



Contents lists available at SciVerse ScienceDirect

# Transportation Research Part B

journal homepage: [www.elsevier.com/locate/trb](http://www.elsevier.com/locate/trb)

## Analytical and grid-free solutions to the Lighthill–Whitham–Richards traffic flow model

Pierre-Emmanuel Mazaré<sup>a</sup>, Ahmad H. Dehwah<sup>b</sup>, Christian G. Claudel<sup>b,\*</sup>, Alexandre M. Bayen<sup>c</sup>

<sup>a</sup> University of California at Berkeley, Transportation Engineering, Department of Civil and Environmental Engineering, Sutardja Hall 642, UC Berkeley, Berkeley, CA 94720-1710, United States

<sup>b</sup> King Abdullah University of Science and Technology (KAUST), Department of Electrical Engineering, Thuwal 23955-6900, Saudi Arabia

<sup>c</sup> University of California at Berkeley, Systems Engineering, Department of Civil and Environmental Engineering, Sutardja Hall 642, UC Berkeley, Berkeley, CA 94720-1710, United States

### ARTICLE INFO

#### Article history:

Received 21 June 2010

Received in revised form 12 July 2011

Accepted 12 July 2011

#### Keywords:

LWR model

Traffic flow

Grid-free numerical scheme

Variational method

### ABSTRACT

In this article, we propose a computational method for solving the Lighthill–Whitham–Richards (LWR) partial differential equation (PDE) semi-analytically for arbitrary piecewise-constant initial and boundary conditions, and for arbitrary concave fundamental diagrams. With these assumptions, we show that the solution to the LWR PDE at any location and time can be computed exactly and semi-analytically for a very low computational cost using the cumulative number of vehicles formulation of the problem. We implement the proposed computational method on a representative traffic flow scenario to illustrate the exactness of the analytical solution. We also show that the proposed scheme can handle more complex scenarios including traffic lights or moving bottlenecks. The computational cost of the method is very favorable, and is compared with existing algorithms. A toolbox implementation available for public download is briefly described, and posted at <http://traffic.berkeley.edu/project/downloads/lwrsolver>.

© 2011 Elsevier Ltd. All rights reserved.

## 1. Introduction

### 1.1. Background

One of the seminal traffic flow models for highways is presented in (Lighthill and Whitham, 1956) and (Richards, 1956), and results in the so called Lighthill–Whitham–Richards (LWR) model or *kinematic wave* theory. Although more sophisticated models of traffic flow are available, the LWR model is widely used to model highway traffic (Newell (1993), Jin (2010) and more recently for urban traffic (Geroliminis and Daganzo, 2008). The LWR *partial differential equation* (PDE) is a first order scalar hyperbolic conservation law that computes the evolution of a density function (the density of vehicles on a road section). This PDE has multiple solutions in general, among which the entropy solution (Ansorge, 1990) is recognized to be the physically meaningful solution.

The LWR PDE can be numerically solved using a variety of computational methods, such as first order numerical schemes, for instance in Godunov (1959) and Daganzo (1994, 1995). Classical numerical methods often require a computational grid, and yield an approximate solution of the PDE. Some exceptions exist however, such as the wave tracking methods, see for instance Henn (2003), Lu et al. (2008), Chen et al. (2009), Yadong et al. (2009), and Wong and Wong (2002). In the present

\* Corresponding author.

E-mail addresses: [mazare@berkeley.edu](mailto:mazare@berkeley.edu) (P.-E. Mazaré), [ahmad.dehwah@kaust.edu.sa](mailto:ahmad.dehwah@kaust.edu.sa) (A.H. Dehwah), [christian.claudel@kaust.edu.sa](mailto:christian.claudel@kaust.edu.sa) (C.G. Claudel), [bayen@ce.berkeley.edu](mailto:bayen@ce.berkeley.edu) (A.M. Bayen).

article, we propose a new algorithm for solving the LWR PDE that does not require a computational grid, and which can be used to compute the solutions to the LWR PDE exactly for any concave fundamental diagram, and for any piecewise constant initial and boundary conditions.

The algorithm presented in this article uses the *cumulative number of vehicles* Newell (1993) function (CVN function) as an intermediate computational abstraction. The CVN function is the integral form of the density function, and solves a Hamilton–Jacobi (HJ) PDE (Daganzo, 2005a, 2006), while the density function itself solves the LWR PDE. As the solution to a Hamilton–Jacobi equation with concave Hamiltonian, the CVN function can be computed at any point by minimizing a functional (or cost function), see Daganzo (2005b) and Aubin et al. (2008) for more details. An approximate minimization can be done numerically for example using *dynamic programming* (DP), which is used in Daganzo (2005a). The present article is also based on the variational method as Daganzo (2005a), but unlike the latter, does not use a dynamic programming approximation for solving the HJ PDE.

We assume that the data used to simulate traffic evolutions is generated by *Eulerian* (fixed) sensors. This measurement data yields initial and boundary conditions, which are here assumed to be piecewise constant. *Lagrangian* data (originating from mobile sensors) can also be used as *internal boundary conditions* (Leclercq et al., 2004). In this article, we use a semi-analytic method to compute the solution to the LWR PDE and the corresponding HJ PDE, for any set of initial, boundary and internal boundary conditions.

To the best knowledge of the authors, only front tracking (Lu et al., 2008; Chen et al., 2009; Yadong et al., 2009; Henn, 2003) and dynamic programming methods (Daganzo, 2005a) can compute the solutions to the LWR PDE exactly, for specific classes of initial/boundary conditions and fundamental diagrams. The proposed method extends these computational methods for situations in which the initial/boundary conditions are piecewise constant and the fundamental diagram is an arbitrary concave function. Unlike the front-tracking methods, our computational method is not event-based, and can compute the solution at any given point without any knowledge or computation of prior events. Unlike dynamic programming methods used to compute the CVN function, our computational method is exact and can also compute the derivatives of the CVN function (that is, the solution to the LWR PDE) exactly, whereas the former method require (inexact) numerical differentiation. Most notably, the proposed method does not require to grid the space-time domain to compute the solution at a given point provided by the user.

Note that this method could also be extended to more complex initial/boundary conditions (for instance piecewise linear (Lu et al., 2008)), at the expense of a greater complexity. Symmetrically, we show in this article that the method can be simplified in the case of triangular fundamental diagrams. Triangular fundamental diagrams are of great importance and relevance for modeling and control applications, see Daganzo (1994, 2005a) for instance.

The rest of this article is organized as follows. Section 2 defines the LWR and HJ PDEs investigated in this article. Section 3 introduces the concept of partial solutions, which are used later as building blocks of the solution to the HJ PDE. In this section, we also show that the partial solutions can be computed analytically for any concave fundamental diagram (smooth or not). We also compute the solution to the LWR PDE on different traffic flow scenarios to illustrate the algorithm's exactness. In particular, we show that the algorithm can be extended to accommodate internal boundary conditions simulating traffic lights or moving bottlenecks. The properties of the solution are investigated in Section 4. Finally, Section 5 presents a fast algorithm specific to triangular fundamental diagrams, for which additional simplifications can be made.

## 2. Modeling

### 2.1. The LWR PDE

We quickly summarize standard material related to the LWR PDE and its connection to the Hamilton–Jacobi PDE through the so-called Moskowitz equation. We consider a one-dimensional, uniform section of highway, limited by  $x_0$  upstream and  $x_n$  downstream. For a given time  $t \in [0, t_m]$  and position  $x \in [x_0, x_n]$ , we define the local traffic density  $k(x, t)$  in vehicles per unit length, and the instantaneous flow  $q(x, t)$  in vehicles per unit time. The conservation of vehicles on the highway is written as follows (Lighthill and Whitham, 1956; Richards, 1956; Garavello and Piccoli, 2006):

$$\forall x, t \in [x_0, x_n] \times [0, t_m], \quad \frac{\partial k(x, t)}{\partial t} + \frac{\partial q(x, t)}{\partial x} = 0 \quad (1)$$

For first order traffic flow models, flow and density are related by the *fundamental diagram*  $Q: (x, t, k(x, t)) \mapsto q(x, t)$ , which is an empirically measured law (Greenshields, 1935). Through this article, we consider the *homogeneous problem* (Daganzo, 2006) in which the fundamental diagram is a function of density  $k$  only, i.e.  $q(x, t, k(x, t)) = Q(k(x, t))$ .

The fundamental diagram is a positive function defined on  $[0, \kappa]$ , where  $\kappa$  is the maximal density (jam density). It ranges in  $[0, q_{\max}]$  where  $q_{\max}$  is the maximum flow (capacity). It is assumed to be differentiable at 0 and  $\kappa$ , with  $Q'(0) = v_f > 0$  the free flow speed, and  $Q'(\kappa) = w < 0$  the congested wave speed (Lighthill and Whitham, 1956). We assume that the fundamental diagram is concave and continuous. Both assumptions are not dictated by physical laws but are required for the mathematical well-posedness of the approach. Non-concave and non-continuous fundamental diagrams are sometimes necessary to model specific traffic patterns (Kerner and Konhäuser, 1994; Edie, 1961) but they require a separate mathematical treatment. Examples of fundamental diagrams satisfying all the above assumptions are shown in Fig. 1.

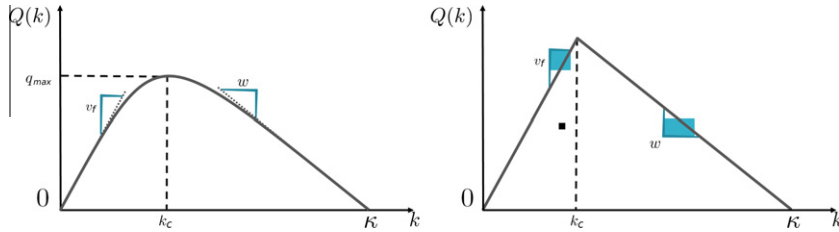


Fig. 1. Left: Generic concave fundamental diagram. Right: Triangular fundamental diagram.

The introduction of the fundamental diagram yields the Lighthill–Whitham–Richards (LWR) PDE:

$$\forall x, t \in [x_0, x_n] \times [0, t_m], \quad \frac{\partial k(x, t)}{\partial t} + \frac{\partial Q(k(x, t))}{\partial x} = 0 \tag{2}$$

### 2.2. The Moskowitz function

The cumulated vehicle count  $N(x, t)$ , also called Moskowitz function (Moskowitz, 1965), represents the continuous vehicle count at location  $x$  and time  $t$ . It has been developed for instance in (Newell, 1993; Daganzo, 2005a, 2006) in the context of transportation engineering, and goes back to (Makigami et al., 1971; Moskowitz, 1965).

In the Moskowitz framework, one assumes that all vehicles are labeled by increasing integers as they pass the entry point  $x_0$  of a highway section, and that they cannot pass each other. If the latest car that passed an observer standing at location  $x$  and time  $t$  is labeled  $n$ , then  $\lfloor N(x, t) \rfloor = n$ . This count function is interpolated continuously between the discrete labels. The Moskowitz function contains traffic information that one can infer from experimental traffic measurements as long as vehicles do not pass each other. In this situation, the isolines of  $N(x, t)$  correspond to vehicle trajectories.

Moreover, local density  $k(x, t)$  and flow  $q(x, t)$  can be computed from the vehicle count using the equalities

$$k(x, t) = - \frac{\partial N(x, t)}{\partial x} \tag{3}$$

$$q(x, t) = \frac{\partial N(x, t)}{\partial t} \tag{4}$$

Introducing the Moskowitz function in (2) yields the Hamilton–Jacobi PDE (Newell, 1993; Daganzo, 2005a, 2006; Claudel and Bayen, 2010a,) in which the fundamental diagram  $Q$  plays the role of Hamiltonian Evans (1998):

$$\frac{\partial N(x, t)}{\partial t} - Q\left(- \frac{\partial N(x, t)}{\partial x}\right) = 0 \tag{5}$$

The above equation can be interpreted as the fundamental diagram (or flow–density relationship) defined in CVN space.

### 2.3. The Cauchy problem

Eq. (5) is a scalar Hamilton–Jacobi partial differential equation, which can be solved using an initial condition function  $N_{ini}(x)$ , an upstream boundary condition function  $N_{up}(t)$  and a downstream boundary condition function  $N_{down}(t)$ . In general, finding such solutions while enforcing arbitrary boundary conditions is impossible with experimental data because the data is not necessarily consistent with the model, and contains measurement errors, leading to ill-posed problems. Weak boundary conditions were introduced in Bardos et al. (1979), Le Floch (1988), and Strub and Bayen (2006) to resolve this problem by integrating situations in which prescribed boundary conditions do not apply. In the context of the Hamilton–Jacobi equation, we can introduce the Cauchy problem:

$$\forall x, t \in [x_0, x_n] \times [0, t_m], \quad \begin{cases} \frac{\partial N(x, t)}{\partial t} - Q\left(- \frac{\partial N(x, t)}{\partial x}\right) = 0 \\ N(x, 0) \stackrel{\triangle}{=} N_{ini}(x) \\ N(x_0, t) \stackrel{\triangle}{=} N_{up}(t) \\ N(x_n, t) \stackrel{\triangle}{=} N_{down}(t) \end{cases} \tag{6}$$

where  $\stackrel{\triangle}{=}$  represents the imposition of a weak boundary condition as developed in Strub and Bayen (2006). Note that weak boundary conditions are related to the concept of demand and supply, see for instance Daganzo (1994, 1995).

2.4. Piecewise affine initial and boundary conditions

For the rest of the article, we use piecewise constant conditions on density and flow, which translate to piecewise-affine conditions on the Moskowitz function. Piecewise constant conditions on density and flow are a natural way to encode discrete measurements in the model, and are used in the Cell Transmission Model (CTM) (Daganzo, 1994). A graphical representation of the studied domain is shown in Fig. 2.

Let  $m$  and  $n \geq 1$  be integers,  $x_0 < x_1 < \dots < x_n$  and  $t_0 < t_1 < \dots < t_m$  the space–time discretization for initial and boundary conditions where  $t_0 = 0$ . We assume that the initial densities  $(k_{ini}^{(i)})_{0 \leq i \leq n-1} \in \mathbb{R}_+^n$ , the upstream flows  $(q_{up}^{(j)})_{0 \leq j \leq m-1} \in \mathbb{R}_+^m$  and the downstream flows  $(q_{down}^{(j)})_{0 \leq j \leq m-1} \in \mathbb{R}_+^m$  are given (known), as in Fig. 2. The initial densities are thus decomposed as piecewise constant in their respective measurement intervals:

$$\forall x \in [x_i, x_{i+1}[, \quad k(x, 0) = k_{ini}^{(i)} \tag{7}$$

and let the upstream and downstream flows also be prescribed as piecewise constant:

$$\forall t \in [t_j, t_{j+1}[, \quad q(x_0, t) = q_{up}^{(j)} \tag{8}$$

$$q(x_n, t) = q_{down}^{(j)} \tag{9}$$

Note that no assumption is made regarding the uniformity of the grid: the spacings  $x_i - x_{i-1}$  and  $t_i - t_{i-1}$  are not necessarily uniform over  $i$ . The method proposed next can handle arbitrary grids and spatio-temporal discretization of the data. However, for notational simplicity, we will not write the method in its full generality in this article. In the toolbox posted online (refer to the link in the abstract), we have coded a general implementation of the method which takes arbitrary measurement intervals  $[x_i, x_{i+1}]$  and  $[t_j, t_{j+1}]$ .

The initial condition of the Moskowitz PDE is obtained by integrating the initial condition of the LWR PDE assuming that  $N_{ini}(x_0) = 0$  and :

$$\forall x \in [x_i, x_{i+1}[, \quad N_{ini}(x) = - \int_{x_0}^x k(\chi, 0) d\chi = - \sum_{m=0}^{i-1} (x_{m+1} - x_m) k_{ini}^{(m)} - (x - x_i) k_{ini}^{(i)} \tag{10}$$

Similarly, the upstream and downstream boundary conditions of the Moskowitz PDE, assuming that  $N_{up}(0) = 0$  and  $N_{ini}(x_n) = N_{down}(0)$  are given by:

$$\forall t \in [t_j, t_{j+1}[, \quad N_{up}(t) = \int_0^t q_{up}(\tau) d\tau = \sum_{m=0}^{j-1} (t_{m+1} - t_m) q_{up}^{(m)} + (t - t_j) q_{up}^{(j)} \tag{11}$$

$$\forall t \in [t_j, t_{j+1}[, \quad N_{down}(t) = N_{ini}(x_n) + \int_0^t q_{down}(\tau) d\tau = N_{ini}(x_n) + \sum_{m=0}^{j-1} (t_{m+1} - t_m) q_{down}^{(m)} + (t - t_j) q_{down}^{(j)} \tag{12}$$

The relationship between initial/boundary conditions in the density space and in the Moskowitz space is illustrated in Fig. 3.

Note that it is possible to set a free downstream boundary condition by not defining the downstream boundary condition, i.e. by skipping the iteration on downstream boundary conditions in Algorithm 1. An example in which the downstream boundary is free is available in Section 3.5.

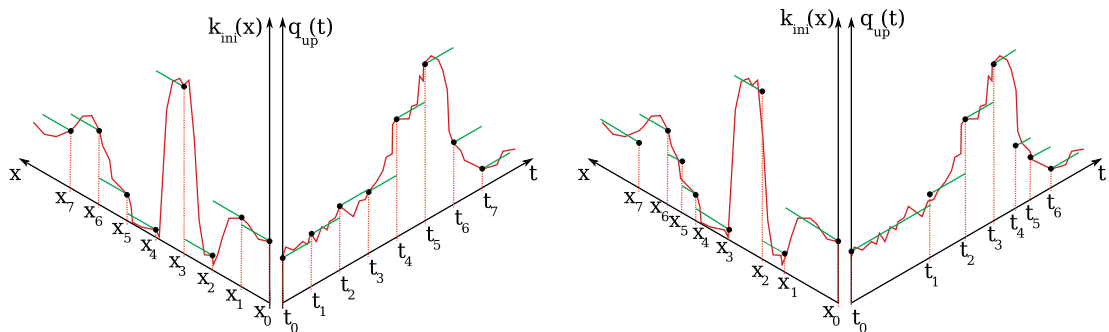
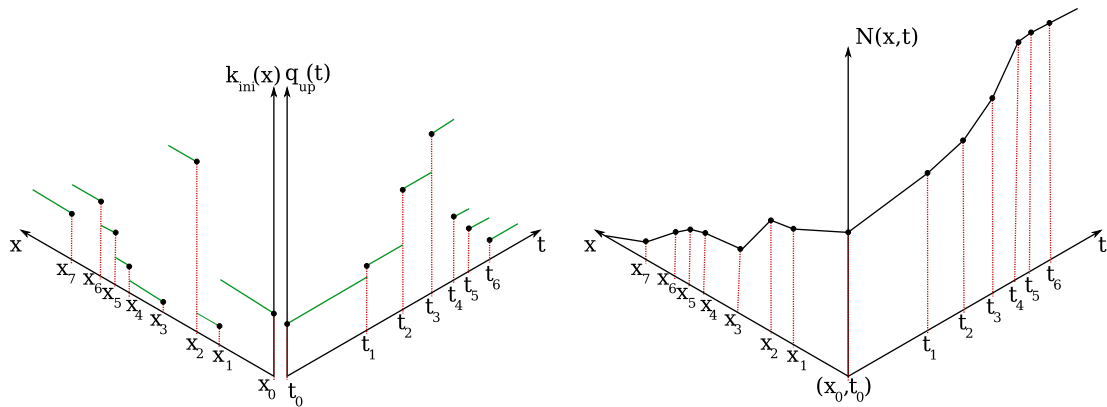


Fig. 2. Different piecewise-constant encodings of the same noisy initial condition  $k_{ini}$  and upstream boundary condition  $q_{up}$ . For clarity, the downstream boundary condition has not been included in this figure. Left: Example of uniformly sampled  $k_{ini}$  and  $q_{up}$ , leading to  $x_i = i\delta x$  and  $t_j = j\delta t$ . Right: Example of unevenly spaced  $x_i$  and  $t_i$ .



**Fig. 3.** Left: Illustration of a piecewise constant initial density and upstream flow conditions. Right: Corresponding piecewise affine Moskowitz function. For clarity, downstream boundary conditions are not shown.

**Algorithm 1.** Pseudo-code implementation for the Lax–Hopf based computation of the Moskowitz function and the associated density at a single point  $(x, t)$  prescribed by the user.

<p><b>Input:</b> <math>x \in [x_0, x_n], t \in [0, t_m]</math>,  <math>i_{up} \leftarrow \min(i - 1)</math> s.t. <math>t_i \geq t - \frac{x-x_0}{v_f}</math>  <math>i_{down} \leftarrow \min(i - 1)</math> s.t. <math>t_i \geq t - \frac{x_0-x}{w}</math>  <math>j_{min} \leftarrow \max(0, \min(j - 1))</math> s.t. <math>x_j \geq x - v_f t</math>  <math>j_{max} \leftarrow \min(m, \max(j + 1))</math> s.t. <math>x_j \leq x - w t</math>  <math>N \leftarrow +\infty</math>  <b>for</b> <math>j = j_{min}</math> <b>to</b> <math>j_{max}</math> <b>do</b>            compute <math>N_{c_{ini}^{(j)}}(x, t)</math> using (21)            <b>if</b> <math>N_{c_{ini}^{(j)}}(x, t) &lt; N</math> <b>then</b>              <math>N \leftarrow N_{c_{ini}^{(j)}}(x, t)</math>              <math>k \leftarrow k_{c_{ini}^{(j)}}(x, t)</math>, computed using (22)            <b>end if</b>  <b>end for</b>  <b>for</b> <math>i = 0</math> <b>to</b> <math>i_{up}</math> <b>do</b>            compute <math>N_{c_{up}^{(i)}}(x, t)</math> using (24)            <b>if</b> <math>N_{c_{up}^{(i)}}(x, t) &lt; N</math> <b>then</b>              <math>N \leftarrow N_{c_{up}^{(i)}}(x, t)</math>              <math>k \leftarrow k_{c_{up}^{(i)}}(x, t)</math>, computed using (25)            <b>end if</b>  <b>end for</b>  <b>for</b> <math>i = 0</math> <b>to</b> <math>i_{down}</math> <b>do</b>            compute <math>N_{c_{down}^{(i)}}(x, t)</math> using (27)            <b>if</b> <math>N_{c_{down}^{(i)}}(x, t) &lt; N</math> <b>then</b>              <math>N \leftarrow N_{c_{down}^{(i)}}(x, t)</math>              <math>k \leftarrow k_{c_{down}^{(i)}}(x, t)</math>, computed using (28)            <b>end if</b>  <b>end for</b> <math>N, k</math></p>	<p>input space domain, time domain            number of components to compute from upstream data            number of components to compute from downstream data            min space index for influencing initial condition            max space index for influencing initial condition            initialization of the Moskowitz function to infinity            iteration on initial conditions            component induced by the initial condition <math>\mathbf{c}_{ini}^{(j)}</math>            if the current component contributes to the solution            update Moskowitz function            compute density              iteration on upstream boundary conditions            component induced by the upstream boundary condition <math>\mathbf{c}_{up}^{(i)}</math>            if the current component contributes to the solution            update Moskowitz function            compute density              iteration on downstream boundary conditions            component induced by the downstream boundary condition <math>\mathbf{c}_{down}^{(i)}</math>            if the current component contributes to the solution            update Moskowitz function            compute density</p>
--	---

### 2.5. Affine internal boundary conditions

While we mainly focus on Cauchy problems in the present article, the proposed algorithm can also integrate any number of internal boundary conditions, which can be used to model fixed or moving bottlenecks (Leclercq et al., 2004). We illustrate this by computing a traffic scenario containing one traffic light and one moving bottleneck in Section 3.5.

An affine internal boundary condition of the Moskowitz function is defined as:

$$N_{\text{intern}}^{(l)}(x, t) = M^{(l)} + q_{\text{intern}}^{(l)}(t - t_{\text{min}}^{(l)}) : t_{\text{min}}^{(l)} \leq t \leq t_{\text{max}}^{(l)} \text{ and } x = x^{(l)} + V_{\text{intern}}^{(l)}(t - t_{\text{min}}^{(l)}) \tag{13}$$

In the above formula, the internal boundary condition imposes a maximal flow of  $q_{\text{intern}}^{(l)}$  on the domain defined by  $t_{\text{min}}^{(l)} \leq t \leq t_{\text{max}}^{(l)}$  and  $x = x_{\text{min}}^{(l)} + V_{\text{intern}}^{(l)}(t - t_{\text{min}}^{(l)})$ . It can represent in practice a fixed ( $V_{\text{intern}}^{(l)} = 0$ ) or a moving ( $V_{\text{intern}}^{(l)} > 0$ ) bottleneck, restricting the relative capacity of the road to  $q_{\text{intern}}^{(l)}$  on its path.

### 3. Analytical solutions to the Moskowitz HJ PDE and LWR PDE

#### 3.1. Solutions to the Hamilton–Jacobi equation

In order to compute the analytical solution of Eq. (6) with conditions of the type (10)–(12) or possibly (13), we define (based on Claudel and Bayen (2010a), Aubin et al. (2008), and Daganzo (2006)) the following convex transform associated with the fundamental diagram:

$$\forall u \in [w, v_f], \quad R(u) = \sup_{k \in [0, x]} (Q(k) - u \cdot k) \tag{14}$$

Note that  $R$  is a convex function, since it is the supremum of affine functions. The function  $-R$  is the Legendre–Fenchel transform of the function  $Q$  (fundamental diagram).

We aggregate the initial and boundary conditions of the Cauchy problem (6) into a single value condition function  $\mathbf{c}(x, t)$ :

$$\mathbf{c}(x, t) = \begin{cases} N_{\text{ini}}(x) & t = 0 \\ N_{\text{up}}(t) & x = x_0 \\ N_{\text{down}}(t) & x = x_n \end{cases}$$

With this definition of initial, upstream and downstream boundary conditions, the domain of definition of  $\mathbf{c}$  is  $\text{Dom}(\mathbf{c}) = (\{x_0, x_n\} \times [0, t_m]) \cup (\{x_0, x_n\} \times \{0\})$ .

Variational theory as known since (Angel and Bellman, 1972) is a possible method for solving the HJ PDE (5). A possible use of variational theory is presented in Daganzo (2005b) in the context of this problem. The mathematical foundations of variational theory can be found in Aubin et al. (2008) for the specific case of the HJ PDEs. From Aubin et al. (2008), the solution associated with the value condition function  $\mathbf{c}$ , denoted by  $N_{\mathbf{c}}$ , is the infimum of an infinite number of functions of the value condition:

$$N_{\mathbf{c}}(x, t) = \inf \{ \mathbf{c}(t - T, x - Tu) + TR(u) \} \tag{15}$$

s.t.  $(u, T) \in [w, v_f] \times \mathbb{R}_+$  and  $(t - T, x - Tu) \in \text{Dom}(\mathbf{c})$

Eq. (15) is well known in the Hamilton–Jacobi literature and often referred to as Lax–Hopf formula (Aubin et al., 2008; Evans, 1998). Note that it can be solved using dynamic programming methods (Daganzo, 2005b), which are proven to be exact for homogeneous problems under two conditions: a concave, piecewise-affine fundamental diagram, and the computation of the Moskowitz function on a uniform grid  $(j\delta x, i\delta t)_{ij}$  which has to be invariant by any translation of vector  $(\delta x, \delta x/w_i)$  for each wave speed  $w_i$  (Daganzo, 2005b). These conditions can be restrictive based on the data used for practical applications since they dictate a sampling frequency. Also, they can be computationally intensive.

In this article, we present a method which guarantees an analytical solution, and has the same computational complexity for arbitrary grids and concave fundamental diagrams. In addition, we show that this method enables us to compute the solution to the associated Cauchy problem (6) exactly.

For this, we first decompose the piecewise affine value condition function  $\mathbf{c}$  in affine, locally-defined value conditions indexed by ini, up and down based on the condition (initial, upstream, downstream), and  $i$  or  $j$  depending on the sampling interval:

$$\forall x \in [x_i, x_{i+1}[, \quad \mathbf{c}_{\text{ini}}^{(i)}(x, 0) = N_{\text{ini}}(x) \tag{16}$$

$$\forall t \in [t_j, t_{j+1}[, \quad \mathbf{c}_{\text{up}}^{(j)}(x_0, t) = N_{\text{up}}(t) \tag{17}$$

$$\forall t \in [t_j, t_{j+1}[, \quad \mathbf{c}_{\text{down}}^{(j)}(x_n, t) = N_{\text{down}}(t) \tag{18}$$

where  $N_{\text{ini}}$ ,  $N_{\text{up}}$  and  $N_{\text{down}}$  are defined in (10)–(12). One can note that the functions  $\mathbf{c}_{\text{ini}}^{(i)}$ ,  $\mathbf{c}_{\text{up}}^{(j)}$  and  $\mathbf{c}_{\text{down}}^{(j)}$  are restrictions of the piecewise-affine function  $\mathbf{c}$  on intervals on which it is affine.

If the problem contains fixed or moving bottlenecks, we can also define a finite number of affine internal boundary conditions indexed by intern and  $l$ :

$$\forall t \in [t_{\text{min}}^{(l)}, t_{\text{max}}^{(l)}] \text{ and for } x = x_{\text{min}}^{(l)} + V_{\text{intern}}^{(l)}(t - t_{\text{min}}^{(l)}), \quad \mathbf{c}_{\text{intern}}^{(l)}(x, t) = N_{\text{intern}}^{(l)}(x, t) \tag{19}$$

We also define the induced solution components  $N_{\mathbf{c}_{\text{ini}}^{(i)}}$ ,  $N_{\mathbf{c}_{\text{up}}^{(j)}}$ ,  $N_{\mathbf{c}_{\text{down}}^{(j)}}$  and  $N_{\mathbf{c}_{\text{intern}}^{(l)}}$  respectively associated with (16)–(19) when computed by (15) as follows. The induced solution components must be understood as the partial solution to a subset of the

initial problem, since we assume that we have no information and do not impose any value on the other initial and boundary condition domains. The contribution of the present article is the construction of the full solution of the problem from these partial solutions.

In general, the Moskowitz function  $N_c$ , solving the HJ PDE (5) for the value condition  $\mathbf{c}$ , cannot be computed analytically using (15) for arbitrary piecewise affine initial, boundary and internal boundary conditions. However, the induced Moskowitz components defined above can be computed analytically using (15). These partial computations involve the minimization of a convex function (Claudel and Bayen, 2010b) which we present later in the article. Using the minimum property (Newell, 1993), also known as inf-morphism property (Aubin et al., 2008), the Moskowitz function  $N_c$ , solving the HJ PDE (5) for the value condition  $\mathbf{c}$  is the minimum of the induced Moskowitz components. This fundamental property (Aubin et al., 2008) is the basis of the algorithm presented in this article.

### 3.2. Discussion and comparison with the variational method

The algorithm that we propose in this article is related to the variational theory introduced in Daganzo (2005a), though the computational method is different. In Daganzo (2005a, 2006), Eq. (15) is solved numerically using dynamic programming methods. For this, a grid is chosen, which define possible paths. Each path to a given point of the grid is associated with a value (or cost). The solution at a given point  $(t, x)$  is the minimum cost over all possible paths originating from the initial or boundary conditions and reaching this point.

In contrast, in this article, the solution at a given point  $(t, x)$  is the minimum of the solution components induced by the affine initial condition blocks, the affine upstream and downstream boundary condition blocks, and the affine internal boundary condition blocks (if any). The solution components of these affine condition blocks can all be computed explicitly (the explicit formulas are presented in the following section). Thus, the solution at a given point  $(t, x)$  is actually the minimum of a finite number of quantities which can all be explicitly computed (using a closed form formula).

The following table summarizes the differences between our proposed algorithm and the dynamic programming method:

	Lax–Hopf algorithm	Dynamic programming (variational method)
Computational principle	Minimization of closed form partial solutions (grid-free)	Minimization of a cost function over a computational grid
Computational cost	Proportional to the total number of initial and boundary condition blocks (depends upon the complexity of the problem)	Function of the grid (depends upon the complexity of the grid)
Exactness	Exact solution for the Moskowitz and density functions for piecewise affine initial boundary and internal boundary conditions and arbitrary concave fundamental diagrams	Exact for piecewise affine initial and boundary conditions and piecewise affine fundamental diagrams. Not exact for arbitrary concave fundamental diagrams, and no exact computation of the density function
Integration of internal boundary conditions (moving bottlenecks, traffic lights...)	Straightforward (explicit formulas also exist for internal boundary conditions)	Possible, but requires the definition of shortcuts in the domain of the internal boundary condition
Integration of space/time varying fundamental diagrams	Possible, but requires a completely new derivation of explicit formulas for space/time varying diagrams	Straightforward (requires the introduction of a space or time dependent cost)

The number of operations required to compute the solution to any given point  $(t, x)$  using the proposed algorithm is proportional to the number of initial and boundary conditions blocks. Thus, our proposed algorithm is more efficient than the dynamic programming approach, which requires at least one path originating from each of the initial or boundary condition blocks. Nevertheless, the variational method is more flexible than the proposed algorithm, and can easily compute solutions associated with inhomogeneous problems. While our algorithm could also compute such solutions, each problem would require the derivation of new explicit formulas.

### 3.3. Solution components associated with affine conditions

In this section, we use the notation  $Q'$  for the derivative of the fundamental diagram  $Q$  and  $R'$  for the derivative of the convex transform  $R$ . If  $Q$  is not differentiable at a particular point  $k$ ,  $Q'(k)$  can be replaced in the formulas below by the left (or the right) derivative at  $k$  (a similar property applies to  $R'$ ). Note that we have also assumed earlier that  $Q$  has a right derivative  $v_f$  in  $0$  and a left derivative  $w$  in  $\kappa$ .

We now present the computation of the partial components of the solution.

3.3.1. Initial conditions

We want to compute the solution component induced by an affine, locally defined initial condition indexed by  $i$ :

$$\forall x \in [x_i, x_{i+1}], \quad N_{ini}^{(i)}(x) = -k_i x + b_i \tag{20}$$

with  $b_i = k_i x_i - \sum_{l=0}^{i-1} (x_{l+1} - x_l) k_l^{(l)}$  allowing for the continuity of the initial conditions in  $(0, x_i)$ . Using the results of **Claudel and Bayen (2010b)**, the analytical solution to the problem associated with (6) with this sole initial condition can be written as

$$N_{c_{ini}^{(i)}}(x, t) = \begin{cases} tQ(k_i) - k_i x + b_i & : x_i + tQ'(k_i) \leq x \leq x_{i+1} + tQ'(k_i) \\ tR\left(\frac{x-x_i}{t}\right) - k_i x_i + b_i & : x_i + tw \leq x \leq x_i + tQ'(k_i) \\ tR\left(\frac{x-x_{i+1}}{t}\right) - k_i x_{i+1} + b_i & : x_{i+1} + tQ'(k_i) \leq x \leq x_{i+1} + tv_f \end{cases} \tag{21}$$

$$k_{c_{ini}^{(i)}}(x, t) = -\frac{\partial N_{c_{ini}^{(i)}}(x, t)}{\partial x} = \begin{cases} k_i & : x_i + tQ'(k_i) \leq x \leq x_{i+1} + tQ'(k_i) \\ -R'\left(\frac{x-x_i}{t}\right) & : x_i + tw \leq x \leq x_i + tQ'(k_i) \\ -R'\left(\frac{x-x_{i+1}}{t}\right) & : x_{i+1} + tQ'(k_i) \leq x \leq x_{i+1} + tv_f \end{cases} \tag{22}$$

3.3.2. Upstream boundary conditions

We now compute the solution component induced by an affine, locally defined upstream boundary condition indexed by  $j$ .

$$\forall t \in [t_j, t_{j+1}], \quad N_{up}^{(j)}(t) = q_j t + d_j \tag{23}$$

with  $d_j = -q_j t_j + \sum_{l=0}^{j-1} (t_{l+1} - t_l) q_l^{(l)}$ . Following **Claudel and Bayen (2010b)**, we define the free flow density function  $K_{up}$ , which is the inverse of the restriction of the fundamental diagram  $Q$  to the domain  $[0, k_c]$ :

$$K_{up}(q) = \min\{k \in [0, \kappa] | Q(k) = q\}$$

Using the results of **Claudel and Bayen (2010b)**, one can prove that:

$$N_{c_{up}^{(j)}}(x, t) = \begin{cases} d_j + q_j t_{j+1} + (t - t_{j+1})R\left(\frac{x-x_0}{t-t_{j+1}}\right) & : 0 \leq x - x_0 \leq Q'(K_{up}(q_j))(t - t_{j+1}) \\ d_j + q_j t - K_{up}(q_j)(x - x_0) & : Q'(K_{up}(q_j))(t - t_{j+1}) \leq x - x_0 \leq Q'(K_{up}(q_j))(t - t_j) \\ d_j + q_j t_j + (t - t_j)R\left(\frac{x-x_0}{t-t_j}\right) & : Q'(K_{up}(q_j))(t - t_j) \leq x - x_0 \leq v_f(t - t_j) \end{cases} \tag{24}$$

$$k_{c_{up}^{(j)}}(x, t) = -\frac{\partial N_{c_{up}^{(j)}}(x, t)}{\partial x} = \begin{cases} -R'\left(\frac{x-x_0}{t-t_{j+1}}\right) & : 0 \leq x - x_0 \leq (t - t_{j+1})Q'(K_{up}(q_j)) \\ K_{up}(q_j) & : Q'(K_{up}(q_j))(t - t_{j+1}) \leq x - x_0 \leq Q'(K_{up}(q_j))(t - t_j) \\ -R'\left(\frac{x-x_0}{t-t_j}\right) & : Q'(K_{up}(q_j))(t - t_j) \leq x - x_0 \leq v_f(t - t_j) \end{cases} \tag{25}$$

3.3.3. Downstream boundary conditions

Again, the same process can be repeated for the downstream boundary:

$$\forall t \in [t_j, t_{j+1}], \quad N_{down}^{(j)}(t) = p_j t + b_j \tag{26}$$

with  $b_j = -p_j t_j + N_{ini}^{(n-1)}(x_n) + \sum_{l=0}^{j-1} (t_{l+1} - t_l) q_l^{(l)}$ . In a symmetric way from the upstream case, we define the congestion density function  $K_{down}$ , which is the inverse of the restriction of the fundamental diagram  $Q$  to the domain  $[k_c, \kappa]$ :

$$K_{down}(q) = \max\{k \in [0, \kappa] | Q(k) = q\}$$

Using the results of **Claudel and Bayen (2010b)**, we can similarly prove that:

$$N_{c_{down}^{(j)}}(x, t) = \begin{cases} b_j + p_j t + K_{down}(p_j)(x_n - x) & : Q'(K_{down}(p_j))(t - t_j) \leq x - x_n \leq Q'(K_{down}(p_j))(t - t_{j+1}) \\ b_j + p_j t_j + (t - t_j)R\left(\frac{x_n-x}{t-t_j}\right) & : w(t - t_j) \leq x - x_n \leq Q'(K_{down}(p_j))(t - t_j) \\ b_j + p_j t_{j+1} + (t - t_{j+1})R\left(\frac{x_n-x}{t_{j+1}-t}\right) & : Q'(K_{down}(p_j))(t - t_{j+1}) \leq x - x_n \leq 0 \end{cases} \tag{27}$$

$$k_{c_{down}^{(j)}}(x, t) = -\frac{\partial N_{c_{down}^{(j)}}(x, t)}{\partial x} = \begin{cases} K_{down}(p_j) & : Q'(K_{down}(p_j))(t - t_j) \leq x - x_n \leq Q'(K_{down}(p_j))(t - t_{j+1}) \\ -R'\left(\frac{x-x_n}{t-t_j}\right) & : w \leq \frac{x_n-x}{t-t_j} \leq Q'(K_{down}(p_j)) \\ -R'\left(\frac{x-x_n}{t-t_{j+1}}\right) & : Q'(K_{down}(p_j))(t - t_{j+1}) \leq x - x_n \leq 0 \end{cases} \tag{28}$$



3.3.4. Internal boundary conditions

In order to write the internal boundary condition component explicitly, we first have to define  $k_1$  and  $k_2$  such that  $k_1 \leq k_2$  and:

$$Q(k_1) - k_1 V_{\text{intern}}^{(l)} = q_{\text{intern}}^{(l)} \tag{29}$$

$$Q(k_2) - k_2 V_{\text{intern}}^{(l)} = q_{\text{intern}}^{(l)} \tag{30}$$

Using the results of Claudel and Bayen (2010b), we can similarly prove that:

$$N_{\mathbf{c}_{\text{intern}}^{(l)}}(x, t) = \left\{ \begin{array}{l} Q(k_1)(t - t_{\text{min}}^{(l)}) + (x^{(l)} - x)k_1 + M^{(l)} : x^{(l)} + V_{\text{intern}}^{(l)}(t - t_{\text{min}}^{(l)}) \leq x \text{ and} \\ t - t_{\text{max}}^{(l)} \leq \frac{x^{(l)} + V_{\text{intern}}^{(l)}(t - t_{\text{min}}^{(l)}) - x}{-Q'(k_1) + V_{\text{intern}}^{(l)}} \leq t - t_{\text{min}}^{(l)} \\ Q(k_2)(t - t_{\text{min}}^{(l)}) + (x^{(l)} - x)k_2 + M^{(l)} : x^{(l)} + V_{\text{intern}}^{(l)}(t - t_{\text{min}}^{(l)}) \geq x \text{ and} \\ t - t_{\text{max}}^{(l)} \leq \frac{x^{(l)} + V_{\text{intern}}^{(l)}(t - t_{\text{min}}^{(l)}) - x}{-Q'(k_2) + V_{\text{intern}}^{(l)}} \leq t - t_{\text{min}}^{(l)} \\ M^{(l)} + (t - t_{\text{min}}^{(l)})R\left(\frac{x^{(l)} - x}{t_{\text{min}}^{(l)} - t}\right) : x^{(l)} + V_{\text{intern}}^{(l)}(t - t_{\text{min}}^{(l)}) \leq x \text{ and} \\ t - t_{\text{min}}^{(l)} \leq \frac{x^{(l)} + V_{\text{intern}}^{(l)}(t - t_{\text{min}}^{(l)}) - x}{-Q'(k_1) + V_{\text{intern}}^{(l)}} \\ \text{OR} \\ x^{(l)} + V_{\text{intern}}^{(l)}(t - t_{\text{min}}^{(l)}) \geq x \text{ and} \\ t - t_{\text{min}}^{(l)} \leq \frac{x^{(l)} + V_{\text{intern}}^{(l)}(t - t_{\text{min}}^{(l)}) - x}{-Q'(k_2) + V_{\text{intern}}^{(l)}} \\ q_{\text{intern}}^{(l)}(t_{\text{max}}^{(l)} - t_{\text{min}}^{(l)}) + M^{(l)} + : x^{(l)} + V_{\text{intern}}^{(l)}(t - t_{\text{min}}^{(l)}) \leq x \text{ and} \\ (t - t_{\text{max}}^{(l)})R\left(\frac{x^{(l)} + V_{\text{intern}}^{(l)}(t_{\text{max}}^{(l)} - t_{\text{min}}^{(l)}) - x}{t_{\text{max}}^{(l)} - t}\right) \frac{x^{(l)} + V_{\text{intern}}^{(l)}(t - t_{\text{min}}^{(l)}) - x}{-Q'(k_1) + V_{\text{intern}}^{(l)}} \leq t - t_{\text{max}}^{(l)} \\ \text{OR} \\ x^{(l)} + V_{\text{intern}}^{(l)}(t - t_{\text{min}}^{(l)}) \geq x \text{ and} \\ \frac{x^{(l)} + V_{\text{intern}}^{(l)}(t - t_{\text{min}}^{(l)}) - x}{-Q'(k_2) + V_{\text{intern}}^{(l)}} \leq t - t_{\text{max}}^{(l)} \end{array} \right. \tag{31}$$

$$k_{\mathbf{c}_{\text{intern}}^{(l)}}(x, t) = -\frac{\partial N_{\mathbf{c}_{\text{up}}^{(i)}}}{\partial x}(x, t) = \left\{ \begin{array}{l} k_1 : x^{(l)} + V_{\text{intern}}^{(l)}(t - t_{\text{min}}^{(l)}) < x \text{ and} \\ t - t_{\text{max}}^{(l)} \leq \frac{x^{(l)} + V_{\text{intern}}^{(l)}(t - t_{\text{min}}^{(l)}) - x}{-Q'(k_1) + V_{\text{intern}}^{(l)}} \leq t - t_{\text{min}}^{(l)} \\ k_2 : x^{(l)} + V_{\text{intern}}^{(l)}(t - t_{\text{min}}^{(l)}) > x \text{ and} \\ t - t_{\text{max}}^{(l)} \leq \frac{x^{(l)} + V_{\text{intern}}^{(l)}(t - t_{\text{min}}^{(l)}) - x}{-Q'(k_2) + V_{\text{intern}}^{(l)}} \leq t - t_{\text{min}}^{(l)} \\ R'\left(\frac{x - x^{(l)}}{t - t_{\text{min}}^{(l)}}\right) : x^{(l)} + V_{\text{intern}}^{(l)}(t - t_{\text{min}}^{(l)}) < x \text{ and} \\ t - t_{\text{min}}^{(l)} < \frac{x^{(l)} + V_{\text{intern}}^{(l)}(t - t_{\text{min}}^{(l)}) - x}{-Q'(k_1) + V_{\text{intern}}^{(l)}} \\ \text{OR} \\ x^{(l)} + V_{\text{intern}}^{(l)}(t - t_{\text{min}}^{(l)}) > x \text{ and} \\ t - t_{\text{min}}^{(l)} < \frac{x^{(l)} + V_{\text{intern}}^{(l)}(t - t_{\text{min}}^{(l)}) - x}{-Q'(k_2) + V_{\text{intern}}^{(l)}} \\ R'\left(\frac{x^{(l)} + V_{\text{intern}}^{(l)}(t_{\text{max}}^{(l)} - t_{\text{min}}^{(l)}) - x}{t_{\text{max}}^{(l)} - t}\right) : x^{(l)} + V_{\text{intern}}^{(l)}(t - t_{\text{min}}^{(l)}) < x \text{ and} \\ t - t_{\text{max}}^{(l)} > \frac{x^{(l)} + V_{\text{intern}}^{(l)}(t - t_{\text{min}}^{(l)}) - x}{-Q'(k_1) + V_{\text{intern}}^{(l)}} \\ \text{OR} \\ x^{(l)} + V_{\text{intern}}^{(l)}(t - t_{\text{min}}^{(l)}) > x \text{ and} \\ t - t_{\text{max}}^{(l)} > \frac{x^{(l)} + V_{\text{intern}}^{(l)}(t - t_{\text{min}}^{(l)}) - x}{-Q'(k_2) + V_{\text{intern}}^{(l)}} \end{array} \right. \tag{32}$$

### 3.4. Componentwise computation of the Moskowitz/LWR function

It is shown in Aubin (1991) and Aubin et al. (2008) that one way to express the solution of the full problem (6), taking into account all contributions of initial and boundary conditions can be obtained by a union property of capture basins, called inf-morphism property. The inf-morphism property applied to the functions  $(\mathbf{c}_{ini}^{(i)})_i$ ,  $(\mathbf{c}_{up}^{(j)})_j$  and  $(\mathbf{c}_{down}^{(j)})_j$  is expressed by the following equality:

$$N(x, t) = \min_{ij} \left\{ N_{\mathbf{c}_{ini}^{(i)}}(x, t), N_{\mathbf{c}_{up}^{(j)}}(x, t), N_{\mathbf{c}_{down}^{(j)}}(x, t) \right\} \tag{33}$$

This last result is fundamental: it shows that in order to solve the HJ PDE, we only have to apply the formulas above (21), (24), and (27) for each affine piece of initial and boundary condition, which will give the associated solution component, and then compute the minimum of all results.

Integrating internal boundary conditions into the problem follows similarly, by more the corresponding solution components into Eq. (33).

### 3.5. Implementation

We summarize the proposed method for the computation of the Moskowitz function in the algorithm labeled Algorithm 1. The algorithm supposes the knowledge of the value conditions  $k_{ini}^{(i)}$ ,  $q_{up}^{(j)}$ ,  $q_{down}^{(j)}$  and their boundaries  $x_i$ ,  $t_j$ . Note that in order to preserve the exactness of the algorithm (i.e. the analytical expression of the solution), we also need the explicit expressions of the following functions:

- Fundamental diagram  $Q(k)$ .
- Derivative of the fundamental diagram  $Q'(k)$ .
- Convex transform  $R(v)$ .
- Derivative of the convex transform  $R'(v)$ .
- Exact solutions  $k_1$  and  $k_2$  to  $Q(k) - kV_{intern}^{(l)} = q_{intern}^{(l)}$ , where  $V_{intern}^{(l)}$  and  $q_{intern}^{(l)}$  are given (this is required only if the problem contains internal boundary conditions).

As mentioned earlier, the functions  $Q$  and  $R$  may be non-differentiable at a finite number of points. In practice, the function  $Q'$  can be defined as any function comprised between the right and left derivatives (which are defined everywhere) of  $Q$ , and similarly for  $R'$  (any function comprised between the left and right derivatives of  $R$ ). These choices do not influence the solution in practice: they will only modify the value of the computed solution to the LWR problem at points for which the true solution is discontinuous anyway (and thus has no definite value).

As can be seen from the implementation of Algorithm 1 posted online (refer to the link in the abstract), all what is required is to define the five above functions analytically, and to prescribe the data at  $[x_0, x_1], \dots, [x_{n-1}, x_n]$  and  $[t_0, t_1], \dots, [t_{m-1}, t_m]$  and the point  $(x, t)$  where one wants to compute the solution. The implementation of Algorithm 1 contains numerical examples involving the Greenshields flux function and the triangular flux function.

In the following sections, we illustrate the capabilities of the algorithm on three different examples. The first example deals with a free downstream boundary condition problem, using a Greenshields fundamental diagram. The second example shows how the algorithm can integrate internal boundary conditions (moving and fixed bottlenecks), using a triangular fundamental diagram. The third example deals with the same internal boundary condition problem, solved using a more general piecewise quadratic and non-differentiable fundamental diagram.

#### 3.5.1. Example 1: Initial-boundary conditions and Greenshields fundamental diagram

The performance of the algorithm is illustrated for an arbitrary set of numerical values, summarized in the tables below. The NaN for the downstream boundary encodes the fact that the downstream boundary is free in this example.

$i$	$t_{i-1}$	$t_i$	$q_{up}^{(i)}$	$q_{down}^{(i)}$
1	0	10	0	NaN
2	10	20	.4	NaN
3	20	40	.1	NaN
4	40	50	0	NaN

$i$	$x_{i-1}$	$x_i$	$k_{ini}^{(i)}$
1	0	100	.08
2	100	200	0
3	200	450	.04
4	450	1000	.003

The solution of the Cauchy problem associated with these value conditions and a Greenshields fundamental diagram (Greenshields, 1935; Garavello and Piccoli, 2006):  $Q(k) = k v_f(1 - k/\kappa)$ , with  $v_f = 30$  m/s the free-flow speed and  $\kappa = .1$  veh/m the maximum density, is represented in Fig. 5.

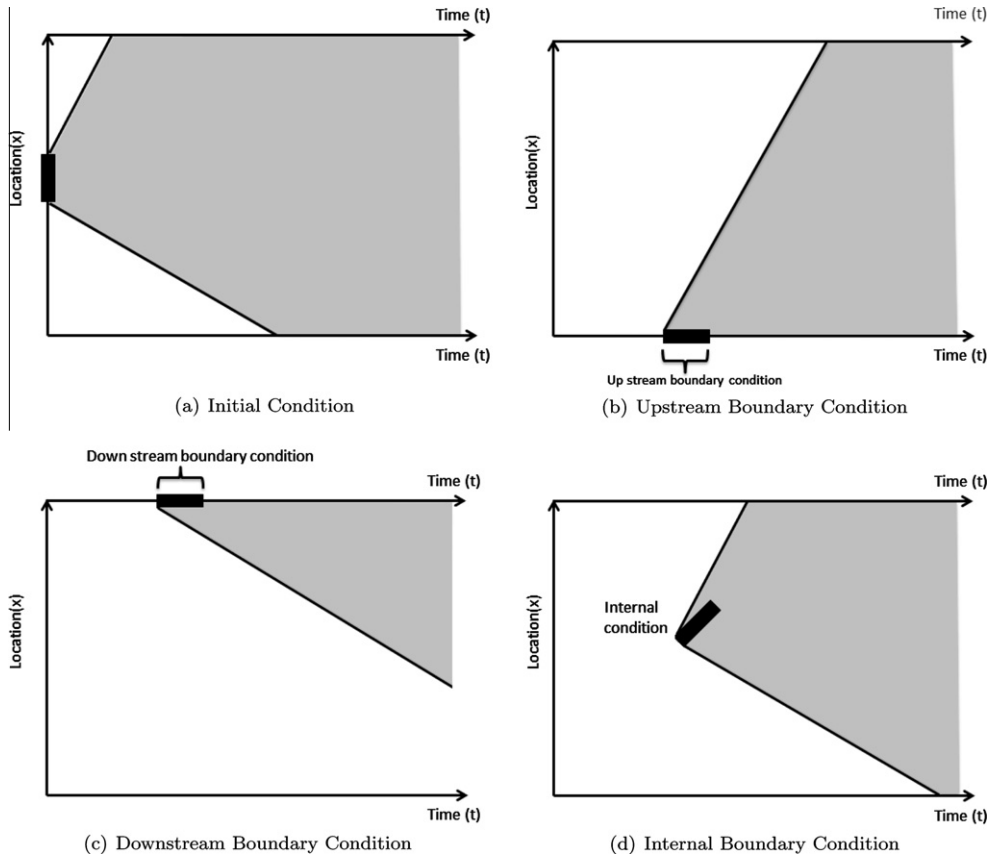


Fig. 4. Illustration of the domains of influence of initial, boundary and internal boundary conditions.

3.5.2. Example 2: Moving and fixed bottlenecks and triangular fundamental diagram

Another example is illustrated below to show the capability of our algorithm to compute the solution of an arbitrary problem along with internal boundary conditions. In this problem, we use the following initial and boundary conditions:

$i$	$t_{i-1}$	$t_i$	$q_{up}^{(i)}$	$q_{down}^{(i)}$
1	0	30	.4	.3
2	30	35	.1	0
3	35	50	.2	.1

$i$	$x_{i-1}$	$x_i$	$k_{ini}^{(i)}$
1	0	200	.08
2	200	500	.01
3	500	1000	.03

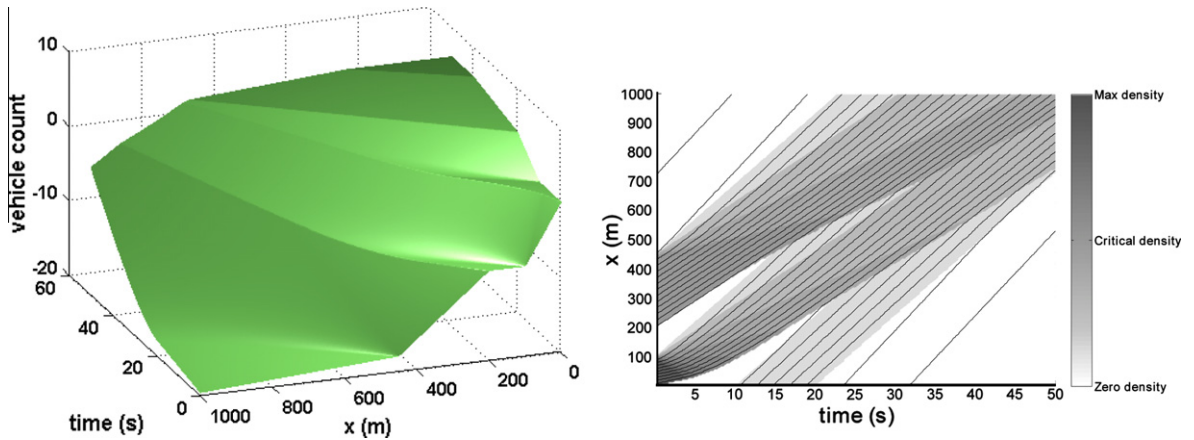
For this problem, we consider a triangular fundamental diagram as defined in Section 5.1, with  $v_f = 30$  m/s as the free-flow speed,  $\kappa = .1$  veh/m as the maximum capacity, and  $w = -5$  m/s as the congested wave speed. The solution of the Cauchy problem associated with these value conditions is represented in the upper part of Fig. 6.

We then add an arbitrary moving bottleneck to the problem. The moving bottleneck is initially located at  $x^{(1)} = 600$  m and is active between times  $t_{min}^{(1)} = 10$  s and  $t_{max}^{(1)} = 15$  s, moving at a speed of 6 m/s, and allowing a passing rate of 0.002 veh/s. We show the exact solution after adding this internal boundary condition in the lower part of Fig. 6. Another example of internal boundary condition is a fixed red traffic light located at  $x_{min}^{(2)} = 800$  m, active between times  $t_{min}^{(2)} = 15$  s and  $t_{max}^{(2)} = 20$  s is shown in the upper part of Fig. 7. As a final illustration, the solution for the combination of initial, boundary and both internal boundary conditions is also presented in the lower part of Fig. 7.

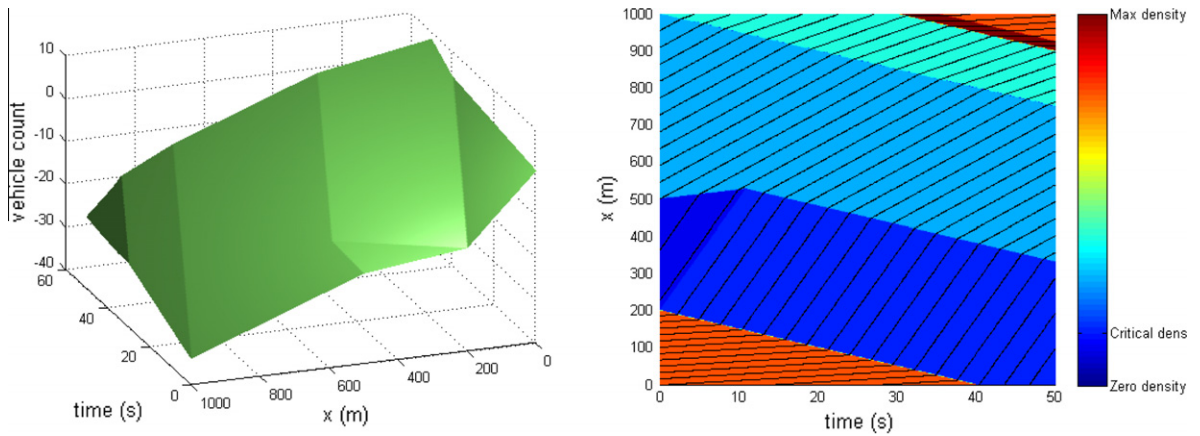
3.5.3. Example 3: Moving and fixed bottlenecks and piecewise quadratic fundamental diagram

A third example is added to show the capability of our algorithm to compute the solution of an arbitrary problem along with internal boundary conditions for a piecewise quadratic and non-differentiable fundamental diagram defined as follows:

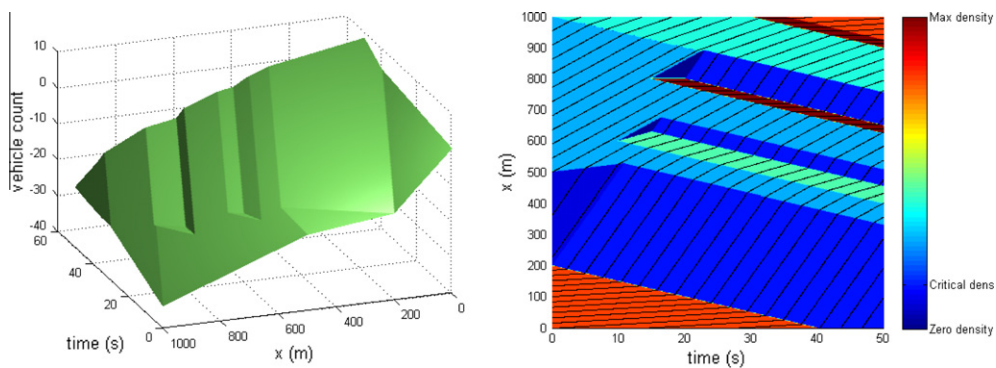
$$Q(k) = \begin{cases} -600k^2 + 30k & \text{if } 0 < k < k_c \\ -5k + 0.5 & \text{if } k_c < k < k_{max} \end{cases} \tag{34}$$



**Fig. 5.** Exact solution to an arbitrary set of initial and value conditions (Example 1), using a Greenshields fundamental diagram. *Left:* Three-dimensional representation of the count function. *Right:* Two-dimensional plot of the density and associated trajectories.



**Fig. 6.** Exact solution to an arbitrary set of initial and boundary conditions (Example 2), using a triangular fundamental diagram. *Left:* Three-dimensional representation of the count function. *Right:* Two-dimensional plot of the density and associated trajectories.



**Fig. 7.** Exact solution to an arbitrary set of initial, boundary and internal boundary conditions, using a Triangular fundamental diagram. In this example, we use the initial and boundary conditions of Fig. 6, and add two internal boundary conditions. The first is located between 600 m and 630 m, covering the period between 10 s and 15 s and allowing a maximal passing rate of 0.002 veh/s. The second one simulates a red traffic light located at 800 m and active between 15 s and 20 s. *Left:* Three-dimensional representation of the count function. *Right:* Two-dimensional plot of the density and associated trajectories.

The fundamental diagram (34) is illustrated in Fig. 8. For this particular example, we use the exact same initial, boundary and internal boundary conditions as in Example 2. The solution associated with the piecewise quadratic and non-differentiable fundamental diagram (34) is presented in Fig. 9.

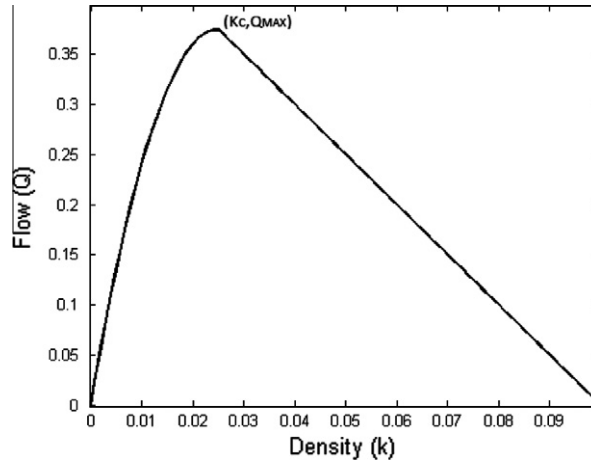


Fig. 8. Piecewise quadratic and non-differentiable fundamental diagram used for the computation of Example 3. This function is defined by  $Q(k) = -600k^2 + 30k$ , for  $0 \leq k \leq k_c$  and by  $Q(k) = -5k + 0.5$ , for  $k_c \leq k \leq k_{max}$ .

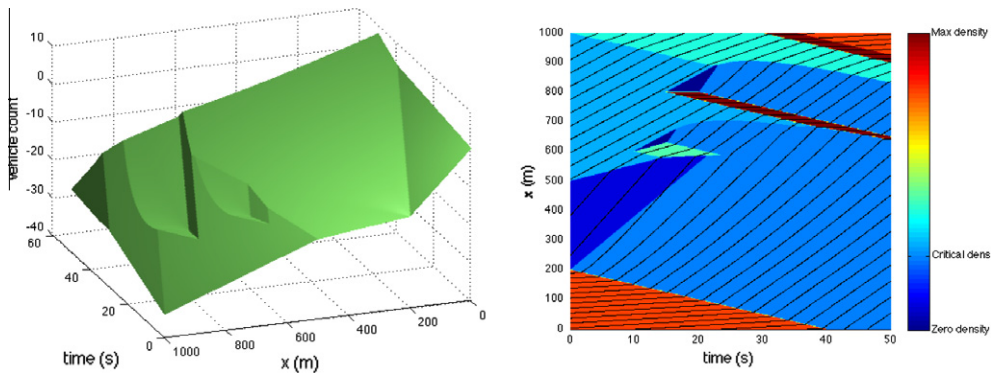


Fig. 9. Exact solution to the arbitrary set of initial, boundary and internal boundary conditions of Example 3, using the piecewise quadratic and non-differentiable fundamental diagram (34) shown in Fig. 8. For this figure, we reuse the same initial, boundary and internal boundary conditions as in Fig. 7. Left: Three-dimensional representation of the count function. Right: Two-dimensional plot of the density and associated trajectories.

### 4. Properties of the solution

#### 4.1. Interpretation and analysis of the solution

##### 4.1.1. Structure

Each solution component  $N_{c^{(i)}}$  induced by a value condition  $c^{(i)}$  is defined in the convex domain:

$$\text{Dom}(N_{c^{(i)}}) = [x_0, x_n] \times [0, t_m] \cap \{(x + v\delta t, t + \delta t) | \delta t \geq 0, v \in [w, v_f], (x, t) \in \text{Dom}(c^{(i)})\} \tag{35}$$

which is the union of all areas that are attainable by a characteristic starting from any point  $(x, t)$  of the value condition domain, and at an information propagation speed  $v \in [w, v_f]$ , sometimes also referred to as the reachable set. In Eq. (35),  $c^{(i)}$  can encode either an initial, upstream, downstream or internal component. The reachable sets associated with the possible value conditions are represented in Fig. 4.

A solution component induced by an affine value condition generally consists of fans and domains of characteristics. An initial or boundary condition component contains two fans and one domain of characteristics. An internal boundary condition component contains three fans and two domains of characteristics.

For instance, a solution component induced by an initial condition associated with the density  $k_i$  has the following structure:

- The *forward fan*: this characteristic fan starts at the most downstream point of the initial condition. Shaped as a cone, it propagates at speeds  $v \in [Q'(k_i), v_f]$ . It is a transition area where, on a given trajectory, the vehicle speeds go from the vehicle speed imposed by the initial condition to the free flow speed, while the density decreases.

- The characteristic domain: this domain propagates at a speed  $Q(k_i)$ . In this area, vehicle speed and density are constant and match those imposed by the initial condition.
- The *backward fan*: this wave starts at the most upstream point of the initial condition. Shaped as a cone, it propagates at speeds  $v \in [w, Q(k_i)]$ . It is a transition area where, on any given trajectory, the vehicle speed goes from zero to the vehicle speed imposed by the initial condition, while the density decreases.

The other solution components have a similar structure. The general structure of the solution components induced by affine initial, boundary and internal boundary conditions is represented in Fig. 10.

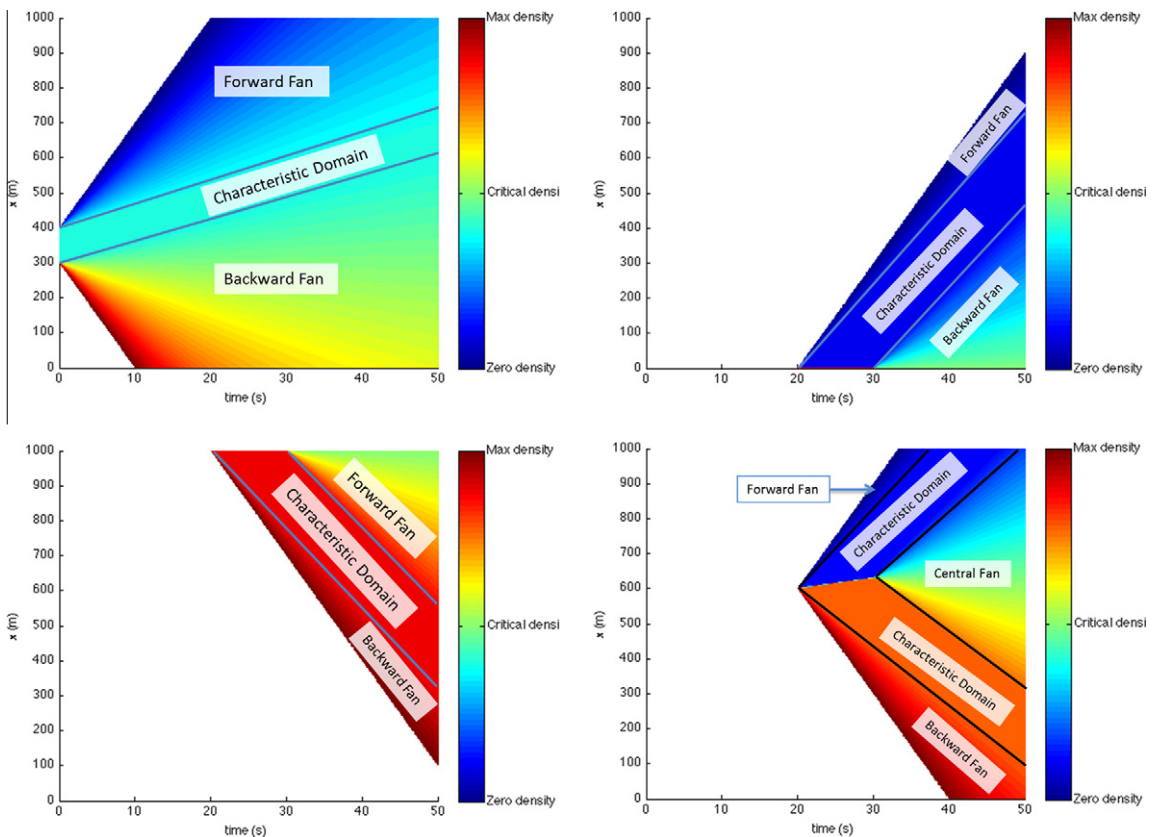
4.1.2. Flows and speeds associated with the solution components

Spatial derivatives are bounded, as shown by the analytical calculation of the densities in (22), (25), (28). From these equations, one can also obtain the flow and speed by using the classical formulas:  $q(x, t) = Q(k(x, t))$  and

$$v(k) = \begin{cases} \frac{Q(k)}{k} & \text{if } 0 < k \leq \kappa \\ Q'(0) & \text{if } k = 0 \end{cases}$$

Note that the results of these inversions are only as good as the approximation used for the fundamental diagram. In particular, the fundamental diagram sometime fails to capture the variability of traffic in congestion (Blandin et al., 2011; Varaiya, 2005).

The convexity of the solution components has been proved in Claudel and Bayen (2010b). Each solution component is Lipschitz-continuous as long as the initial and boundary conditions are such that the imposed densities are positive and below the maximum density, and the imposed flows are positive and below the maximum flow. One can also note that as long as these conditions are met, the solution only exhibits positive flows. These conditions are equivalent to the *well-posedness* conditions introduced in Daganzo (2006).



**Fig. 10.** Structure of the different induced value condition components. *Top left:* Solution component induced by an initial condition. *Top right:* Solution component induced by an upstream boundary condition. *Bottom left:* Solution component induced by a downstream boundary condition. *Bottom right:* Solution component induced by an internal boundary condition.

### 4.2. Lipschitz-continuity of the solution

The Cauchy problem associated to the LWR PDE (6) is called a *well-posed problem* if the initial densities and upstream and downstream flows lie within their physically-imposed limits:

$$0 \leq k_{\text{ini}}^{(i)} \leq \kappa \tag{36}$$

$$0 \leq q_{\text{up}}^{(j)} \leq q_{\text{max}} \tag{37}$$

$$0 \leq q_{\text{down}}^{(j)} \leq q_{\text{max}} \tag{38}$$

The Lipschitz-continuity of the solution to such problems is proven in [Appendix A](#) for initial-boundary condition problems. Interestingly, this proof relies solely on the form of the solution and not on the LWR PDE (though obviously the solution encodes the PDE in its structure). The interpretation is that despite the fact that the solution components are irregularly shaped, they are never strictly below the neighboring components on the edges of their domain of definition. This means that the solution has to be continuous, and the associated flow and density are respectively in  $[0, q_{\text{max}}]$  and  $[0, \kappa]$ .

As will be seen later, considering only well-posed problems allows for simplifications in the computation of the solution by reducing the number of solution components to consider during the computation.

### 4.3. Comparison with other numerical schemes

Our proposed method comes with several benefits: as a grid-free method, it does not require any intermediate computations (small steps) in order to give a forecast of the future traffic state. In general, for finite differences methods such as the CTM or the Godunov scheme, the constraint on the time-space grid spacing  $\delta t, \delta x$  is defined by the CFL condition:  $\frac{\delta x}{\delta t} \geq \sup_k |Q'(k)|$ . Variational theory has an inverse constraint: computational accuracy is maximized for large time steps.

Gridless methods such as wave-tracking methods or the proposed method have no such constraints, and their complexity is problem-dependent. A simple problem with few initial-boundary conditions will be computed faster than a more complex problem with a large number of initial-boundary conditions. However, both methods have distinct computational costs. Our algorithm (as presented in [Algorithm 1](#)) requires  $n_i + n_u + n_b$  operations to compute the solution in one point (where  $n_i$  is the number of initial condition blocks,  $n_u$  is the number of upstream condition blocks, and  $n_d$  is the number of downstream condition blocks). In contrast, the wave-tracking method will compute the evolution of *at least*  $n_i + n_u + n_b$  waves and their intersections. In addition, the number of waves will also depend upon the complexity of the fundamental diagram. For instance, a fundamental diagram with a large number of “pieces” will require more waves to be generated (and thus a higher computational time), while the algorithm we propose in this article has a number of operations independent of the fundamental diagram. The complexity of the wave-tracking algorithm will also depend upon the number of shockwaves generated by the data, which cannot be known in advance.

While the proposed algorithm has a lower complexity than wave-tracking algorithms for computing the solution at one point, this does not imply that our algorithm is faster for computing the solution on the full computational domain. Indeed, the wave tracking algorithm computes the solution on the full domain  $[0, t] \times [x_0, x_n]$  in order to compute the solution at one point  $(t, x)$ .

The main advantages and disadvantages of the main computational methods used for solving the LWR equation are summarized below:

	Lax–Hopf algorithm	Variational method	Godunov scheme	Wave-front tracking
Computational principle	Minimization of closed form partial solutions	Minimization of a cost function over a computational grid	First order finite differences scheme	Event-based scheme
Main advantages	Exact for general concave fundamental diagrams. Very fast for computing the solution at one particular point (or on a small domain). Recomputing the solution after adding a fixed or moving bottleneck is very fast	Exact for some classes of fundamental diagrams. Ability to use space or time dependent fundamental diagrams. Ability to use fixed or moving bottlenecks	Easiness of implementation, natural extension to networks. Ability to use space or time dependent fundamental diagrams	Exact for some classes of fundamental diagrams. Ability to use non-concave or non-continuous fundamental diagrams. Performs well for computing the solution on a full space-time domain. Extends naturally to networks

(continued on next page)

	Lax–Hopf algorithm	Variational method	Godunov scheme	Wave-front tracking
Main disadvantages	Inability to use space or time dependent fundamental diagrams. Results are not exact for networks (unless event-based algorithms are used)	Slower than the Lax–Hopf algorithm (requires in the best case the same number of operations as the Lax–Hopf algorithm). No exact derivation of the density function	Not exact. Limited by the CFL condition. Adding moving bottlenecks is difficult	Difficulty to predict the computational time in advance. Can be very slow if the fundamental diagram contains a large number of pieces

#### 4.4. A word on Lagrangian coordinates

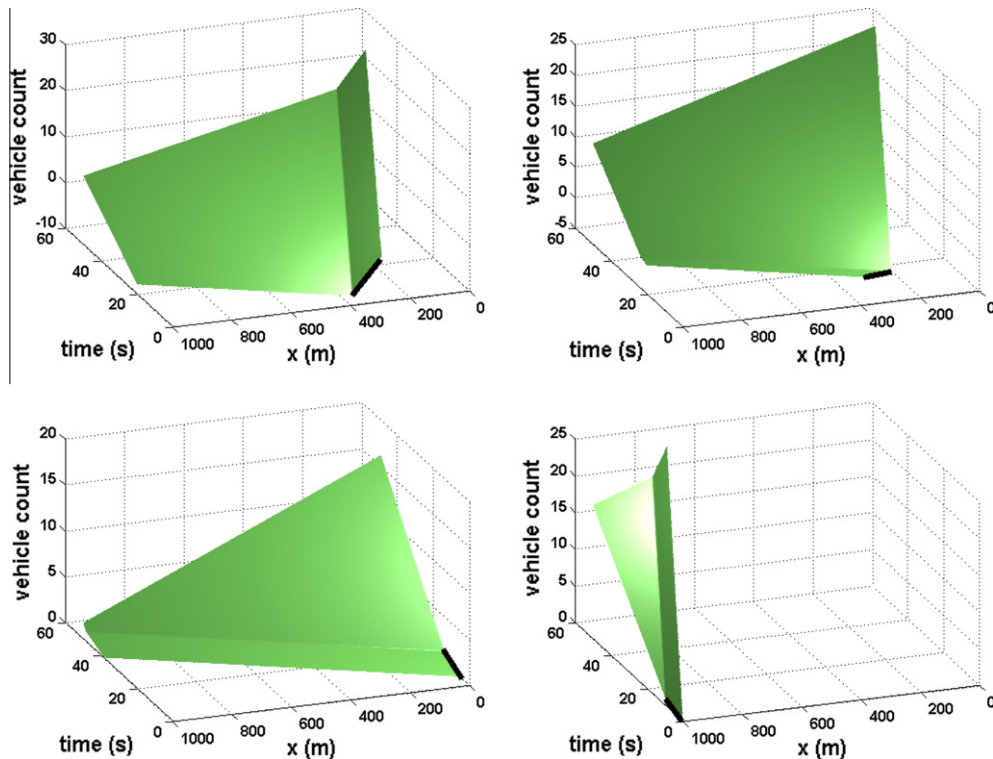
The proposed algorithm can be easily extended to problems in Lagrangian coordinates (Daganzo, 2006). Traffic flow problems in Lagrangian coordinates involve an Hamilton–Jacobi equation with a concave Hamiltonian, for which the solution can also be expressed semi-analytically (for piecewise affine initial and boundary conditions). Extending the proposed algorithm to Lagrangian coordinates amounts to performing a variable change, available for instance in Leclercq et al. (2007).

### 5. Faster algorithm for triangular fundamental diagrams

#### 5.1. Modeling

The triangular fundamental diagram  $Q$  is defined by:

$$Q(k) = \begin{cases} v_f k & : x \in [0, k_c] \\ w(k - \kappa) & : x \in [k_c, \kappa] \end{cases}$$



**Fig. 11.** Three-dimensional representations of the solution components induced by local affine value conditions, for a triangular fundamental diagram. *Top left:* Solution component induced by a congested initial condition. *Top right:* Solution component induced by a free-flow initial condition. *Bottom left:* Solution component induced by an upstream boundary condition. *Bottom right:* Solution component induced by a downstream boundary condition.



where

$$k_c = \frac{-wK}{v_f - w}$$

is the density corresponding to the maximum capacity. This fundamental diagram is graphically represented in Fig. 1. The calculation of its convex transform  $R$  using (14) yields:

$$\forall u \in [w, v_f], R(u) = k_c(v_f - u)$$

Owing to the simplicity of the triangular fundamental diagram and its convex transform, the solution components associated with affine internal and boundary conditions can be calculated explicitly and lead to an even simpler set of results: for an initial condition  $\mathbf{N}_{ini}^{(i)}$  as defined in (20), plugging the explicit convex transform into (21) yields two cases:

If  $0 \leq k_i \leq k_c$ , the initial condition imposes a free-flow state.

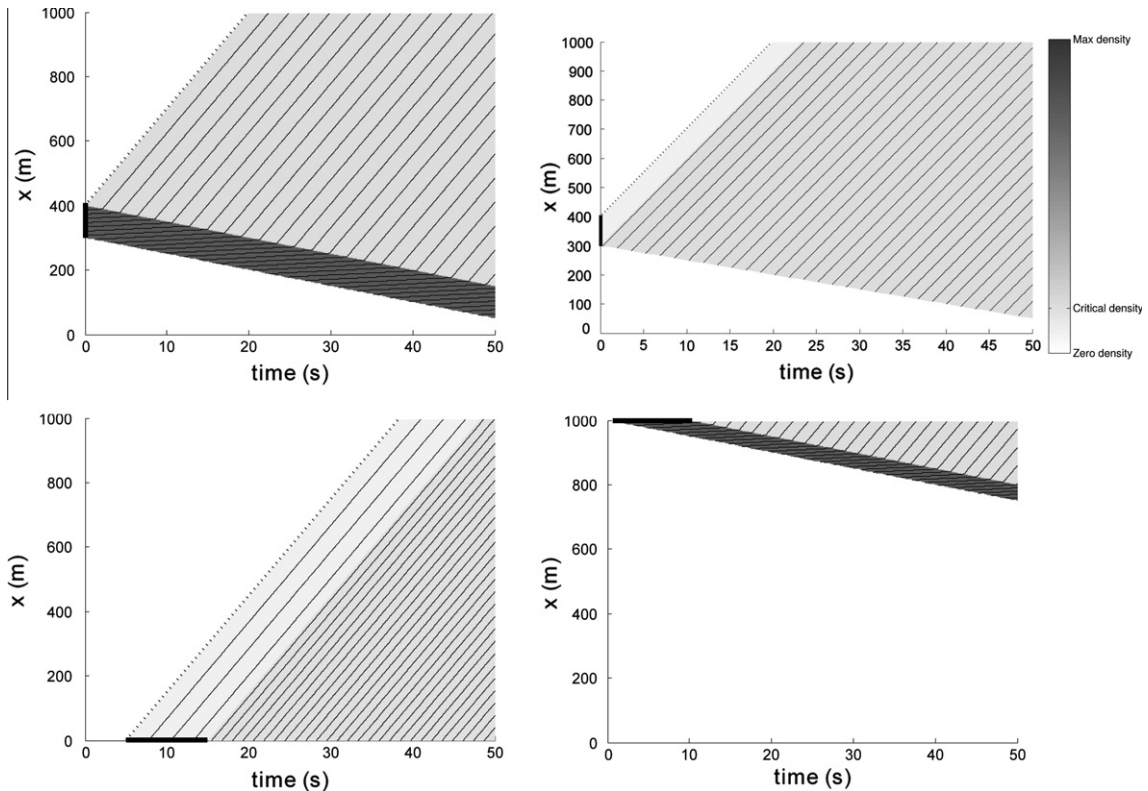
$$N_{c_{ini}^{(i)}}(x, t) = \begin{cases} (i) & k_i(tv_f - x) + b_i & : x_i + tv_f \leq x \leq x_{i+1} + tv_f \\ (ii) & k_c(tv_f - x) + b_i + x_i(k_c - k_i) & : x_i + tw \leq x \leq x_i + tv_f \end{cases} \quad (39)$$

else, if  $k_c < k_i \leq \kappa$ , the initial condition imposes a congested state:

$$N_{c_{ini}^{(i)}}(x, t) = \begin{cases} (i) & k_i(tw - x) - \kappa tw + b_i & : x_i + tw \leq x \leq x_{i+1} + tw \\ (ii) & k_c(tw - x) - \kappa tw + x_{i+1}(k_c - k_i) + b_i & : x_{i+1} + tw \leq x \leq x_{i+1} + tv_f \end{cases} \quad (40)$$

For an upstream boundary condition  $\mathbf{N}_{up}^{(j)}$  as defined in (23), the solution component is expressed by:

$$N_{c_{up}^{(j)}}(x, t) = \begin{cases} (i) & d_j + q_j \left( t - \frac{x-x_0}{v_f} \right) & : x_0 + v_f(t - t_{j+1}) \leq x \leq x_0 + v_f(t - t_j) \\ (ii) & d_j + q_j t_{j+1} + k_c((t - t_{j+1})v_f - (x - x_0)) & : x_0 \leq x \leq x_0 + v_f(t - t_{j+1}) \end{cases} \quad (41)$$



**Fig. 12.** Two-dimensional representations of the solution components induced by local affine value conditions, for a triangular fundamental diagram. Black curves represent isovalues of the Moskowitz function, and therefore vehicle trajectories. *Top left:* Solution component induced by a congested initial condition. *Top right:* Solution component induced by a free-flow initial condition. *Bottom left:* Solution component induced by an upstream boundary condition. *Bottom right:* Solution component induced by a downstream boundary condition.

For a downstream boundary condition  $\mathbf{N}_{\text{down}}^{(j)}$  as defined in (26), the solution component is expressed by:

$$N_{\text{down}}^{(j)}(x, t) = \begin{cases} (i) & b_j + p_j t - (\frac{p_j}{w} + \kappa)(x_n - x) & : x_n + w(t - t_j) \leq x \leq x_n + w(t - t_{j+1}) \\ (ii) & b_j + p_j t_{j+1} + k_c((t - t_{j+1})v_f + x_n - x) & : x_n + w(t - t_{j+1}) \leq x \leq x_n \end{cases} \quad (42)$$

The solution components associated with affine initial and boundary conditions are illustrated in Figs. 11 and 12.

One can note that all solution components consist of two planar portions. The first one, numbered (i) in the solution component equations, is the characteristic domain as defined in Section 4.1.1, and is shaped as a trapezoid. The second one, noted (ii) in the same equations, is cone-shaped, and can represent either an expansion or a rarefaction wave. The characteristic domain propagates directly the density information from the value condition, at a characteristic speed  $v_f$  in free flow or  $w$  in congestion. On the cones, the density is the critical density and the flux is the maximum flux. An interesting point is that this does not depend on whether the cone is part of an expansion or a rarefaction wave. From a mathematical perspective, this comes from the linearity of  $R$ , and its constant derivative.

The algorithm we previously described would still be exact using these simplified formulas for the solution components. Nevertheless, the piecewise-linear shape of the solution components allows us to do several simplifications which yield noticeable improvements for the algorithmic complexity. Our approach to a fast algorithm has some similarities to what has been developed for the variational theory of traffic flow (Daganzo, 2005a): in particular, one can note that for a triangular fundamental diagram, the Lax–Hopf formula can be simplified to become the minimum of two elements, each of them representing a different information propagation speed. This remark yields considerable improvements for the computation of the Moskowitz function.

### 5.2. Fast algorithm for well-posed problems

One can note two important properties of the solution components calculated previously

- The characteristic domains always propagate between speeds  $v_f$  and  $w$ .
- The maximum capacity cones are all parts of parallel plans, since their gradient is  $(v_f k_c, -k_c)$ , and extend at any speed  $u \in [w, v_f]$ .

These properties allow us to restrict the number of solution components we have to compute the Moskowitz function in one point  $(x, t)$ . Indeed, if the problem is well posed, the maximum capacity cones bring limited information since they all are parallel. Thus, we only have to compute the characteristic domains, which restricts the number of value conditions to inspect to two. This simplification yields Algorithm 2, which requires only  $n_i + 2$  operations (where  $n_i$  is the number of initial condition blocks) to compute the solution at one point, while the general algorithm (valid for all fundamental diagrams) presented in Algorithm 1 requires  $n_i + n_u + n_d$  operations (where  $n_u$  and  $n_d$  are the number of upstream and downstream boundary condition blocks respectively).

**Algorithm 2.** Pseudo-code implementation for the Lax–Hopf based computation of the Moskowitz function and the associated density at a single point  $(x, t)$  for a triangular fundamental diagram.

---

<p><b>Input:</b> <math>t \in [0, t_n], x \in [x_0, x_m]</math></p> <p><math>N \leftarrow +\infty</math></p> <p><math>i_{\text{up}} \leftarrow \max \{i   t_i \leq T - \frac{x - x_0}{v_f}\}</math></p> <p><math>i_{\text{down}} \leftarrow \max \{i   t_i \geq T - \frac{x_n - x}{w}\}</math></p> <p><math>j_{\text{min}} \leftarrow \min \{j - 1   x_j \geq X - v_f T\}</math></p> <p><math>j_{\text{max}} \leftarrow \max \{j + 1   x_j \leq X - w T\}</math></p> <p><b>if</b> <math>i_{\text{up}} \neq -\infty</math> <b>then</b></p> <p style="padding-left: 20px;">compute <math>N_{\mathbf{e}_{\text{up}}^{(i_{\text{up}})}(x, t)}</math> using (41)</p> <p style="padding-left: 20px;"><b>if</b> <math>N &lt; N_{\mathbf{e}_{\text{up}}^{(i_{\text{up}})}(x, t)}</math> <b>then</b></p> <p style="padding-left: 40px;"><math>N \leftarrow N_{\mathbf{e}_{\text{up}}^{(i_{\text{up}})}(x, t)}</math></p> <p style="padding-left: 40px;"><math>k \leftarrow k_{\mathbf{e}_{\text{up}}^{(i_{\text{up}})}(x, t)}</math>, computed using (25)</p> <p style="padding-left: 20px;"><b>end if</b></p> <p><b>end if</b></p> <p><b>if</b> <math>i_{\text{down}} \neq -\infty</math> <b>then</b></p> <p style="padding-left: 20px;">compute <math>N_{\mathbf{e}_{\text{down}}^{(i_{\text{down}})}(x, t)}</math> using (42)</p> <p style="padding-left: 20px;"><b>if</b> <math>N &lt; N_{\mathbf{e}_{\text{down}}^{(i_{\text{down}})}(x, t)}</math> <b>then</b></p>	<p>time index for influencing upstream boundary condition</p> <p>time index for influencing downstream boundary condition</p> <p>space index for influencing initial condition</p> <p>space index for influencing initial condition</p> <p>if at least one upstream condition influences <math>(x, t)</math></p> <p>component induced by the upstream condition <math>\mathbf{e}_{\text{up}}^{(i_{\text{up}})}</math></p> <p>if the current component contributes to the solution</p> <p>update Moskowitz function</p> <p>compute density</p> <p>if at least one downstream condition influences <math>(x, t)</math></p> <p>component induced by the downstream condition <math>\mathbf{e}_{\text{down}}^{(i_{\text{down}})}</math></p> <p>if the current component contributes to the solution</p>
--	--

$N \leftarrow N_{\mathbf{c}_{\text{down}}^{(i)}}(x, t)$	update Moskowitz function
$k \leftarrow k_{\mathbf{c}_{\text{down}}^{(i)}}(x, t)$ , computed using (28)	compute density
<b>end if</b>	
<b>end if</b>	
<b>for</b> $j_{\min} \leq j \leq j_{\max}$ <b>do</b>	iteration over initial conditions
compute $N_{\mathbf{c}_{\text{ini}}^{(j)}}(x, t)$ using (39) or (40)	component induced by the initial condition $\mathbf{c}_{\text{ini}}^{(j)}$
<b>if</b> $N < N_{\mathbf{c}_{\text{ini}}^{(j)}}(x, t)$ <b>then</b>	if the current component contributes to the solution
$N \leftarrow N_{\mathbf{c}_{\text{ini}}^{(j)}}(x, t)$	update Moskowitz function
$k \leftarrow k_{\mathbf{c}_{\text{ini}}^{(j)}}(x, t)$ , computed using (22)	compute density
<b>end if</b>	
<b>end for</b> $N, k$	

---

### 6. Conclusions

This article develops an analytical expression of the entropy solution of the Lighthill–Whitham–Richards partial differential equation with an arbitrary flow–density relationship, and with piecewise constant initial, boundary and internal boundary conditions. The analytical nature of the solution enables the construction of algorithms for exact solutions of the partial differential equation (up to machine accuracy) without numerical discretization error. The procedure is straightforward to implement, and has a low computational cost. It enables the computation of the solution at user prescribed points without gridding the space–time domain. We show that this method enable us to derive classical results for the density function and the cumulative number of vehicles function (its integral form). For the specific case of triangular flow–density relationships, we show that the proposed method can be reduced to a simpler and faster algorithm. Future work will deal with the use of the method for traffic flow estimation and/or control, for which some preliminary results have already been obtained (Claudel and Bayen, 2011) in the context of Lagrangian sensing.

### Acknowledgments

The authors are grateful to Professor Lawrence Craig Evans for his guidance on the treatment of non-smoothness arising in solutions to Hamilton–Jacobi equations, and to Professor Jean–Pierre Aubin for his guidance and vision, and for his help to pose the Hamilton–Jacobi problem as a viability problem. Professor Carlos Daganzo is gratefully acknowledged for his guidance on traffic flow engineering. The authors want to express their warmest thanks to Professor Patrick Saint–Pierre for his initial guidance in generating code for numerical solutions of the HJ PDE. The numerical computations presented in this work have been done using technologies initially developed by the company VIMADES.

### Appendix A. Lipschitz-continuity of the solution

**Lemma A.1.** *Let  $f, g: A \rightarrow B$  be two Lipschitz-continuous functions. Then  $h: x \rightarrow \min\{f(x), g(x)\}$  is Lipschitz-continuous on  $A$ .*

**Proof.** We remark that  $2 \cdot \min(a, b) = a + b - |a - b|$ , therefore  $h$  is a linear combination of Lipschitz-continuous functions. Hence, it is Lipschitz-continuous.  $\square$

For notational convenience in the appendix, we use  $\mathbf{c}^{(i)}$  as a generic notation for  $\mathbf{c}_{\text{ini}}^{(i)}, \mathbf{c}_{\text{up}}^{(j)}, \mathbf{c}_{\text{down}}^{(j)}$  through this appendix.

**Definition A.2.** Let  $N_{\mathbf{c}^{(i)}}$  be a solution component. We define the *upper-boundary extension*  $\widehat{N}^{(i)}$  of  $N_{\mathbf{c}^{(i)}}$  the following way: If  $N_{\mathbf{c}^{(i)}}$  is a component induced by an initial boundary condition as defined in (10), i.e.  $\mathbf{c}^{(i)} = \mathbf{c}_{\text{ini}}^{(i)}$ , then

$$\widehat{N}_{\mathbf{c}^{(i)}}(x, t) = \begin{cases} N_{\mathbf{c}^{(i)}}(x, t) & : \text{if } (x, t) \in \text{Dom}(N^{(i)}) \\ N_{\mathbf{c}^{(i)}}(x_{i+1} + t v_f, t) & : \text{if } x > x_{i+1} + t v_f \\ N_{\mathbf{c}^{(i)}}(x_i + t w, t) + (x_i + t w - x) \kappa & : \text{if } x < x_i + t w \end{cases} \quad (43)$$

If  $N_{\mathbf{c}^{(i)}}$  is induced by an upstream boundary condition as defined in (11) i.e.  $\mathbf{c}^{(i)} = \mathbf{c}_{\text{up}}^{(i)}$ , then

$$\widehat{N}_{\mathbf{c}^{(i)}}(x, t) = \begin{cases} N_{\mathbf{c}^{(i)}}(x, t) & : \text{if } (x, t) \in \text{Dom}(N_{\mathbf{c}^{(i)}}) \\ N_{\mathbf{c}^{(i)}}\left(x, t_i + \frac{x - x_0}{v_f}\right) & : \text{if } (x, t) \notin \text{Dom}(N_{\mathbf{c}^{(i)}}) \end{cases} \quad (44)$$

If  $N_{c^{(i)}}$  is induced by a downstream boundary condition as defined in (12) i.e.  $c^{(i)} = c_{\text{down}}^{(i)}$ , then

$$\widehat{N}_{c^{(i)}}(x, t) = \begin{cases} N_{c^{(i)}}(x, t) & : \text{if } (x, t) \in \text{Dom}(N_{c^{(i)}}) \\ N_{c^{(i)}}(x, t_i + \frac{x-x_n}{w}) + (t_i + \frac{x-x_n}{w} - t)q_{\text{max}} & : \text{if } (x, t) \notin \text{Dom}(N_{c^{(i)}}) \end{cases} \tag{45}$$

The interpretation of the upper-boundary extensions is as follows. Downstream from the free flow characteristic line, the solution component is extended with isolines parallel to the  $x$  axis, which corresponds to no cars added downstream. Upstream from the characteristic line with slope  $w$  emanating from  $(x_k, 0)$ , the solution component induced by an initial condition  $c_{\text{ini}}^{(k)}$  is extended so that the corresponding density is the jam density  $\kappa$  at any time. Similarly, the solution component induced by a downstream condition is extended so that the corresponding flow is the maximum flow  $q_{\text{max}}$ . Note that these two parts of the upper-boundary extensions do not satisfy the Hamilton–Jacobi PDE. This mathematical argument does not pose any problem since another component will fill the corresponding area, with a smaller value (the extension on this side of the domain is only required for the mathematical proof of Lipschitz-continuity). Thus, by the Lax–Hopf formula, the extensions do not change the nature of the solution. One can note that every function  $\widehat{N}_{c^{(i)}}$  is designed to be defined and Lipschitz-continuous on the entire domain  $[x_0, x_n] \times [0, t_m]$ , and to coincide with  $N_{c^{(i)}}$  when the latter is defined.

**Lemma A.3.**

$$\forall x \in [x_0, x_n], \forall t \in [0, t_m], \min_{0 \leq i < n} \widehat{N}_{c_{\text{ini}}^{(i)}}(x, t) = \min_{i \text{ s.t. } 0 \leq i < n \text{ and } (x, t) \in \text{Dom}(N_{c_{\text{ini}}^{(i)}})} N_{c_{\text{ini}}^{(i)}}(x, t) \tag{46}$$

**Proof.** Let  $x \in [x_0, x_n], t \in [0, t_m]$  be fixed for the entire proof. We define  $r_{\text{min}} = \max\{k | x_k \leq x - vt\}$  and  $r_{\text{max}} = \max\{k | x_k \leq x - wt\}$ , then  $\forall r \in \{r_{\text{min}}, \dots, r_{\text{max}}\}, (x, t) \in \text{Dom}(N_{c_{\text{ini}}^{(r)}})$ ; thus every point  $(x, t)$  lies in the domain of at least one solution component induced by an initial condition.

The Lax–Hopf formula (15) gives

$$N_{c_{\text{ini}}^{(k)}}(x_k + tw, t) \geq N_{c_{\text{ini}}^{(k-1)}}(x_k + tw, t) \tag{47}$$

as long as  $x_k + tw \in [x_0, x_n]$  and  $t \in [0, t_m]$ . This is due to the fact that the point  $(x_k + tw, t)$  can be reached by more than one characteristic induced by  $c_{\text{ini}}^{(k-1)}$ .

Let  $k > r_{\text{max}}$ , then we have  $x < x_k + tw$  and  $(x, t) \notin \text{Dom}(N_{c_{\text{ini}}^{(k)}})$ . In respective order, the definition of the upper-boundary extension, the property (47), the fact that  $N_{c_{\text{ini}}^{(k-1)}}(x_k + tw, t) = \widehat{N}_{c_{\text{ini}}^{(k-1)}}(x_k + tw, t)$ , and that  $\widehat{N}_{c_{\text{ini}}^{(k-1)}}(\cdot, t)$  is  $\kappa$ -Lipschitz, yield

$$\begin{aligned} \widehat{N}_{c_{\text{ini}}^{(k)}}(x, t) &= N_{c_{\text{ini}}^{(k)}}(x_k + tw, t) + (x_k + tw - x)\kappa \geq N_{c_{\text{ini}}^{(k-1)}}(x_k + tw, t) + (x_k + tw - x)\kappa \geq \widehat{N}_{c_{\text{ini}}^{(k-1)}}(x_k + tw, t) + (x_k + tw - x)\kappa \\ &\geq \widehat{N}_{c_{\text{ini}}^{(k-1)}}(x, t) \end{aligned}$$

Therefore  $\forall k > r_{\text{max}}, \widehat{N}_{c_{\text{ini}}^{(k)}}(x, t) \geq \widehat{N}_{c_{\text{ini}}^{(k-1)}}(x, t)$ . By induction on  $k$ , and using the fact that  $\widehat{N}_{c_{\text{ini}}^{(r_{\text{max}})}}(x, t) = N_{c_{\text{ini}}^{(r_{\text{max}})}}(x, t)$  because  $(x, t) \in \text{Dom}(N_{c_{\text{ini}}^{(r_{\text{max}})}})$ ,

$$\min_{r_{\text{max}} \leq i < n} \widehat{N}_{c_{\text{ini}}^{(i)}}(x, t) = N_{c_{\text{ini}}^{(r_{\text{max}})}}(x, t) \tag{48}$$

The Lax–Hopf formula (15) also gives

$$N_{c_{\text{ini}}^{(k-1)}}(x_k + tv_f, t) \geq N_{c_{\text{ini}}^{(k)}}(x_k + tv_f, t) \tag{49}$$

Let  $k \leq r_{\text{min}}$  such that  $(x, t) \notin \text{Dom}(N_{c_{\text{ini}}^{(k-1)}})$ , then  $x > x_k + tv_f$ . Similarly as for the first case, the definition of the upper-boundary extension, the property (49), the fact that  $\widehat{N}_{c_{\text{ini}}^{(k)}}(\cdot, t)$  is a decreasing function, respectively give the following set of inequalities.

$$\widehat{N}_{c_{\text{ini}}^{(k-1)}}(x, t) = N_{c_{\text{ini}}^{(k-1)}}(x_k + tv_f, t) \geq N_{c_{\text{ini}}^{(k)}}(x_k + tv_f, t) \geq \widehat{N}_{c_{\text{ini}}^{(k)}}(x_k + tv_f, t) \geq \widehat{N}_{c_{\text{ini}}^{(k)}}(x, t)$$

Therefore  $\forall k \leq r_{\text{min}}, \widehat{N}_{c_{\text{ini}}^{(k-1)}}(x, t) \geq \widehat{N}_{c_{\text{ini}}^{(k)}}(x, t)$ . By induction on  $k$ , and using the fact that  $\widehat{N}_{c_{\text{ini}}^{(r_{\text{min}})}}(x, t) = N_{c_{\text{ini}}^{(r_{\text{min}})}}(x, t)$  because  $(x, t) \in \text{Dom}(N_{c_{\text{ini}}^{(r_{\text{min}})}})$ ,

$$\min_{0 \leq i \leq r_{\text{min}}} \widehat{N}_{c_{\text{ini}}^{(i)}}(x, t) = N_{c_{\text{ini}}^{(r_{\text{min}})}}(x, t) \tag{50}$$

Since  $\forall k \in \{r_{\min}, \dots, r_{\max}\}$ ,  $(x, t) \in \text{Dom}\left(N_{\text{ini}}^{(k)}\right)$ , Eqs. (48) and (50) give:

$$\exists r \in \{r_{\min}, \dots, r_{\max}\} : N_{\text{ini}}^{(r)}(x, t) = \min_{0 \leq i \leq n} \widehat{N}_{\text{ini}}^{(i)}(x, t)$$

This last equality concludes the proof of Eq. (46) of the lemma.

This proves that for initial conditions the upper boundary extension of the components does not modify the final solution.  $\square$

**Proposition A.4.**

$$\forall x \in [x_0, x_n], \forall t \in [0, t_m], \min_i \widehat{N}_{\text{c}^{(i)}}(x, t) = \min_{i \text{ s.t. } (x,t) \in \text{Dom}N_{\text{c}^{(i)}}} N_{\text{c}^{(i)}}(x, t)$$

**Proof.** Lemma A.3 states that the above equality holds when one only considers initial boundary conditions. We use the same method to prove the continuity of the entire solution. Let  $x \in [x_0, x_n]$ ,  $\forall t \in [0, t_m]$  be fixed for the entire proof. Let  $j > 0$  such that  $(x, t) \notin \text{Dom}N_{\text{c}^{(j)}}^{\text{up}}$ . The same process as previously, knowing that  $N_{\text{c}^{(j-1)}}^{\text{up}}(x, \cdot)$  is  $q_{\max}$ -Lipschitz, gives

$$\widehat{N}_{\text{c}^{(j)}}^{\text{up}}(x, t) = N_{\text{c}^{(j)}}^{\text{up}}\left(x, t_j + \frac{x - x_0}{v_f}\right) + \left(t_j + \frac{x - x_0}{v_f} - t\right)q_{\max} \geq N_{\text{c}^{(j-1)}}^{\text{up}}\left(x, t_j + \frac{x - x_0}{v_f}\right) + \left(t_j + \frac{x - x_0}{v_f} - t\right)q_{\max} \geq \widehat{N}_{\text{c}^{(j-1)}}^{\text{up}}(x, t)$$

If  $(x, t) \notin \text{Dom}N_{\text{c}^{(0)}}^{\text{up}}$ , one can prove similarly that:

$$\widehat{N}_{\text{c}^{(0)}}^{\text{up}}(x, t) = N_{\text{c}^{(0)}}^{\text{up}}\left(x, \frac{x - x_0}{v_f}\right) + \left(\frac{x - x_0}{v_f} - t\right)q_{\max} \geq N_{\text{c}^{(0)}}^{\text{ini}}\left(x, \frac{x - x_0}{v_f}\right) + \left(\frac{x - x_0}{v_f} - t\right)q_{\max} \geq \widehat{N}_{\text{c}^{(0)}}^{\text{ini}}(x, t)$$

Because any point  $(x, t)$  is on a characteristic curve emanating from  $(x, 0)$ , there exists  $i$  such that  $(x, t) \in \text{Dom}(N_{\text{c}^{(i)}})$ . Furthermore, because of the two previous inequalities, we can extend this statement to the following statement: there exists  $i$  such that  $(x, t) \in \text{Dom}N_{\text{c}^{(i)}}^{\text{up}}$  and  $\widehat{N}_{\text{c}^{(j)}}^{\text{up}}(x, t) \geq N_{\text{c}^{(i)}}(t, x)$ . The same induction as before gives

$$\exists r : N_{\text{c}^{(r)}}(x, t) = \min \left( \left\{ \widehat{N}_{\text{c}^{(i)}}^{\text{up}}(x, t) \mid 0 \leq i < n \right\} \cup \left\{ \widehat{N}_{\text{c}^{(j)}}^{\text{up}}(x, t) \mid 0 \leq j < m \right\} \right)$$

Symmetrically for downstream conditions, we can generate the same results: we know that if  $(x, t) \notin \text{Dom}N_{\text{c}^{(j)}}^{\text{down}}$ ,  $N_{\text{c}^{(j)}}^{\text{down}}(x, t) \geq \widehat{N}_{\text{c}^{(j-1)}}^{\text{down}}(x, t)$  and if  $(x, t) \notin \text{Dom}N_{\text{c}^{(0)}}^{\text{down}}$ ,  $N_{\text{c}^{(0)}}^{\text{down}}(x, t) \geq \widehat{N}_{\text{c}^{(n-1)}}^{\text{down}}(x, t)$ . Therefore,

$\exists i : (x, t) \in \text{Dom}N_{\text{c}^{(i)}}^{\text{down}}$  and  $\widehat{N}_{\text{c}^{(j)}}^{\text{down}}(x, t) \geq N_{\text{c}^{(i)}}(t, x)$ . A quick induction then shows that

$$\min_{i \text{ s.t. } (x,t) \in \text{Dom}N_{\text{c}^{(i)}}^{\text{down}}} \widehat{N}_{\text{c}^{(i)}}^{\text{down}}(x, t) = \min_i N_{\text{c}^{(i)}}(x, t) \quad \square$$

**Theorem A.5.** Let us consider the Cauchy problem as stated in Eq. (6). If the imposed initial densities are in  $[0, \kappa]$  and if the imposed upstream and downstream flows are in  $[0, q_{\max}]$ , then the solution to the Cauchy problem is Lipschitz-continuous.

**Proof.** We apply lemma (A.1) to the result of Proposition A.4.  $\square$

**References**

Angel, E., Bellman, R., 1972. Dynamic Programming and Partial Differential Equations. Academic Press, New York, NY.  
 Ansgore, R., 1990. What does the entropy condition mean in traffic flow theory? Transportation Research Part B 24 (2), 133–143.  
 Aubin, J.-P., 1991. Viability Theory. Systems and Control: Foundations and Applications. Birkhäuser, Boston, MA.  
 Aubin, J.-P., Bayen, A.M., Saint-Pierre, P., 2008. Dirichlet problems for some Hamilton–Jacobi equations with inequality constraints. SIAM Journal on Control and Optimization 47 (5), 2348–2380.  
 Bardos, C., Leroux, A.Y., Nedelec, J.C., 1979. First order quasilinear equations with boundary conditions. Communications in Partial Differential Equations 4 (9), 1017–1034.  
 Blandin, S., Work, D., Goatin, P., Piccoli, B., Bayen, A., 2011. A general phase transition model for vehicular traffic. SIAM Journal on Applied Mathematics 71 (1), 107–127.  
 Chen, W., Wong, S., Shu, C., Zhang, P., 2009. Front tracking algorithm for the Lighthill–Whitham–Richards traffic flow model with a piecewise quadratic, continuous, non-smooth, and non-concave fundamental diagram. International Journal of Numerical Analysis and Modeling 6 (4), 562–585.  
 Claudel, C.G., Bayen, A.M., 2010a. Lax–Hopf based incorporation of internal boundary conditions into Hamilton–Jacobi equation. Part I: Theory. IEEE Transactions on Automatic Control 55 (5), 1142–1157.  
 Claudel, C.G., Bayen, A.M., 2010b. Lax–Hopf based incorporation of internal boundary conditions into Hamilton–Jacobi equation. Part II: Computational methods. IEEE Transactions on Automatic Control 55 (5), 1158–1174.  
 Claudel, C.G., Bayen, A.M., 2011. Convex formulations of data assimilation problems for a class of Hamilton–Jacobi equations. SIAM Journal on Control and Optimization 49 (2), 383–402.  
 Daganzo, C.F., 1994. The cell transmission model: a dynamic representation of highway traffic consistent with the hydrodynamic theory. Transportation Research Part B 28 (4), 269–287.

- Daganzo, C.F., 1995. The cell transmission model. Part II: Network traffic. *Transportation Research Part B* 29 (2), 79–93.
- Daganzo, C.F., 2005a. A variational formulation of kinematic waves: basic theory and complex boundary conditions. *Transportation Research Part B* 39 (2), 187–196.
- Daganzo, C.F., 2005b. A variational formulation of kinematic waves: solution methods. *Transportation Research Part B* 39 (10), 934–950.
- Daganzo, C.F., 2006. On the variational theory of traffic flow: well-posedness, duality and applications. *Networks and Heterogeneous Media* 1 (4), 601–619.
- Edie, L., 1961. Car-following and steady-state theory for noncongested traffic. *Operations Research* 9 (1), 66–76.
- Evans, L.C., 1998. *Partial Differential Equations*. American Mathematical Society, Providence, RI.
- Garavello, M., Piccoli, B., 2006. *Traffic Flow on Networks*. American Institute of Mathematical Sciences, Springfield, MO.
- Geroliminis, N., Daganzo, C.F., 2008. Existence of urban-scale macroscopic fundamental diagrams: some experimental findings. *Transportation Research Part B* 42 (9), 759–770.
- Godunov, S.K., 1959. A difference method for numerical calculation of discontinuous solutions of the equations of hydrodynamics. *Matematicheskii Sbornik* 47 (3), 271–306.
- Greenshields, B.D., 1935. A study of traffic capacity. *Highway Research Board Proceedings* 14, 448–477.
- Henn, V., 2003. A wave-based resolution scheme for the hydrodynamic LWR traffic flow model. In: Hoogendoorn, Luding, B.S.W. (Eds.), *Proceedings of the Workshop on Traffic and Granular Flow 03*, Amsterdam, pp. 105–124.
- Jin, W.-L., 2010. Continuous kinematic wave models of merging traffic flow. *Transportation Research Part B* 44 (8–9), 1084–1103.
- Kerner, B.S., Konhäuser, P., 1994. Structure and parameters of clusters in traffic flow. *Physical Review E* 50 (1), 54–83.
- Leclercq, L., Chanut, S., Lesort, J.B., 2004. Moving bottlenecks in Lighthill–Whitham–Richards model: a unified theory. *Transportation Research Record* 1883, 3–13.
- Leclercq, L., Laval, J., Chevallier, E., 2007. The Lagrangian coordinates and what it means for first order traffic flow models. In: *Proceedings of the International Symposium on Transportation and Traffic Theory*, pp. 735–754.
- Le Floch, P., 1988. Explicit formula for scalar non-linear conservation laws with boundary condition. *Mathematical Methods in the Applied Sciences* 10 (3), 265–287.
- Lighthill, M.J., Whitham, G.B., 1956. On kinematic waves. II. A theory of traffic flow on long crowded roads. *Proceedings of the Royal Society of London* 229 (1178), 317–345.
- Lu, Y., Wong, S., Zhang, M., Shu, C., Chen, W., 2008. Explicit construction of entropy solutions for the Lighthill–Whitham–Richards traffic flow model with a piecewise quadratic flow–density relationship. *Transportation Research Part B* 42 (4), 355–372.
- Makigami, Y., Newell, G.F., Rothery, R., 1971. Three-dimensional representation of traffic flow. *Transportation Science* 5 (3), 303–313.
- Moskowitz, K., 1965. Discussion of ‘freeway level of service as influenced by volume and capacity characteristics’ by D.R. Drew and C.J. Keese. *Highway Research Record* 99, 43–44.
- Newell, G.F., 1993. A simplified theory of kinematic waves in highway traffic. Part (I), (II) and (III). *Transportation Research Part B* 27 (4), 281–313.
- Richards, P.I., 1956. Shock waves on the highway. *Operations Research* 4 (1), 42–51.
- Strub, I.S., Bayen, A.M., 2006. Weak formulation of boundary conditions for scalar conservation laws: an application to highway traffic modelling. *International Journal of Robust and Nonlinear Control* 16 (16), 733–748.
- Varaiya, P., 2005. Reducing highway congestion: an empirical approach. *European Journal of Control* 11 (4–5), 301–309.
- Wong, S., Wong, G., 2002. An analytical shock-fitting algorithm for LWR kinematic wave model embedded with linear speed–density relationship. *Transportation Research Part B* 36 (8), 683–706.
- Yadong, L., Wong, S., Mengping, Z., Chi-wang, S., 2009. The entropy solutions for the Lighthill–Whitham–Richards traffic flow model with a discontinuous flow–density relationship. *Transportation Science* 43 (4), 511–530.