

Fast algorithms for the approximation of a fluid-dynamic model on networks

GABRIELLA BRETTI *ROBERTO NATALINI †

BENEDETTO PICCOLI ‡

Abstract

New computation algorithms for a fluid-dynamic mathematical model of flows on networks are proposed, described and tested. First we improve the classical Godunov scheme (G) for a special flux function, thus obtaining a more efficient method, the Fast Godunov scheme (FG) which reduces the number of evaluations for the numerical flux. Then a new method, namely the Fast Shock Fitting method (FSF), based on good theoretical properties of the solution of the problem is introduced. Numerical results and efficiency tests are presented in order to show the behaviour of FSF in comparison with G, FG and a conservative scheme of second order.

Key Words: scalar conservation laws, traffic flow, fluid-dynamic models, finite difference schemes. *AMS Subject Classification* 65M06. Secondary 90B20, 35L65, 90B10.

1 Introduction.

In this paper we want to introduce new computation algorithms to deal with a fluid-dynamic mathematical model for flows on networks. In particular, we consider traffic and telecommunication networks. Here we deal with the models proposed in [4, 6], which are macroscopic fluid-dynamic models, respectively, for traffic flow on road networks and for flows of information on a telecommunication network. In the 1950s James Lighthill and Gerald Whitham in [9], and independently Richards in [11], proposed to apply fluid-dynamics concepts to traffic. This nonlinear model is based on the single conservation law expressing the conservation of cars:

$$\partial_t \rho + \partial_x f(\rho) = 0, \quad (1.1)$$

*Department of Information Engineering and Applied Mathematics of the University of Salerno, via Ponte don Melillo, 84084 Fisciano (SA), Italy; E-mail: g.bretti@iac.cnr.it.

†Istituto per le Applicazioni del Calcolo “M. Picone”, Viale del Policlinico 137, 00161 - Roma, Italy; E-mail: r.natalini@iac.cnr.it.

‡Istituto per le Applicazioni del Calcolo “M. Picone”, Viale del Policlinico 137, 00161 - Roma, Italy; E-mail: b.piccoli@iac.cnr.it.

where $\rho = \rho(t, x)$ is the density, with $\rho \in [0, \rho_{max}]$, $(t, x) \in \mathbb{R}^2$ and ρ_{max} is the maximum density; $f(\rho)$ is the flux, which can be written as $f(\rho) = \rho v(\rho)$. The function $v(t, x)$ is the velocity. Typically v is assumed to be a smooth decreasing function of ρ and f is concave. Fluid-dynamic models can describe macroscopic phenomena as shocks formation and propagation. Since they can develop discontinuities in a finite time even starting from smooth initial data, the study of the analytical and numerical aspects is fundamental. Recently, in [3, 4, 6], some models based on the conservation formulation (1.1) have been proposed for flows on networks and it was proved existence of solutions to Cauchy problems. Road (transmission lines) networks are a finite set of roads (lines) linked by junctions and, in such models, some rules (see [6]) are introduced to uniquely solve Riemann problems at junctions.

The numerical approximation of the traffic models described in [3, 4] has been studied in [2], where simulation algorithms based on some numerical schemes such as Godunov scheme and 3-velocities Kinetic schemes of first and second order have been proposed, with the introduction of suitable boundary conditions at the junctions. Motivated by real applications we address the following issues demanding for fast algorithms:

- Real urban networks are usually formed by thousands of arcs, representing roads, walking paths etc., and nodes, representing junctions of various types.
- In case of accidents or other reasons for partial interruption of part of the network, it is important to have a real-time simulation tool in order to study the evolution of heavy traffic load and planning a strategy for traffic redirection.
- The future aim is to have an important ability of communication with drivers (via information panels, GPS system etc.) This opens the possibility of optimization via traffic regulation even in real time. Clearly any optimization algorithm needs some computational effort, whose basic step is the solution of an evolution equation for the flow on the network.

First, starting from the numerical approximation methods described in [2], we improve the performance of the Godunov scheme using a simplified flux function. Indeed, by considering only the possible values that the numerical flux correspondent to the analytical flux (2.2) can assume, we introduce the Fast Godunov scheme (FG). The latter can be implemented substituting the usual computations for numerical flux with some “if then” instructions based on a-priori given flux values. This clearly strongly speeds up computations.

Then, supported by theoretical considerations, we can introduce a new numerical method, which is valid only for solutions starting from an empty network, or, alternatively, loaded starting by an empty configuration. Again we focus on the analytical flux (2.2). Under these conditions we can introduce an “exact” procedure to solve this problem, the Fast Shock Fitting (FSF), which is a fast algorithm that constructs the solutions by localizing shocks. More precisely,

we establish the existence of a unique shock, in fact a generalised characteristic, which separates traffic phases inside each road. Such shock is called separating shock and is tracked by an exact fitting procedure. If we start from a non empty network, which was anyway loaded from an empty one, it is possible to use an approximate fitting procedure and then use our algorithm. Once the separating shock is tracked, thanks to the special flux function considered, the rest of the solution is easily computed by a shift procedure.

Notice that FG is applicable also for general initial configurations or in the case of conservation laws with a source term, while FSF cannot be used for these problems.

To complete the analysis, we also consider the 3-velocities Kinetic scheme of second order (K3V) for the same flux function. Then we compare through some numerical experiments the performances of the schemes regarding the accuracy and the time of execution. We observe that FG allows to save more than 50% of CPU time with respect to the classical Godunov. On the other hand, FSF outperforms FG of an increasing percentage, which is around 70% for large networks and small space discretization steps.

The paper is organized as follows. Section 2 is devoted to the description of the model. In Section 3 we describe the numerical schemes to approximate the problem, namely FG, K3V and FSF. In Section 4 we present some numerical experiments and analyse the features of the new approximation methods in terms of accuracy and CPU time.

2 Analytical framework

We consider the equation (1.1) on a traffic or a telecommunication network. From now on we will focus only on the road networks, since the case of telecommunication networks, where roads are replaced by transmission lines and the role of junctions is played by routers, presents the same features. The parametrization of arcs composing a network is made through a set of intervals $I_i = [a_i, b_i] \subset \mathbb{R}$, $i \in 1, \dots, N$, with the endpoints possibly infinite. The datum is given by a finite set of functions ρ_i defined on $[0, +\infty[\times I_i$. Each road can be incoming at most for one junction and outgoing for at most one junction, hence the complete model is given by a pair $(\mathcal{I}, \mathcal{J})$, with $\mathcal{I} = \{I_i : i = 1, \dots, N\}$ the collection of arcs and \mathcal{J} the set of junctions.

Let us consider the conservation equation:

$$\partial_t \rho + \partial_x f(\rho) = 0.$$

If the velocity function is chosen as follows:

$$v(\rho) = \begin{cases} \bar{v} & \text{if } 0 \leq \rho \leq \sigma, \\ \bar{v} \left(\frac{2\sigma}{\rho} - 1 \right) & \text{if } \sigma \leq \rho \leq \rho_{max}, \end{cases} \quad (2.1)$$

then the flux $f = v\rho$ is

$$f(\rho) = \begin{cases} \bar{v}\rho & \text{if } 0 \leq \rho \leq \sigma, \\ \bar{v}(2\sigma - \rho) & \text{if } \sigma \leq \rho \leq \rho_{max}. \end{cases} \quad (2.2)$$

If we fix $\rho_{max} = 1$, $\bar{v} = 1$ and $\sigma = \frac{1}{2}$, then the graph of the velocity is depicted in Fig. 1, while flux function assumes the shape described by Fig. 2. Notice that the velocities of Lax waves not crossing σ is \bar{v} or $-\bar{v}$.

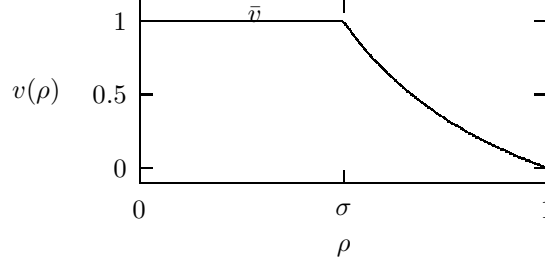


Figure 1: Velocity as a function of the density.

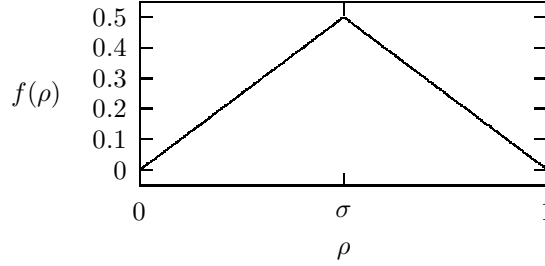


Figure 2: The flux function.

Since solutions are well understood on roads I_i , due to final velocity of propagation, it is enough to define solutions at junctions. Consider a junction J with n incoming arcs and m outgoing arcs. A *weak solution at the junction J* is a collection of functions $\rho_l : [0, +\infty[\times I_l \rightarrow \mathbb{R}$, $l = 1, \dots, n + m$, such that

$$\sum_{l=0}^{n+m} \left(\int_0^{+\infty} \int_{a_l}^{b_l} \left(\rho_l \frac{\partial \varphi_l}{\partial t} + f(\rho_l) \frac{\partial \varphi_l}{\partial x} \right) dx dt \right) = 0, \quad (2.3)$$

for every φ_l , $l = 1, \dots, n + m$, smooth having compact support in $(0, +\infty) \times (a_l, b_l]$ for $l = 1, \dots, n$ (incoming roads) and in $(0, +\infty) \times [a_l, b_l)$ for $l = n + 1, \dots, n + m$ (outgoing roads), that are also smooth across the junction.

Remark 2.1 Let $\rho = (\rho_1, \dots, \rho_{n+m})$ be a weak solution at the junction such that each $x \rightarrow \rho_i(t, x)$ has bounded variation. We can deduce that ρ satisfies the conservation of fluxes at the junction J , for almost every $t > 0$, namely

$$\sum_{i=1}^n f(\rho_i(b_i-, t)) = \sum_{j=n+1}^{n+m} f(\rho_j(a_j+, t)). \quad (2.4)$$

A Riemann problem is a Cauchy problem with Heaviside type initial data. In case of concave or convex fluxes, the Riemann solutions are of two types: continuous waves called rarefactions and travelling discontinuities called shocks.

For a junction a Riemann problem is a Cauchy problem with an initial data constant on each road (the natural discontinuity being the junction). We look for self-similar solutions with one wave on each road.

Definition 2.1 A Riemann solver for the junction J is a map $RS : [0, 1]^n \times [0, 1]^m \rightarrow [0, 1]^n \times [0, 1]^m$ that associates to Riemann data $\rho_0 = (\rho_{10}, \dots, \rho_{n+m0})$ at J a vector $\hat{\rho} = (\hat{\rho}_1, \dots, \hat{\rho}_{n+m})$ so that the solution on an incoming transmission line I_i , $i = 1, \dots, n$ is given by the wave $(\rho_i, \hat{\rho}_i)$ and on an outgoing one I_j , $j = n + 1, \dots, n + m$, is given by the wave $(\hat{\rho}_j, \rho_j)$. We require the consistency condition

$$(CC) \quad RS(RS(\rho_0)) = RS(\rho_0).$$

For instance, two different Riemann solvers at a junction, were introduced in [4, 6], that are represented by the criteria:

RS1 The RS is described by two rules:

- (A) there are some fixed coefficients, the prescribed preferences of drivers, that express the distribution of traffic from incoming to outgoing roads;
- (B) respecting (A), drivers choices are made in order to maximize the flux.

RS2 The flow through the junction is maximized both over incoming and outgoing arcs.

Once the solutions to Riemann problems are given, one can use a wave front tracking algorithm to construct a sequence of approximate solutions.

Definition 2.2 Fix an approximate wave front tracking solution ρ and a road I_i , $i = 1, \dots, N$. A wave θ in I_i is said a big shock if

$$\text{sgn}(\rho_-^\theta - \sigma) \cdot \text{sgn}(\rho_+^\theta - \sigma) < 0,$$

with $\rho_-^\theta < \rho_+^\theta$ for concave flux.

Definition 2.3 Fix an approximate wave front tracking solution ρ and a junction J . We say that an incoming road I_i has a good datum at J at time $t > 0$ if

$$\rho_i(t, b_i-) \in [\sigma, 1]$$

and a bad datum otherwise. We say that an outgoing road I_i has a good datum at J at time $t > 0$ if

$$\rho_i(t, a_i+) \in [0, \sigma]$$

and a bad datum otherwise.

For the proof of the next Lemma, we need to introduce some more notation:

Definition 2.4 Let $\tau : [0, 1] \rightarrow [0, 1]$ be the map such that:

1. $f(\tau(\rho)) = f(\rho)$ for every $\rho \in [0, 1]$;
2. $\tau(\rho) \neq \rho$ for every $\rho \in [0, 1] \setminus \{\sigma\}$.

Obviously τ is well defined and satisfies:

$$0 \leq \rho \leq \sigma \iff \sigma \leq \tau(\rho) \leq 1, \quad \sigma \leq \rho \leq 1 \iff 0 \leq \tau(\rho) \leq \sigma.$$

Lemma 2.2 If road I_i of a junction J has a good datum, then it remains good after interactions with J of waves coming from other roads. Then, no big shock can be produced in this way. If road I_i has a bad datum, then after interactions with J of waves coming from other roads, either the datum of I_i is unchanged or a big shock is produced (and the new datum is good).

PROOF. Let us consider the case in which I_i is an incoming road, depicted in Fig. 3. In order to enter from the junction J , a wave must have a negative speed. Therefore, if $\rho_{i,0}(t, b_i-) \in [\sigma, 1]$ (good datum), then the new state $\hat{\rho}_i \in [\sigma, 1]$, thus the resulting wave can be a rarefaction or a shock. If the initial data $\rho_i(t, b_i-) \in [0, \sigma]$ (bad datum), then the new state $\hat{\rho}_i \in \{\rho_i(t, b_i-)\} \cup [\tau(\rho_i(t, b_i-), 1]$, hence, if a wave is created, it is a shock. In the case of outgoing roads, represented in Fig. 4, the wave must have a positive speed. Thus, if $\rho_i(t, a_i+) \in [0, \sigma]$ (good datum), then the new state $\hat{\rho}_i \in [0, \sigma]$, thus the resulting wave can be a shock or a rarefaction. If $\rho_i(t, a_i+) \in [\sigma, 1]$ (bad datum), then the new state $\hat{\rho}_i \in \{\rho_i(t, a_i+)\} \cup [0, \tau(\rho_i(t, a_i+))]$, therefore, if a wave is created, it is a big shock. \square

Now we focus on solutions generated starting by an empty network, i.e. $\rho(0, x) = 0$, to give some theoretical results on which the FSF scheme is based. We introduce below the definition of separating shock, which is in fact a generalised characteristic, which separates each road in two zones of light and heavy traffic. Roughly speaking, a generalised characteristic is a curve which solves the classical equation for characteristic as long as the solution is smooth, while it follows a shock when some discontinuity appears. For precise definition and theoretical results we refer to [5].

INCOMING ROADS

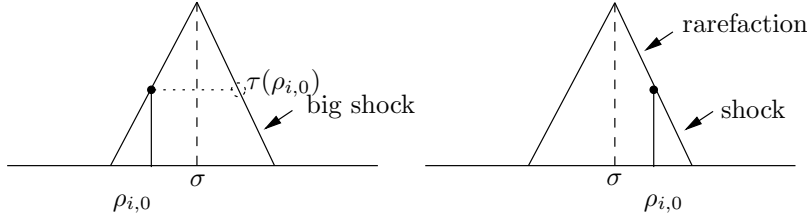


Figure 3: Wave with negative speed.

OUTGOING ROADS

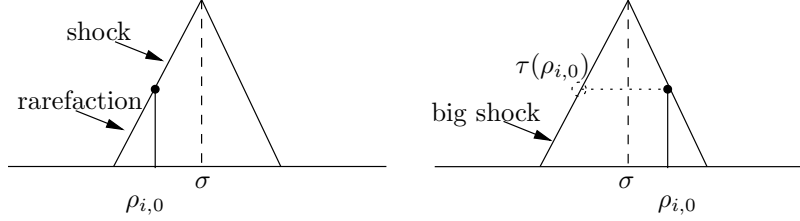


Figure 4: Wave with positive speed.

Definition 2.5 Consider a road network $(\mathcal{I}, \mathcal{J})$ and an approximate wave front tracking solution ρ , such that $\rho(0, x) = 0$. For every road I_i , we define $S_i(t)$ to be the generalised characteristic such that $S_i(0) = b_i$, with the following conventions. If $S_i(t) = a_i$ (respectively b_i), then $S_i \equiv a_i$ (respectively b_i) up to the first time in which a generalised characteristic starts from a_i (respectively b_i). Moreover, we assume the characteristic velocity to be equal to zero at σ . We call S_i the separating shock of I_i .

Definition 2.6 We say that there is a latent separation shock on road I_i at a_i (on the left) if $S_i(t) = a_i$. Analogously, there is a latent separation shock on road I_i at b_i (on the right) if $S_i(t) = b_i$.

Lemma 2.3 For every solution ρ with $\rho(0, x) = 0$ and for every I_i the following statements hold:

a) For every $t \geq 0$ such that $S_i(t) \in]a_i, b_i[$, there exists a (possibly null) shock $(\rho_-^\theta, \rho_+^\theta)$ at $S_i(t)$ with $\text{sgn}(\rho_-^\theta - \sigma) \cdot \text{sgn}(\rho_+^\theta - \sigma) \leq 0$.

b) $\forall x \in [a_i, b_i]$

$$\begin{aligned} x > S_i(t) &\Rightarrow \rho_i(t, x) \in [\sigma, 1] \\ x < S_i(t) &\Rightarrow \rho_i(t, x) \in [0, \sigma]. \end{aligned}$$

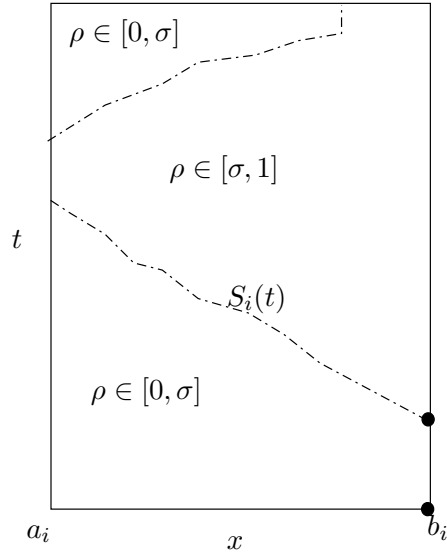


Figure 5: The generalised characteristic S_i .

PROOF. We prove both claims at the same time by recursion. Let \bar{t} be the first time such that $S_i(t) < b_i$ for t in a right neighbourhood of \bar{t} . Then both claims are easily verified up to time \bar{t} . Then, from Lemma 2.2, we have that for t in a neighbourhood of \bar{t} , $S_i(t)$ is a big shock and a) and b) hold. Now, if $S_i(t) \in]a_i, b_i[$ and a), b) hold, then a), b) hold true for every interaction with a wave from the right or on the left. If $S_i(t) = a_i$, then S_i may leave the left endpoint only if a shock $(\tilde{\rho}, \sigma)$, with $\tilde{\rho} < \sigma$, is generated from a_i . Thus a) and b) still hold. If $S_i(t) = b_i$, then S_i may leave the right endpoint only if a shock $(\sigma, \tilde{\rho})$, with $\tilde{\rho} > \sigma$, is generated from b_i . Hence a) and b) still hold. \square

The main idea for FSF is the following. Since we have at most one separation shock per arc, the procedure of the algorithm for determining exactly the solutions consists of the two main steps below:

- exact localization of the separation shock;
- shift by velocities \bar{v} or $-\bar{v}$.

3 Approximation schemes

Here we present the kinetic scheme for the initial-boundary value conservation equation:

$$u_t + F(u)_x = 0, \quad (3.1)$$

$$u(0, x) = u_0(x), \quad x \geq 0, \quad (3.2)$$

$$u(t, 0) = u_b(t), \quad t \geq 0, \quad (3.3)$$

In order to find approximate solutions, we use three different schemes, namely a modified version of the classical Godunov scheme (FG), the 3-velocities Kinetic scheme of second order (K3V) presented in [2] and the Fast Shock Fitting (FSF) scheme, which solves exactly the problem. In particular, FG brings a simplification in the computations of the numerical flux respect to the classical scheme. In fact, by analysing the expression of the flux function (2.2) we are able to restrict the range of possible values which can be assumed by the flux itself (see paragraph 3.1). The second scheme, namely K3V, is a quite recent scheme for conservation laws [10], that has been adapted to traffic flow problem in [2]. FSF represents a completely original method, which is based on a localization technique of big shocks and allows us to compute the exact solution, see paragraph 3.2.

Notice that the interactions at junctions are solved by a Linear Programming algorithm that compute the maximized fluxes for all the schemes.

We define a *numerical grid* in $\mathbb{R}^N \times (0, T)$ using the following notations:

- Δx is the space grid size;
- Δt is the time grid size;
- $(t_n, x_m) = (n\Delta t, m\Delta x)$ for $n \in \mathbb{N}$ and $m \in \mathbb{Z}$ are the grid points.

For a function v defined on the grid we write $v_m^n = v(t_n, x_m)$ for m, n varying on a subset of \mathbb{Z} and \mathbb{N} respectively. We also use the notation u_m^n for $u(t_n, x_m)$ when u is a continuous function on the (t, x) plane.

3.1 Fast Godunov scheme (FG)

A good numerical method to solve the equations along the roads is represented by the Godunov scheme, which solves Riemann problems exactly, [7, 8]. A solution of the problem is constructed taking a piecewise constant approximation of the initial data, v_0^Δ . We set

$$v_m^0 = \frac{1}{\Delta x} \int_{x_m}^{x_{m+1}} u_0(x) dx, \quad m \geq 0 \quad (3.4)$$

and the scheme defines v_m^n recursively starting from v_m^0 .

Remark 3.1 A wave starting from $x_{m-1/2}$ will not reach the lines $x = x_{m-1}$ and $x = x_m$ before time t_{n+1} if

$$\Delta t \sup_{m,n} \left\{ \sup_{u \in I(u_m^n, u_{m+1}^n)} |F'(u)| \right\} \leq \frac{1}{2} \Delta x. \quad (3.5)$$

Then we define the projection of the exact solution on a piecewise constant function

$$v_m^{n+1} = \frac{1}{\Delta x} \int_{x_m}^{x_{m+1}} v^\Delta(t_{n+1}, x) dx. \quad (3.6)$$

Since v is an exact solution of (1.1), we use the Gauss-Green formula in (1.1) to compute this value. Under the CFL condition

$$\Delta t \sup_{m,n} \left\{ \sup_{u \in I(u_m^n, u_{m+1}^n)} |F'(u)| \right\} \leq \Delta x, \quad (3.7)$$

the waves do not influence the solutions in $x = x_m$, for $t \in (t_n, t_{n+1})$. Hence the solutions are locally given by the Riemann problems and in particular the flux in $x = x_m$ for $t \in (t_n, t_{n+1})$ is given by $F(u(t, x_m)) = F(W_R(0; v_{m-1}^n, v_m^n))$, where $W_R(\frac{x}{t}; v_-, v_+)$ is the self-similar solution between v_- and v_+ . As the flux is time invariant and continuous, we can put it out of the integral and, setting $g^G(u, v) = F(W_R(0; u, v))$ under the condition (3.7), the scheme can be written as:

$$v_m^{n+1} = v_m^n - \frac{\Delta t}{\Delta x} (g^G(v_m^n, v_{m+1}^n) - g^G(v_{m-1}^n, v_m^n)). \quad (3.8)$$

The expression of numerical flux for Godunov method is in general

$$g^G(u, w) = \begin{cases} \min_{z \in [u, w]} f(z) & \text{if } u \leq w, \\ \max_{z \in [w, u]} f(z) & \text{if } w \leq u, \end{cases}$$

and for the flux function under consideration, we have:

$$g^G(u, w) = \begin{cases} \bar{v}u & \text{if } \begin{cases} u \leq w \leq \sigma, \\ u \leq \sigma \leq w \leq 1-u, \\ w \leq u \leq \sigma, \end{cases} \\ \bar{v}(2\sigma - w) & \text{if } \begin{cases} u \leq \sigma \leq 1-u \leq w, \\ \sigma \leq u \leq w, \\ \sigma \leq w \leq u, \end{cases} \\ \bar{v}\sigma & \text{if } w \leq \sigma \leq u. \end{cases} \quad (3.9)$$

In the particular case of $\bar{v} = 1$ and $\sigma = \frac{1}{2}$, the numerical flux determines the regions depicted in Fig. 6.

Now we display all the admissible configurations in the next table 7. Note that there are only twelve possible cases, schematized in Fig. 7. The numerical flux,

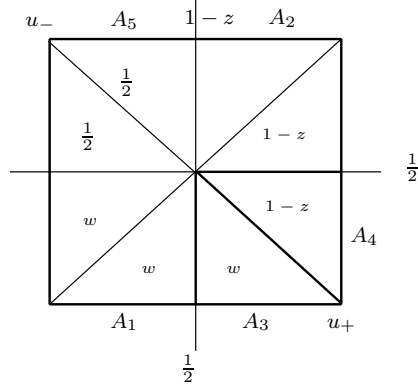


Figure 6: Numerical flux regions, A_1, A_2, A_3, A_4, A_5 .

for $\bar{v} = 1$ and $\sigma = \frac{1}{2}$, reads:

$$g^G(u, w) = \begin{cases} u & \text{if } \begin{cases} u \leq w \leq \frac{1}{2}, \\ u \leq \frac{1}{2} \leq w \leq 1 - u, \\ w \leq u \leq \frac{1}{2}, \end{cases} \\ 1 - w & \text{if } \begin{cases} u \leq \frac{1}{2} \leq 1 - u \leq w, \\ \frac{1}{2} \leq u \leq w, \\ \frac{1}{2} \leq w \leq u, \end{cases} \\ \frac{1}{2} & \text{if } w \leq \frac{1}{2} \leq u. \end{cases}$$

The scheme, under the CFL condition

$$\Delta t = \Delta x, \quad (3.10)$$

reads as

$$v_j^{n+1} = v_j^n - (g^G(v_j^n, v_{j+1}^n) - g^G(v_{j-1}^n, v_j^n)). \quad (3.11)$$

Looking at the values of the pairs (v_{j-1}^n, v_j^n) and (v_j^n, v_{j+1}^n) we have the possibilities:

- (1,1) $(v_{j-1}^n, v_j^n), (v_j^n, v_{j+1}^n) \in A_1$: $v_j^{n+1} = v_{j-1}^n$;
- (1,3) $(v_{j-1}^n, v_j^n) \in A_1, (v_j^n, v_{j+1}^n) \in A_3$: $v_j^{n+1} = v_{j-1}^n$;
- (1,4) $(v_{j-1}^n, v_j^n) \in A_1, (v_j^n, v_{j+1}^n) \in A_4$: $v_j^{n+1} = v_{j-1}^n + v_j^n + v_{j+1}^n - 1$;
- (2,2) $(v_{j-1}^n, v_j^n), (v_j^n, v_{j+1}^n) \in A_2$: $v_j^{n+1} = v_{j+1}^n$;
- (2,5) $(v_{j-1}^n, v_j^n) \in A_2, (v_j^n, v_{j+1}^n) \in A_5$: $v_j^{n+1} = \frac{1}{2}$;
- (3,2) $(v_{j-1}^n, v_j^n) \in A_3, (v_j^n, v_{j+1}^n) \in A_2$: $v_j^{n+1} = v_{j-1}^n + v_j^n + v_{j+1}^n - 1$;

- (3,5) $(v_{j-1}^n, v_j^n) \in A_3, (v_j^n, v_{j+1}^n) \in A_5: v_j^{n+1} = v_{j-1}^n + v_j^n - \frac{1}{2};$
(4,2) $(v_{j-1}^n, v_j^n) \in A_4, (v_j^n, v_{j+1}^n) \in A_2: v_j^{n+1} = v_{j+1}^n;$
(4,5) $(v_{j-1}^n, v_j^n) \in A_4, (v_j^n, v_{j+1}^n) \in A_5: v_j^{n+1} = \frac{1}{2};$
(5,1) $(v_{j-1}^n, v_j^n) \in A_5, (v_j^n, v_{j+1}^n) \in A_1: v_j^{n+1} = \frac{1}{2};$
(5,3) $(v_{j-1}^n, v_j^n) \in A_5, (v_j^n, v_{j+1}^n) \in A_3: v_j^{n+1} = \frac{1}{2};$
(5,4) $(v_{j-1}^n, v_j^n) \in A_5, (v_j^n, v_{j+1}^n) \in A_4: v_j^{n+1} = v_{j+1}^n + v_j^n - \frac{1}{2}.$

$(u, w) \backslash (w, z)$	$w \leq \frac{1}{2}$	$w \geq \frac{1}{2}$	$w \leq \frac{1}{2}$	$w \leq \frac{1}{2}$	$w \geq \frac{1}{2}$
	1	2	3	4	5
$w \leq \frac{1}{2}$ 1	YES	NO	YES	YES	NO
$w \geq \frac{1}{2}$ 2	NO	YES	NO	NO	YES
$w \geq \frac{1}{2}$ 3	NO	YES	NO	NO	YES
$w \geq \frac{1}{2}$ 4	NO	YES	NO	NO	YES
$w \leq \frac{1}{2}$ 5	YES	NO	YES	YES	NO

Figure 7: The admissible configurations.

Boundary conditions. Suppose to assign a condition at the incoming boundary $x = 0$:

$$u(0, t) = \rho_1(t) \quad t > 0$$

and study equation only for $x > 0$. Now we are considering the initial-boundary value problem (3.1)-(3.2)-(3.3) with $u_0 \in C^1(\mathbb{R}^+)$, $u_1(t) \in C^1((0, T))$, $F \in$

$C^1(\mathbb{R})$. It is not easy to find a function u that satisfies (3.3) in a classical sense, because, in general, the boundary data cannot be assumed. One seeks a condition which is to be effective only in the inflow part of the boundary. Following [1] the rigorous way of assigning the boundary condition is:

$$\max_{k \in I(u(0,t), \rho_1(t))} \left\{ \text{sgn}(u(0,t) - \rho_1(t)) [F(u(0,t)) - F(k)] \right\} = 0. \quad (3.12)$$

We practically proceed by inserting a ghost cell and defining

$$v_0^{n+1} = v_0^n - \frac{\Delta t}{\Delta x} (g^G(v_0^n, v_1^n) - g^G(u_1^n, v_0^n)), \quad (3.13)$$

where

$$u_1^n(t) = \frac{1}{\Delta t} \int_{t_n}^{t_{n+1}} \rho_1(t) dt$$

takes the place of v_{-1}^n .

An outgoing boundary can be treated analogously. Let $x < L = x_N$. Then the discretization reads:

$$v_N^{n+1} = v_N^n - \frac{\Delta t}{\Delta x} (g^G(v_N^n, u_2^n) - g^G(v_{N-1}^n, v_N^n)), \quad (3.14)$$

where

$$u_2^n(t) = \frac{1}{\Delta t} \int_{t_n}^{t_{n+1}} \rho_2(t) dt$$

takes the place of v_{N+1}^n , that is a ghost cell value.

3.2 Fast Shock Fitting (FSF) method

The main framework of this method is determined by theoretical results of Section 2. In particular, we know that for solutions starting from an empty network, taking the flux as in (2.2), there exists one separating shock (possibly latent) on each road (see Lemma 2.3). Such property allows us to construct an exact procedure for the computation of the approximate solutions. This scheme is a finite volumes method, since we consider the values of densities on the cells $C_j = [x_{j-1}, x_j)$, with $x_j = x_0 + j\Delta x$ and $j = 1, \dots, N$, instead of considering the densities on the nodes of the grid. FSF is composed by two main steps:

- 1) separating shock tracking;
- 2) shift by constant velocities.

Let us now describe the method starting from step 2, since is the simplest. Consider Riemann problems on each road, with constant states ρ_-, ρ_+ . If $\rho_-, \rho_+ < \sigma$ we have a wave moving with speed \bar{v} as showed by the case *a*) in Fig. 8. If

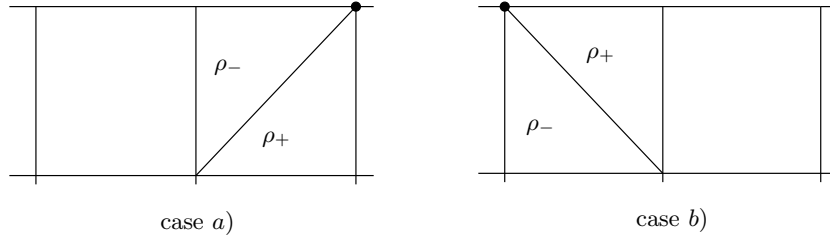


Figure 8: Riemann problems.

$\rho_-, \rho_+ > \sigma$, instead, the speed is $-\bar{v}$, as represented in Fig. 8 case b). Therefore we can proceed in the following way. The points on the left respect to σ are shifted on the right of one index:

$$\rho(t^{n+1}, x_j) = \rho(t^n, x_{j+1})$$

and, analogously, the points on the right are shifted on the left of one index:

$$\rho(t^{n+1}, x_j) = \rho(t^n, x_{j-1}),$$

with j the index of the space cells, n the index of the time nodes. For the numerical scheme, the shift reads as

$$u_j^{n+1} = u_{j+1}^n \text{ for } j > j^* \text{ (forward)} \quad \text{or} \quad u_j^{n+1} = u_{j-1}^n \text{ for } j \leq j^* \text{ (backward)}.$$

Now we describe step 1 of the algorithm. Under a suitable CFL condition which avoids the interaction of two neighbouring cells, namely $\Delta t \bar{v} = \Delta x$, we capture the separating shock as follows. The separating shock is described by the 4-tuple (x, ρ_-, ρ_+, j) , where x is the starting point, ρ_-, ρ_+ are the densities, respectively, on the left and on the right respect to x and j indicates the cell containing x . We indicate by $(x^*, \rho_-^*, \rho_+^*, j^*)$ the shock parameters after a time step. For simplicity, we consider a scaling in the parametrization of each road I_i . In particular, we set $[a_i, b_i] = [0, 1] = [0, N\Delta x]$.

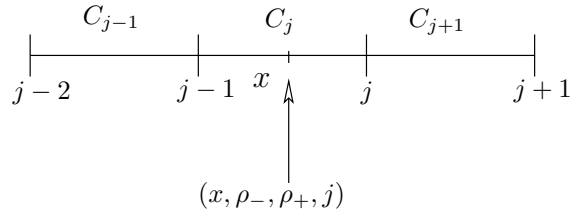


Figure 9: The big shock is described by four parameters: (x, ρ_-, ρ_+, j) .

3.2.1 Evolution: separating shock tracking

We distinguish three cases:

1. Shock inside I_i .
2. Latent shock on the left.
3. Latent shock on the right.

We set:

$$\rho_i(\cdot, x_j) = \rho_j,$$

intending that the index of the road i is fixed.

Case 1: Shock inside

We call $x_{\Delta/2}$ the position of the big shock at time $\frac{\Delta t}{2}$ if no interaction occurs:

$$x_{\Delta/2} = x + \frac{\lambda \Delta t}{2}, \quad \text{with} \quad \lambda = \frac{f(\rho_+) - f(\rho_-)}{\rho_+ - \rho_-}.$$

We consider two cases:

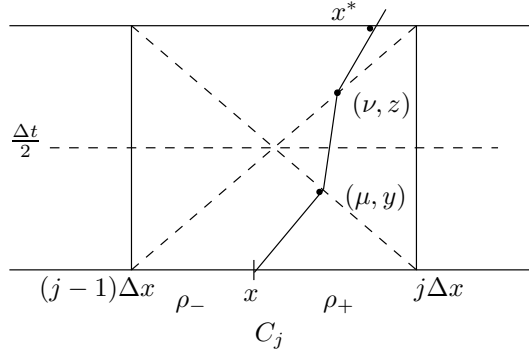


Figure 10: Possible evolution of the big shock inside the cell.

I. $x_{\Delta/2} \leq (j - \frac{1}{2}) \Delta x$;

II. $x_{\Delta/2} > (j - \frac{1}{2}) \Delta x$.

Case I.

In this case there is an interaction with the wave from $j\Delta x$ and we want to compute the interaction point, namely (μ, y) , where $\mu \in (0, \Delta t/2)$, $y \in ((j - 1)\Delta x, (j - 1/2)\Delta x)$. We get

$$y = (j - 1)\Delta x + \mu = x + \lambda\mu, \quad \text{thus} \quad \mu = \frac{x - (j - 1)\Delta x}{1 - \lambda}.$$

Then the position of the big shock at time Δt , if no other interaction occurs, would be:

$$x_\Delta = y + \lambda_1(\Delta t - \mu), \quad \text{with} \quad \lambda_1 = \frac{f(\rho_+) - f(\rho_{j-1})}{\rho_+ - \rho_{j-1}}.$$

We have two possibilities:

A. $x_\Delta \leq (j-1)\Delta x$;

B. $x_\Delta > (j-1)\Delta x$.

Case IA.

If $j = 1$ then we set $x^* = 0$ and $j^* = 0$, otherwise a second interaction verifies. After this interaction, the big shock position x^* is:

$$x^* = z + \lambda^*(\Delta t - \nu), \quad \text{with} \quad \lambda^* = \frac{f(\rho_+) - f(\rho_{j-2})}{\rho_+ - \rho_{j-2}}$$

and x^* is surely contained in the cell C_{j-1} due to its speed (< 1). The coordinates of the interaction point are (ν, z) , with $\nu \in (0, \Delta t)$, $z \in ((j-3/2)\Delta x, (j-1)\Delta x)$. We have:

$$z = (j-2)\Delta x + \nu = y + \lambda_1(\nu - \mu), \quad \text{thus} \quad \nu = \frac{(j-2)\Delta x - y + \lambda_1\mu}{\lambda_1 - 1}$$

and we let:

$$\begin{aligned} \rho_-^* &= \rho_{j-2} \\ \rho_+^* &= \rho_+, \\ j^* &= j-1. \end{aligned}$$

Case IB.

After the second interaction, the big shock position is:

$$x^* = z + \lambda^*(\Delta t - \nu), \quad \text{with} \quad \lambda^* = \frac{f(\rho_{j+1}) - f(\rho_{j-1})}{\rho_{j+1} - \rho_{j-1}}.$$

The coordinates of the interaction point are (ν, z) , with $\nu \in (0, \Delta t)$, $z \in ((j-3/2)\Delta x, (j-1)\Delta x)$. We obtain:

$$z = j\Delta x - \nu = y + \lambda_1(\nu - \mu) \quad \text{so that} \quad \nu = \frac{j\Delta x - y + \lambda_1\mu}{\lambda_1 + 1}$$

and we assign:

$$\begin{aligned} \rho_-^* &= \rho_{j-1} \\ \rho_+^* &= \rho_{j+1}, \\ j^* &= j. \end{aligned}$$

Case II.

After the first interaction, the coordinates of the intersection point are (μ, y) , with $\mu \in (0, \Delta t/2)$, $y \in ((j-1/2)\Delta x, j\Delta x)$. In particular,

$$y = j\Delta x - \mu = x + \lambda\mu, \quad \text{thus} \quad \mu = \frac{j\Delta x - x}{1 + \lambda}.$$

Then the position at time Δt is

$$x_\Delta = y + \lambda_1(\Delta t - \mu), \quad \text{with} \quad \lambda_1 = \frac{f(\rho_{j+1}) - f(\rho_-)}{\rho_{j+1} - \rho_-}.$$

We have two possibilities:

- A. $x_\Delta \leq j\Delta x$;
- B. $x_\Delta > j\Delta x$.

Case IIA.

After the second interaction, the big shock position x^* is:

$$x^* = z + \lambda^*(\Delta t - \nu), \quad \text{with} \quad \lambda^* = \frac{f(\rho_{j+1}) - f(\rho_{j-1})}{\rho_{j+1} - \rho_{j-1}}.$$

The coordinates of the interaction point are (ν, z) , with $\nu \in (0, \Delta t)$, $z \in ((j-1/2)\Delta x, j\Delta x)$. We get:

$$z = (j-1)\Delta x + \nu = y + \lambda_1(\nu - \mu), \quad \text{so that} \quad \nu = \frac{(j-1)\Delta x - y + \lambda_1\mu}{\lambda_1 - 1}$$

and we set:

$$\begin{aligned} \rho_-^* &= \rho_{j-1}, \\ \rho_+^* &= \rho_{j+1}, \\ j^* &= j. \end{aligned}$$

Case IIB.

If $j = N$ we set $x^* = N\Delta x$ and $j^* = N + 1$, otherwise we have the second interaction. After this interaction the big shock position is:

$$x^* = z + \lambda^*(\Delta t - \nu), \quad \text{with} \quad \lambda^* = \frac{f(\rho_{j+2}) - f(\rho_-)}{\rho_{j+2} - \rho_-}.$$

The coordinates of the interaction point are (ν, z) , with $\nu \in (0, \Delta t)$, $z \in (j\Delta x, (j+1)\Delta x)$. We obtain:

$$z = (j+1)\Delta x - \nu = y + \lambda_1(\nu - \mu), \quad \text{thus} \quad \nu = \frac{(j+1)\Delta x - y + \lambda_1\mu}{\lambda_1 + 1}$$

and we let:

$$\begin{aligned}\rho_-^* &= \rho_-, \\ \rho_+^* &= \rho_{j+2}, \\ j^* &= j + 1.\end{aligned}$$

Lemma 3.2 *If we have a latent shock at a_i (on the left), then S_i enters the first cell under the condition $\rho_0 < \tau(\rho_1)$.*

PROOF. We have $\rho_1 > \sigma$. To enter the first cell, the shock wave must have a positive speed, thus $\rho_0 < \tau(\rho_1)$. \square

Case 2: latent shock on the left boundary

In the case of a shock at the left boundary (as an example see Fig. 11), the big shock position at the interaction is:

$$x_\Delta = y + \lambda_1(\Delta t - \mu), \quad \text{with} \quad \lambda_1 = \frac{f(\rho_2) - f(\rho_0)}{\rho_2 - \rho_0},$$

where

$$y = \lambda\mu = \Delta x - \mu \quad \text{and} \quad \mu = \frac{\Delta x}{\lambda + 1}.$$

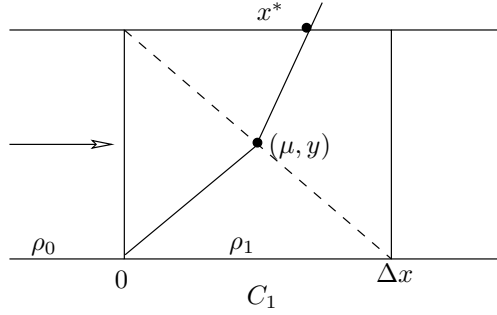


Figure 11: Latent shock on the left.

Now we have only the case $x_\Delta > 0$, since the case $x_\Delta < 0$ is already treated in IA. In order to determine the position after the interaction we set:

$$\begin{aligned}x^* &= x_\Delta, \\ \rho_-^* &= \rho_0, \\ \rho_+^* &= \rho_2, \\ j^* &= 1.\end{aligned}$$

Lemma 3.3 *If we have a latent shock at b_i (on the right), then S_i comes out the last cell under the condition $\rho_{N+1} > \tau(\rho_N)$.*

PROOF. We have $\rho_N < \sigma$. In order to exit from the last cell, the shock wave must have a negative speed, thus $\rho_{N+1} > \tau(\rho_N)$. \square

Case 3: latent shock on the right boundary

If we have a shock at the right boundary (as an example see Fig. 12), shock position at the interaction can be determined in the following way:

$$x_\Delta = y + \lambda_1(\Delta t - \mu), \quad \text{with} \quad \lambda_1 = \frac{f(\rho_{N+1}) - f(\rho_{N-1})}{\rho_{N+1} - \rho_{N-1}},$$

where

$$y = N\Delta x + \lambda\mu = (N-1)\Delta x + \mu \quad \text{and} \quad \mu = \frac{-\Delta x}{\lambda - 1}.$$

Now we have only the case $x_\Delta < 1$, since the case $x_\Delta > 1$ is already treated in IIB. We determine the position after interactions:

$$\begin{aligned} x^* &= x_\Delta, \\ \rho_-^* &= \rho_{N-1}, \\ \rho_+^* &= \rho_{N+1}, \\ j^* &= N. \end{aligned}$$

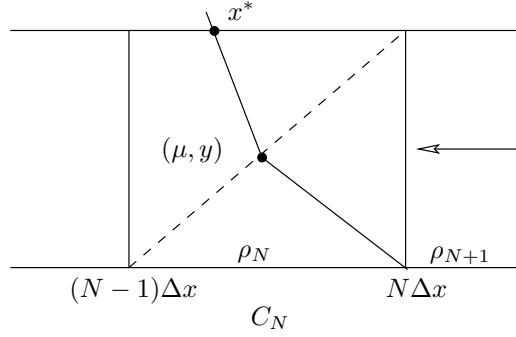


Figure 12: Latent shock on the right.

Boundary conditions. If there is no latent shock on the left, we set:

$$u_0^n = \min\{u_0^n, \sigma\}, \quad \text{if } j^* \geq 2,$$

since in this case $u_1^n < \sigma$ and we need to have a positive speed of the wave.

If there is no latent shock on the right, we put:

$$u_{N+1}^n = \max\{u_{N+1}^n, \sigma\}, \quad \text{if } j^* \leq N - 1,$$

since $u_N^n > \sigma$ and we want the wave to have a negative speed. At the outgoing boundary it can also be applied a Neumann condition. Finally, the structure of the FSF method at each iteration in time can be schematized as in Fig. 13.

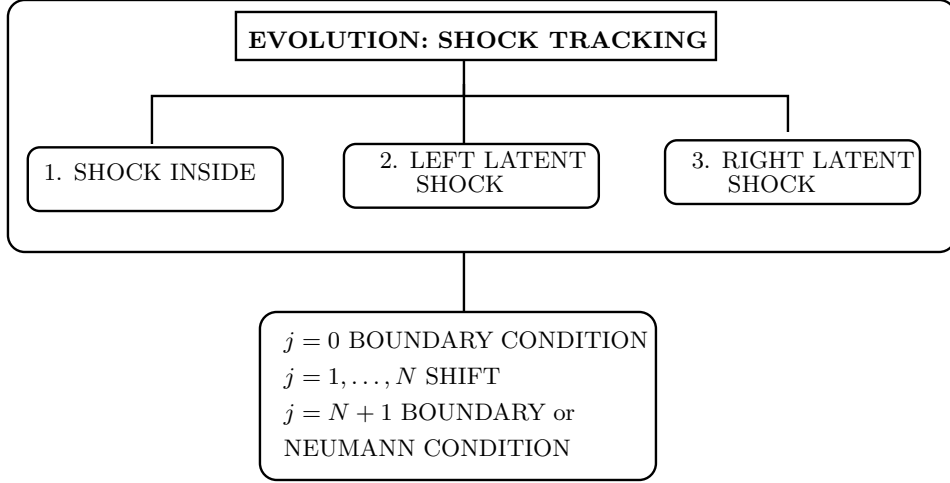


Figure 13: Schematization of FSF at each time step.

4 Numerical experiments

Next we present some tests produced by a simulation program based on the numerical schemes described in the previous Section.

Example 4.1 *In the numerical tests we consider the following initial data:*

- $$\rho(0, x) = 0.7, \tag{4.1}$$

- $$\rho(0, x) = 0. \tag{4.2}$$

We take the boundary datum below:

- *constant*
$$\rho_b(t) = 0.15; \tag{4.3}$$

- *Brownian-motion boundary datum*

$$\begin{cases} W(0) = 0, \\ W(t) = W(t - \Delta t) + \mathcal{N}(0, 1)\sqrt{\Delta t}, \quad t > 0. \end{cases} \quad (4.4)$$

4.1 Orders and errors

We introduce the formal numerical order γ of a numerical method for a single road in the following way:

$$\gamma = \log_2 \left(\frac{e(1)}{e(2)} \right). \quad (4.5)$$

The L^1 -error on each road is

$$e(p) = \frac{h}{p} \sum_{j=0, \dots, pN} \left| w_j^{pM} \left(\frac{h}{p} \right) - w_{2j}^{pM} \left(\frac{h}{2p} \right) \right| \quad p = 1, 2, \quad (4.6)$$

where $w_m^M(h)$ denotes the numerical solution obtained with the space step discretization equal to h , calculated in x_m at the final time $t_M = T$. The quantity $w_m^M(h)$ denotes the numerical solution obtained with the space step discretization equal to h , computed in x_m at the final time $t_M = T$. In the following table T1 we used the exact solution to compute L^1 errors of the approximation schemes.

h	FG		K3V		FSF	
	γ	L^1 Error	γ	L^1 Error	γ	L^1 Error
0.2	3.3	5.000e-02	0.8	7.000e-02	1.0	1.100e-01
0.1	1.0	5.000e-03	1.4	4.000e-02	-	5.500e-02
0.05	1.0	2.500e-03	1.6	1.500e-02	-	0.000e+00
0.025	-1.3	6.250e-03	0.2	5.000e-03	-	0.000e+00

TABLE T1: Orders and errors for FG, K3V and FSF for data (4.1), with boundary data (4.3), $T = 2.5$.

h	FG		K3V		FSF	
	γ	L^1 Error	γ	L^1 Error	γ	L^1 Error
0.2	0.3	5.510e-01	0.0	4.713e-01	-0.2	3.713e-01
0.1	2.2	4.559e-01	2.3	4.661e-01	2.4	4.336e-01
0.05	1.3	9.936e-02	1.1	9.204e-02	1.0	8.326e-02
0.025	-	4.133e-02	-	4.133e-02	-	4.133e-02

TABLE T2: Orders and errors for FG, K3V, FSF for initial data (4.2), with boundary data (4.4), $T = 3$.

Recalling that numerical order is computed by the approximate formula (4.5), whenever it is undefined we put the symbol “-” into the tables above.

4.2 Accuracy: correct reconstruction of solutions

In the next Figures 14 and 15, we compare the solutions obtained by the Fast Godunov scheme (FG), by the 3-velocities Kinetic scheme of second order (K3V) and by the Fast Shock Fitting method (FSF) for different space (and time) steps.

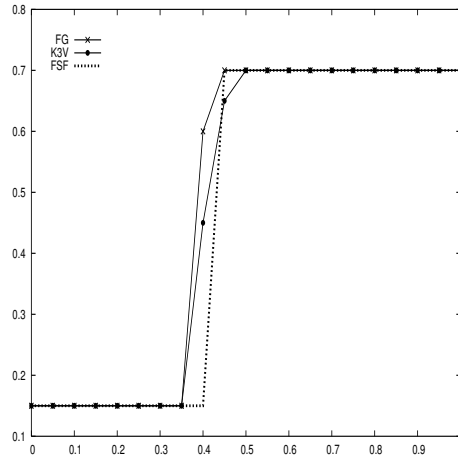


Figure 14: Approximate solution computed by FG, K3V and FSF, with initial data (4.1) and boundary data (4.3), $t = 1.45$, $h = 0.05$.

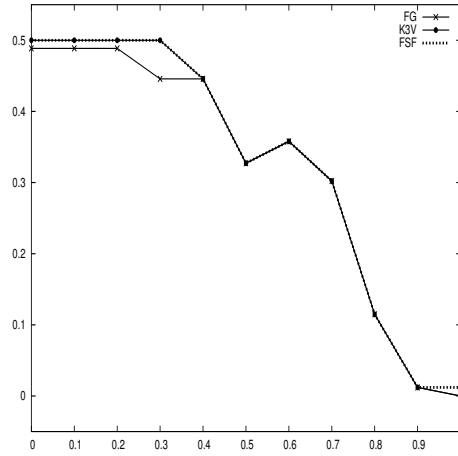


Figure 15: Approximate solution computed by FG, K3V and FSF, with initial density $\rho(0, x) = 0$ and boundary data of Brownian-motion type (4.4), $t = 2.2$, $h = 0.1$.

The solution is well reconstructed by the FSF scheme, even in case of periodic data or in the presence of Brownian data at the boundary, where the best approximation is provided by FSF, see Fig. 14 and Fig. 15.

4.3 CPU time

Now we are interested in the analysis of the CPU time. In particular, we want to compare the performances of programs based on the following approximation schemes: the classical Godunov scheme (G), the Fast Godunov scheme (FG), the 3-velocities Kinetic scheme of second order (K3V) and the Fast Shock Fitting method (FSF).

Let us consider a network composed by 1000 roads that do not meet at any junction. In table T4 we can see the CPU time of execution for $T = 10$, $T = 30$ for initial data (4.1). In table T6 is reported the time of execution for $T = 10$, $T = 30$ for initial zero data and boundary data of Brownian-motion type (4.4). We successively consider a network composed by 5000 roads that do not meet at any junction. In table T5 we can see the CPU time of execution for $T = 10$, $T = 30$ for initial data (4.1), while in table T7 is reported the time of execution for $T = 10$, $T = 30$ taking again initial zero data and boundary data of Brownian-motion type (4.4). In the next tables the time of execution is computed in seconds (s).

CPU time				
$T = 10$				
h	G	FG	K3V	FSF
0.2	0.28 s	0.13 s	5.60 s	0.06 s
0.1	1.03 s	0.44 s	20.40 s	0.15 s
0.05	3.81 s	1.54 s	77.19 s	0.40 s
0.025	14.78 s	5.71 s	302.07 s	1.19 s
$T = 30$				
h	G	FG	K3V	FSF
0.2	0.84s	0.41 s	16.67 s	0.18 s
0.1	3.05 s	1.32 s	60.84 s	0.45 s
0.05	11.42 s	4.62 s	230.62 s	1.19 s
0.025	44.07 s	17.01 s	822.50 s	3.60 s

TABLE T4: CPU time for the schemes G, FG, K3V and FSF for data (4.1), boundary data (4.3), $T = 10$, $T = 30$, 1000 roads.

CPU time				
$T = 10$				
h	G	FG	K3V	FSF
0.2	1.42 s	0.68 s	28.16 s	0.30 s
0.1	5.12 s	2.19 s	102.03 s	0.77 s
0.05	19.09 s	7.61 s	386.47 s	2.00 s
0.025	73.42 s	27.70 s	1498.31 s	5.99 s
$T = 30$				
h	G	FG	K3V	FSF
0.2	4.20 s	2.07 s	82.94 s	0.91s
0.1	15.25 s	6.54 s	303.69 s	2.27 s
0.05	56.95 s	22.80 s	1192.796 s	6.05 s
0.025	219.73 s	84.19 s	4500.04 s	17.90 s

TABLE T5: CPU time for the schemes G, FG, K3V and FSF for data (4.1), boundary data (4.3), $T = 10$, $T = 30$, 5000 roads.

CPU time				
$T = 10$				
h	G	FG	K3V	FSF
0.2	0.35 s	0.22 s	6.03 s	0.12 s
0.1	1.14 s	0.61 s	21.44 s	0.27 s
0.05	3.97 s	1.84 s	80.12 s	0.64 s
0.025	14.79 s	6.29 s	295.84 s	1.67 s
$T = 30$				
h	G	FG	K3V	FSF
0.2	1.07 s	0.66 s	17.50 s	0.37 s
0.1	3.42 s	1.84 s	62.75 s	0.81 s
0.05	12.00 s	5.50 s	237.42 s	1.91 s
0.025	44.90 s	17.55 s	900.34 s	5.01 s

TABLE T6: CPU time for the schemes G, FG, K3V and FSF for null initial data (4.2) and Brownian-motion boundary data (4.4), $T = 10, T = 30$, 1000 roads.

CPU time				
$T = 10$				
h	G	FG	K3V	FSF
0.2	1.78 s	1.12 s	29.37 s	0.60 s
0.1	5.68 s	3.05 s	104.74 s	1.35 s
0.05	19.83 s	9.30 s	394.03 s	3.20 s
0.025	73.86 s	31.40 s	1515.32 s	8.39 s
$T = 30$				
h	G	FG	K3V	FSF
0.2	5.34 s	3.32 s	85.71 s	1.83 s
0.1	17.11 s	9.12 s	309.55 s	4.05 s
0.05	59.95 s	27.93 s	1171.10 s	9.58 s
0.025	223.03 s	95.38 s	4527.80 s	25.07 s

TABLE T7: CPU time for the schemes G, FG, K3V and FSF for null initial data (4.2) and Brownian-motion boundary data (4.4), $T = 10, T = 30$, 5000 roads.

From the analysis of the previous tables it is easy to see that the Fast Godunov scheme (FG) performs better than the classical scheme. Moreover, we have that FSF performs better than FG and K3V as regards as the CPU time. Notice that in tables T6 and T7 the CPU time has an increment of about 0.50 seconds for FG and FSF respect to CPU times reported in tables T4 and T5, due to the fact that boundary data (4.4) is loaded at each time step.

All the simulations have been performed by a personal computer, processor AMD Athlon XP 2400 Mhz, RAM 512 Mb.

5 Conclusions

This paper introduces two new algorithms, Fast Shock Fitting (FSF) and Fast Godunov (FG), to compute solutions for traffic flow on a network for a piecewise constant flux function as in (2.2). Our tests show that FSF, whether applicable, represents an efficient and good approximation scheme for this problem. FSF and 3-velocities Kinetic scheme of second order (K3V) substantially provide the same accuracy, thus, whether applicable, FSF is preferable regarding the speed of computation. FSF, indeed, results to be faster than FG and K3V and at the same time more accurate. The speed of computation represents an important aspect, also in consideration of the fact that road networks of cities can be composed by a considerably large number of roads. On the other hand, FG results faster than the classical Godunov scheme.

References

- [1] C. Bardos, A. Y. Le Roux, J. C. Nédélec, First Order Quasilinear Equation with Boundary Conditions, *Commun. Partial Differential Equations*, **4**, pp. 1017-1034 (1979).
- [2] G. Bretti, R. Natalini, B. Piccoli, Numerical Approximations of a Traffic Flow Model on Networks, *Networks and Heterogeneous Media*, **1** (2006) (to appear).
- [3] Y. Chitour, B. Piccoli, Traffic circles and timing of traffic lights for cars flow, *Discrete and Continuous Dynamical Systems-Series B*, **5**, 599-630 (2005).
- [4] G. M. Coclite, M. Garavello, B. Piccoli, Traffic Flow on a Road Network, *Siam Math. Anal.*, **36**, 1862-1886 (2005).
- [5] C.M. Dafermos, Hyperbolic conservation laws in continuum physics. Grundlehren der Mathematischen Wissenschaften. 325. Berlin: Springer (2000).
- [6] C. D'Apice, R. Manzo, B. Piccoli, Packets Flows on a Telecommunication Network, (Preprint).
- [7] S.K. Godunov, A finite difference method for the numerical computation of discontinuous solutions of the equations of fluid dynamics, *Mat. Sb.* **47**, pp. 271-290 (1959).
- [8] E. Godlewski, P. A. Raviart, Hyperbolic systems of conservation laws, *Mathématiques & Applications [Mathematics and Applications]*, 3/4. Ellipses, Paris (1991).
- [9] M. J. Lighthill, G. B. Whitham, *On kinetic waves. II. Theory of Traffic Flows on Long Crowded Roads*, *Proc. Roy. Soc. London Ser. A*, **229**, pp. 317-345 (1955).

- [10] R. Natalini, A Discrete Kinetic Approximation of Entropy Solutions to Multidimensional Scalar Conservation Laws, *Journal of differential equations*, **148**, pp. 292-317 (1998).
- [11] P. I. Richards, *Shock Waves on the Highway*, Oper. Res., **4**, pp. 42-51 (1956).