Numerical methods for nonlocal conservation laws

Filmon Tesfamikael Misgane

October 2025

Contents

Preface

1 Introduction

Since many interesting conservation laws that arise in applications (e.g. traffic simulation) are nonlinear, it is impossible to obtain explicit solution formulas. Hence, we rely on designing numerical methods to approximate the solutions in the best possible way. When designing such methods it is natural to ask ourselves how reliable those methods could be. That is significant to study the stability and convergence of the numerical methods.

For the purpose of numerical experimentation we focus on the specific one dimensional traffic flow model which is modeled by a nonlocal conservation law. In other words the goal is to design an efficient numerical method and ultimately to prove that the method convergences to the correct solution.

2 Preliminaries

2.1 Scalar local and nonlocal conservation laws

Scalar conservation laws are partial differential equations that can be written on the form

$$\begin{aligned} \partial_t u + \nabla \cdot f(u) &= 0 \\ u(x,0) &= u_0(x) \end{aligned} \tag{1}$$

where $\partial_t = \frac{\partial}{\partial_t}$ is temporal differentiation, $\nabla\cdot$ is divergence operator, $u = u(x,t)$ is the unknown function of x and t and f is a given flux function.

If we now let $f(u) = uv(u)$ where $v(u) = 1 - u$, then Eq. (1) can be used for modelling traffic flow. For a detailed setup of this model, we refer to [3]. This model is however based only on local information. when a car driver usually decides his speed, it depends on the traffic information within a road segment of length $\epsilon > 0$ ahead of the car's current location. For managing that case we look for a better way to model the velocity, so we need a nonlocal model which is on the following form

$$\begin{aligned} \partial_t u + \nabla \cdot (uV(u)) &= 0 \\ u(x,0) &= u_0(x) \end{aligned} \tag{2}$$

where now V is a nonlocal operator, say,

$$\begin{aligned} V(u)(x,t) &= v(u * \omega_\epsilon) \\ &= 1 - \int_0^\epsilon \omega_\epsilon(y) u(x+y,t)\, dy \end{aligned} \tag{3}$$

for some nonlocal kernel $\omega_\epsilon : [0,\epsilon] \to \mathbb{R}$ satisfying $\omega_\epsilon \geq 0$ and $\int_0^\epsilon \omega_\epsilon(y)\, dy = 1$. Formally speaking, $V(u) \to v(u)$ as $\epsilon \to 0$, but whether the corresponding solutions u_ϵ of (2) converge to a solution of (1), is the key question that has been studied in the literature (Citation is needed here).

In conclusion, let us denote (1) and (2) the local and nonlocal model, respectively. We are then aiming to approximate solutions of the local model by solutions of the nonlocal model, where the solution $u(x,t)$ represents the density of cars which is the number of cars per square meter.

2.2 Solutions to the Riemann problem

For numerical experiments we need entropy solutions for different initial data. Hence, we briefly mention the notion of weak solution and entropy solution.

Definition 2.1 (Weak solutions) *A function $u \in L^\infty(\mathbb{R} \times \mathbb{R}_+)$ is a weak solution of (1) with initial data $u_0 \in L^\infty(\mathbb{R})$ if the following identity holds for all test functions $\phi \in C^1(\mathbb{R} \times \mathbb{R}_+)$*

$$\int_{\mathbb{R}_+} \int_{\mathbb{R}} u\phi_t + f(u)\phi_x \, dx\, dt + \int_{\mathbb{R}} u_0 \phi(x,0)\, dx = 0. \tag{4}$$

Since weak solutions are not unique, some additional conditions have to be imposed in order to pick a weak solution that describes the physical flow correctly. One such condition is entropy condition. Detailed characteristics of the entropy solution can be found in [4].

We now look at the explicit solutions for the Riemann problem. The Riemann problem is the initial value problem

$$\begin{aligned} \partial_t u + \nabla \cdot f(u) &= 0 \\ u(x,0) &= \begin{cases} u_l & \text{if } x < 0 \\ u_r & \text{if } x > 0 \end{cases} \end{aligned} \tag{5}$$

Solution of this equation is constructed as

$$u(x,t) = \begin{cases} u_l & \text{for } x \le h'(u_l)t \\ (h')^{-1}(\frac{x}{t}) & \text{for } h'(u_l)t \le x \le h'(u_r)t \\ u_r & \text{for } x \ge h'(u_r)t \end{cases} \tag{6}$$

where $h(u)$ is defined by

$$h(u) = \begin{cases} \sup\{g(u) \mid g \le f \text{ and } g \text{ is convex on } [u_l, u_r]\} & \text{if } u_l < u_r, \\ \inf\{g(u) \mid g \ge f \text{ and } g \text{ is concave on } [u_r, u_l]\} & \text{if } u_r < u_l. \end{cases} \tag{7}$$

3 Numerical method

3.1 Finite volume scheme for general f

Designing the finite difference method consists of four steps:

1. Discretizing the domain,
2. Satisfying the equation at discrete points,
3. Replacing derivatives by finite differences,
4. Solving the discretized problem.

If we take a look at step 3, it requires that solutions have to be smooth sufficiently. However, solutions of conservations laws may not be differentiable or even continuous. For this and other reasons, we have to adapt the steps above to design a reasonable numerical method.

3.2 Discretizing the domain and control volumes

For simplicity, we consider a uniform discretization of both the one-dimensional spatial domain $[x_L, x_R]$ and the temporal domain $[0, T]$. We split the spatial

domain $\Omega = [x_L, x_R]$ into K smaller subdomains Ω^j, such that

$$\begin{aligned}\Omega &= \bigcup_{j=0}^{K-1} \Omega^j \\ &= \bigcup_{j=0}^{K-1} [x_L + j\Delta x,\ x_L + (j+1)\Delta x],\end{aligned} \tag{8}$$

where $\Delta x = \dfrac{x_R - x_L}{K}$.

We denote the midpoint of each Ω^j by

$$x_j = x_L + \Big(j + \tfrac{1}{2}\Big)\Delta x, \quad j = 0, \ldots, K-1,$$

and use each midpoint value to define the computational cells or control volumes:

$$\mathcal{C}_j = [x_{j-\frac{1}{2}},\ x_{j+\frac{1}{2}}),$$

where the left interface $x_{j-\frac{1}{2}}$ and the right interface $x_{j+\frac{1}{2}}$ of $\mathcal{C}_j$ are given by

$$x_{j-\frac{1}{2}} = x_j - \tfrac{\Delta x}{2}, \quad x_{j+\frac{1}{2}} = x_j + \tfrac{\Delta x}{2}.$$

We have now split Ω into K smaller non-overlapping subdomains called control volumes $\mathcal{C}_j$. These control volumes are essential for the finite volume method, as pointwise evaluation does not make sense.

Setting $\Delta t = \frac{T}{N}$, we also discretize the temporal domain $[0, T]$ as

$$0 = t^0 < t^1 < \cdots < t^N = T, \quad \text{where } t^n = n\Delta t,\ n = 0, \ldots, N.$$

3.3 Cell averages and a finite volume scheme

As solutions of conservations laws may be discontinuous, pointwise evaluation does not make sense. Instead, at each time level t^n, we take cell averages:

$$u(x_j, t^n) = u_j^n \approx \frac{1}{\Delta x} \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} u(x, t^n)\, dx \tag{9}$$

Assuming now the cell averages are known at some time level t^n, we find the cell averages at next time level t^{n+1} as follows:
Taking an integral of (1) over the domain $[x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}] \times [t^n, t^{n+1}]$

$$\int_{t^n}^{t^{n+1}} \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} u_t\, dx\, dt + \int_{t^n}^{t^{n+1}} \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} f(u)_x\, dx\, dt = 0$$

And applying the fundamental theorem of calculus yields (Clarification is needed on why this theorem is applicable)

$$\begin{aligned} &\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} u(x,t^{n+1})\,dx - \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} u(x,t^{n})\,dx \\ &= -\int_{t^n}^{t^{n+1}} f(u(x_{j+\frac{1}{2}},t))\,dt + \int_{t^n}^{t^{n+1}} f(u(x_{j-\frac{1}{2}},t))\,dt \end{aligned} \tag{10}$$

Defining the numerical fluxes

$$\bar{F}_{j+\frac{1}{2}} = \frac{1}{\Delta t}\int_{t^n}^{t^{n+1}} f(u(x_{j+\frac{1}{2}},t))\,dt \tag{11}$$

and dividing both sides of (10) by Δx, we obtain

$$u_j^{n+1} = u_j^n - \frac{\Delta t}{\Delta x}(\bar{F}_{j+\frac{1}{2}} - \bar{F}_{j-\frac{1}{2}}) \tag{12}$$

Note that the relation in (12) is not explicit, since $\bar{F}$ requires a priori knowledge of the exact solution. By introducing an appropriate approximation of the numerical flux $\bar{F}$ by F, we obtain the finite volume scheme for a general flux f:

$$u_j^{n+1} = u_j^n - \frac{\Delta t}{\Delta x}(F_{j+\frac{1}{2}} - F_{j-\frac{1}{2}}) \tag{13}$$

3.4 Finite Volume Scheme for the Nonlocal Model

Let the flux be defined as $f = uV$, where V is the nonlocal operator given in (3). There are several ways to specify the nonlocal numerical fluxes. In this work, we focus on a Lax-Friedrichs type scheme; see the approaches in [1] and [2]. let us set

$$V_j^n = 1 - \Delta x \sum_{k=0}^{m-1} w_k u_{j+k}^n, \tag{14}$$

Where $\{w_k\}_{k=0}^{m-1}$ is a set of numerical quadrature weights that are used to approximate the nonlocal kernel ω_ϵ and $m = \lceil \frac{\epsilon}{\Delta x} \rceil$ (Clarification is needed on what does $\lceil\rceil$ define) is the number of cells involved in V.
We now take the numerical fluxes in (13) as following:

$$F_{j-\frac{1}{2}} = \frac{1}{2}u_{j-1}^n V_{j-1}^n + \frac{1}{2}u_j^n V_j^n + \frac{\alpha}{2}(u_{j-1}^n - u_j^n) \tag{15}$$

and

$$F_{j+\frac{1}{2}} = \frac{1}{2}u_j^n V_j^n + \frac{1}{2}u_{j+1}^n V_{j+1}^n + \frac{\alpha}{2}(u_j^n - u_{j+1}^n) \tag{16}$$

Where $\alpha > 0$ is a numerical viscosity constant and the numerical quadrature weights w_k involved in (14) can be given by
left endpoints

$$w_k = \omega_\epsilon(k\Delta x)\Delta x \text{ for } k = 0, \cdots, m-1,$$

normalized left endpoint

$$w_k = \frac{\omega_\epsilon(k\Delta x)\Delta x}{\sum_{k=0}^{m-1}\omega_\epsilon(k\Delta x)\Delta} \text{ for } k = 0, \cdots, m-1$$

and exact quadrature

$$w_k = \int_{kh}^{min\{(k+1)\Delta x, \epsilon\}} \omega_\epsilon(y)\, dy \text{ for } k = 0, \cdots, m-1$$

One may ask whether the choice of numerical quadrature will result in different outcomes when implementing the finite volume scheme (13) together with the numerical fluxes (15) and (16). That is, whether the scheme converges for some numerical quadratures but fails to converge for others.
This question is addressed in [1] through numerical experiments and mathematical explanations. We will return to this question in the next section, but for now we consider two main aspects related to the finite volume scheme: the boundary conditions and stability.

3.5 The Boundary Conditions

The algorithm for solving (2) using the finite volume scheme given in (13) together with (15) and (16) becomes:

STEP 1. Initialize u_j^0 by setting:

$$u_j^0 = \frac{1}{\Delta x}\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} u_0(x)\, dx, \quad \forall j \in \{0, \ldots, K-1\}.$$

STEP 2. For $n = 0, \ldots, N-1,$ compute:

$$\begin{aligned} u_j^{n+1} &= u_j^n - \frac{\Delta t}{\Delta x}\big(F_{j+\frac{1}{2}} - F_{j-\frac{1}{2}}\big) \\ &= \frac{1}{2}\big(u_{j+1}^n V_{j+1}^n - u_{j-1}^n V_{j-1}^n\big) - \frac{\alpha}{2}\big(u_{j+1}^n + u_{j-1}^n\big) + \alpha u_j^n. \end{aligned}$$

The second step is also supposed to hold for all $j \in \{0, \ldots, K-1\}$. However, for $j = 0$ we obtain:

$$u_0^{n+1} = \frac{1}{2}\big(u_1^n V_1^n - u_{-1}^n V_{-1}^n\big) - \frac{\alpha}{2}\big(u_1^n + u_{-1}^n\big) + \alpha u_0^n,$$

and for $j = K-1$:

$$u_{K-1}^{n+1} = \frac{1}{2}\big(u_K^n V_K^n - u_{K-2}^n V_{K-2}^n\big) - \frac{\alpha}{2}\big(u_K^n + u_{K-2}^n\big) + \alpha u_{K-1}^n.$$

We observe that for $j = 0$ and $j = K-1$, the scheme involves undefined terms such as u_{-1}, V_{-1}, u_K, and V_K. Therefore, without additional information the finite volume scheme is valid only for internal nodes $j \in \{1, \ldots, K-2\}$.

The above discussion indicate that specifying boundary conditions are needed in order to find the values u_0^{n+1} and u_{K-1}^{n+1}. Well, there are several options for specifying the boundary conditions to the spatial domain $[x_L, x_R]$ such us Artifcial, Dirichlet, Neumann and Periodic. We consider here Artifcial boundary conditions for the remaining see [Citation is needed here]

3.6 The Stability of the Finite Volume Scheme

4 Numerical Experiments

4.1 Comparison of Left Endpoints and Normalized Left Endpoints

We compare the quadrature weights obtained from the left endpoints and the normalized left endpoints.

For the Riemann problem (5), we choose $u_L = 0.6$ and $u_R = 0.1$ such that the problem becomes to solve

$$\begin{aligned} u_t + f(u)_x &= 0 \\ u(x,0) &= \begin{cases} 0.6 & \text{if } x < 0 \\ 0.1 & \text{if } x > 0 \end{cases} \end{aligned} \tag{17}$$

Since $f(u) = u(1-u)$ is a convex function, h in (7) equals f itself. Thus, the entropy solution is given by:

$$u(x,t) = \begin{cases} 0.6 & \text{for } x \leq -0.2t, \\ \dfrac{t-x}{2t} & \text{for } -2t \leq x \leq 0.8t, \\ 0.1 & \text{for } x \geq 0.8t. \end{cases} \tag{18}$$

We plot u at $t = 1$ over the domain $[x_L, x_R] = [-2, 2]$ together with both the local and nonlocal numerical approximations. The linear decreasing kernel $\omega_\epsilon(y) = \frac{2(\epsilon - y)}{\epsilon^2}$ is used. The parameters $\alpha, \Delta x$ and Δt are set to $2, 0.004$ and 0.002(Clarification is needed on why the parameters are sufficiently), respectively. After running the code for $\epsilon = 0.01$, the results are as follows.

5 Convergence of the numerical method to entropy solution

References

[1] S. Blandin and P. Goatin. "Well-posedness of a conservation law with nonlocal flux arising in traffic flow modeling". In: *Numerische Mathematik* 132 (2016), pp. 217–241.

[2] Kuang Huang and Qiang Du. “Asymptotic Compatibility of a Class of Numerical Schemes for a Nonlocal Traffic Flow Model”. In: *SIAM Journal on Numerical Analysis* 62.3 (2024), pp. 1119–1144.

[3] Ulrik Skre Fjordholm Siddhartha Mishra and Rémi Abgrall. “Numerical methods for conservation laws and related equations”. In: - - (-), p. 21.

[4] Ulrik Skre Fjordholm Siddhartha Mishra and Rémi Abgrall. “Numerical methods for conservation laws and related equations”. In: - - (-), p. 29.