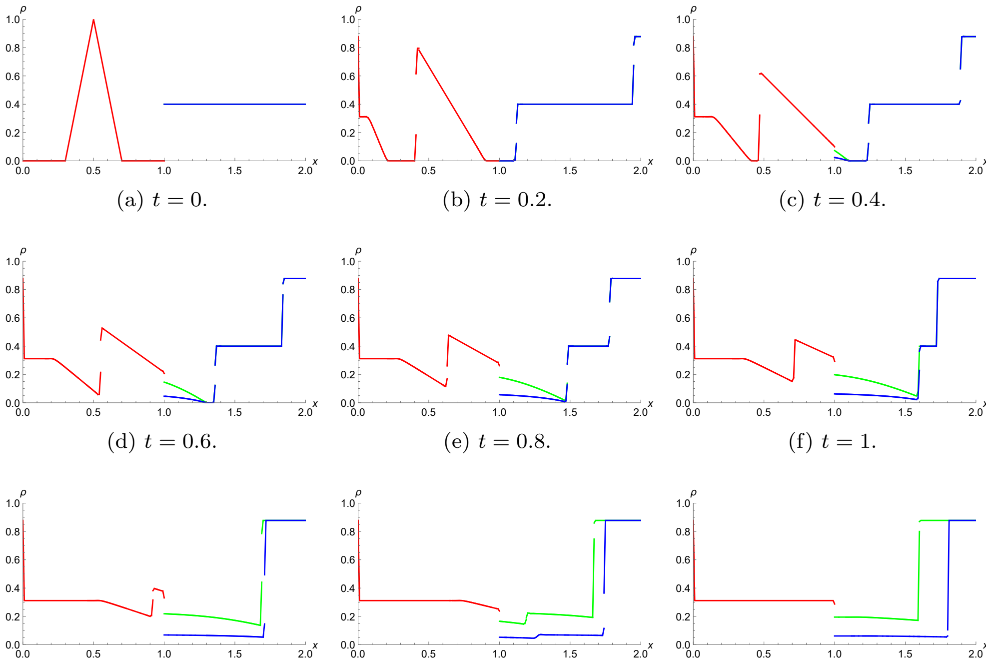


Fig. 5.3: Test network with Road 1, Road 2 and Road 3.



(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) $t = 3$.

Fig. 5.4: Traffic density on network from Fig. 5.3 – Road 1, Road 2 and Road 3.

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.

5.3. Traffic lights. Finally, we demonstrate how our program computes traffic on junctions with traffic lights. We are not strictly forced to use only full green. Our traffic flow at the junction allows us to choose from a large variety of traffic lights.

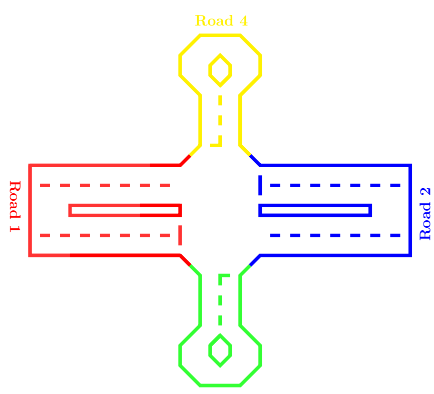
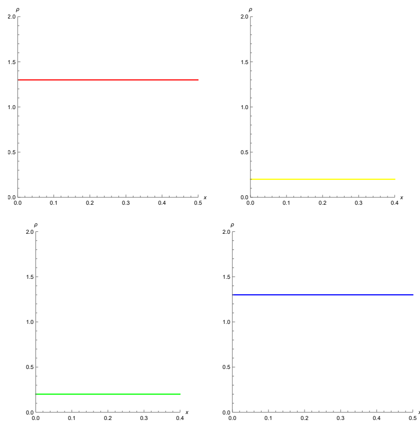
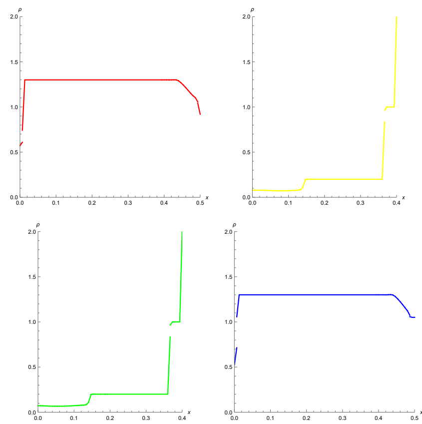


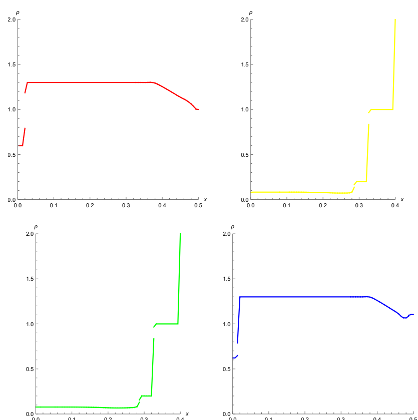
Fig. 5.5: Test network with traffic lights at junction.



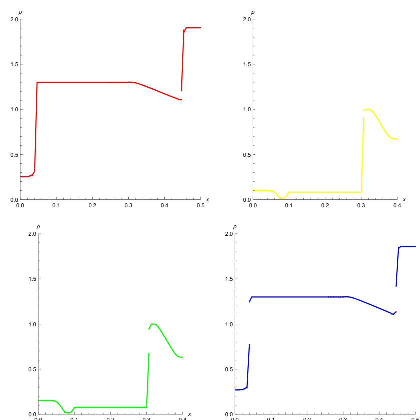
(a) $t = 0$.



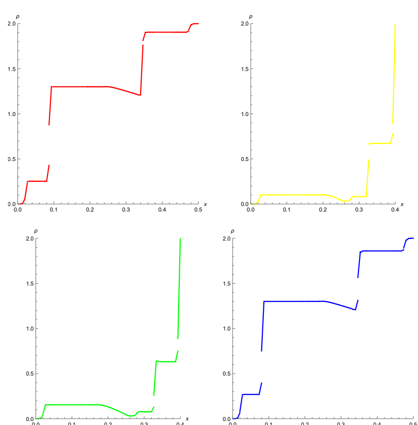
(b) $t = 0.4$.



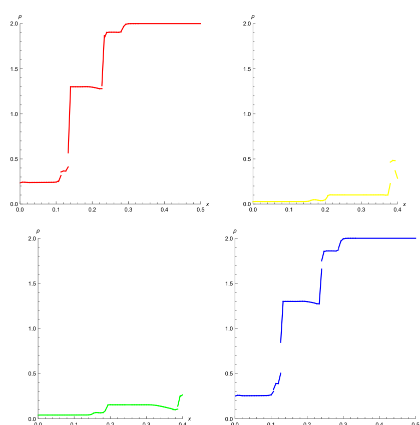
(c) $t = 0.8$.



(d) $t = 1.2$.



(e) $t = 1.6$.



(f) $t = 2$.

Fig. 5.6: Junction with Road 1, Road 2, Road 3 and Road 4.

We define the junction with 4 incoming and 4 outgoing roads, see Fig. 5.5. The outgoing roads turn back and go again to the same junction as the incoming roads. Roads 1 and 2 are the main roads. The maximal density is $\rho_{\max,m} = 2$ and the length is $L_m = 0.5$. The initial condition for the main roads is defined by $\rho_{0,m}(x) = 1.3$. Roads 3 and 4 are the side roads. The maximal density is $\rho_{\max,s} = 1$ and the length is $L_s = 0.4$. The initial condition for the side roads is defined by $\rho_{0,s}(x) = 0.2$.

At the junction we use the traffic–preference matrix

$$A = \begin{bmatrix} 0 & 0.75 & 0.4 & 0.45 \\ 0.8 & 0 & 0.5 & 0.4 \\ 0.1 & 0.15 & 0 & 0.15 \\ 0.1 & 0.1 & 0.1 & 0 \end{bmatrix}.$$

We define three phases for traffic lights. In the first phase, traffic lights allow vehicles from Road 1 to drive to Road 2 or Road 3 and vehicles from Road 2 to drive to Road 1 or Road 4. The first phase lasts $t_1 = 1$. In the second phase, traffic lights allow vehicles from Road 1 to drive to Road 4, vehicles from Road 2 to drive to Road 3, vehicles from Road 3 to drive to Road 2 and vehicles from Road 4 to drive to Road 1. In the third phase, the traffic lights on Road 3 and Road 4 have full green signal. The second and third phase lasts $t_2 = 0.5$. After each phase there are all lights red and this situation lasts $t_r = 0.05$. All three phases are constantly alternating.

The maximal velocity on each roads is $v_{\max} = 0.5$. The maximal density at the junction is $\rho_{\max,j} = 2$. Hence, the maximal density in the first and last elements of both side roads is linearly decreasing and increasing, respectively. We use Greenshields model. The time–step size is $\tau = 10^{-4}$ and the length of each element is $h = \frac{1}{150}$.

We can see the results in Fig. 5.6. We can observe each phase. The first phase is in Fig. 5.6b and Fig. 5.6c. The second phase is in Fig. 5.6d. The third phase is in Fig. 5.6f. There are red lights on each road in Fig. 5.6e.

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.

REFERENCES

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

[2] P. Kachroo and S. Sastry, *Traffic Flow Theory: Mathematical Framework*, University of California Berkeley, `https://www.scribd.com/doc/316334815/Traffic-Flow-Theory` (cited 16^{th} April 2020).

[3] B. D. Greenshields, *A Study of Traffic Capacity*, Highway Research Board, 14 (1935), pp. 448–477.

[4] M. Garavello and B. Piccoli, *Traffic flow on networks*, AIMS Series on Applied Mathematics, 1 (2006), pp. 1–243.

[5] V. Dolejší and M. Feistauer, *Discontinuous Galerkin Method – Analysis and Applications to Compressible Flow*, Analysis and Applications to Compressible Flow, 48 (2015).

[6] C.–W. Shu, *Discontinuous Galerkin methods: general approach and stability*, Numerical solutions of partial differential equations, 201 (2009).

[7] S. Čanić, B. Piccoli, J. Qiu and T. Ren, *Runge–Kutta Discontinuous Galerkin Method for Traffic Flow Model on Networks*, Journal of Scientific Computing, 63 (2014).