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,

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

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 solution) *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 properties 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

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

where $h(u)$ is defined by

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

3 Numerical Method

3.1 Finite Volume Scheme for a General f

Designing a finite difference method typically 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.

However, step 3 requires that solutions are sufficiently smooth, but solutions of conservation laws may not be differentiable or even continuous. In addition, the unknown solution u of a conservation law may represent a density (e.g., the average number of cars per m^2), in this case step 2 is not flexible. For these reasons, we must adapt the steps above to design a suitable 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

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

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

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

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 Lax-Friedrichs Scheme of 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 scheme; see the approaches in [1] and [2].
If we now define the numerical fluxes in (13) as

$$\begin{aligned} F_{j-\frac{1}{2}} &= \frac{1}{2}(u_{j-1}^n V_{j-1}^n + u_j^n V_j^n) + \frac{\alpha}{2}(u_{j-1}^n - u_j^n) \\ F_{j+\frac{1}{2}} &= \frac{1}{2}(u_j^n V_j^n + u_{j+1}^n V_{j+1}^n) + \frac{\alpha}{2}(u_j^n - u_{j+1}^n) \end{aligned} \tag{14}$$

where $\alpha > 0$ is the numerical viscosity, we establish the Lax-Friedrichs scheme designed for the nonlocal model (2).
Mark that V_j^n is the approximation of of the nonlocal density $V(u)$ given by

$$V_j^n = 1 - \Delta x \sum_{k=0}^{m-1} w_k u_{j+k}^n, \tag{15}$$

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 $m = \lceil \frac{\epsilon}{\Delta x} \rceil$ is the number of cells involved in V.

The numerical quadrature weights w_k involved in (15) can be given by left endpoint

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

normalized left endpoint

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

and exact quadrature

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

One may ask whether the choice of numerical quadrature will result in different outcomes when implementing the Lax-Friedrichs scheme. That is, whether the scheme converges for some numerical quadrature weights 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 The Lax-Friedrichs scheme 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:

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

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

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

and for $j = K-1$:

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

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 $j \in \{1, \dots, K - 2\}$.

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}.

There are several options for specifying the boundary conditions on the spatial domain $[x_L, x_R]$, but the choice depends on whether the local model (1) is posed as an initial value problem (IVP) or a boundary value problem (BVP).

- **Initial Value Problem (IVP):** Artificial boundary conditions are required to handle the truncated domain $[x_L, x_R]$. Truncation is necessarily for initial value problem on the entire real line $\mathbb{R}$ as computational domain needs to be bounded.
- **Boundary Value Problem (BVP):** Common choices include Dirichlet, Neumann, and Periodic boundary conditions.

In this work, we focus on **artificial boundary conditions**. For a detailed discussion of the remaining types, see Citation needed here. In this case we set the values u_0^{n+1} and u_{K-1}^{n+1} to

$$u_0^{n+1} = u_1^{n+1} \quad \text{and } u_{K-1}^{n+1} = u_{K-2}^{n+1}$$

That means the complete algorithm for solving (2) using the Lax-Friedrichs scheme becomes:

STEP 1. Initialize u_j^0 by setting:

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

STEP 2. For $n = 0, \dots, N - 1$ and $j = 0, \dots, K - 1$, compute:

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

set

$$u_0^{n+1} = u_0^n \text{ and } u_{K-1}^{n+1} = u_{K-2}^n$$

3.6 The Stability of the Lax-Friedrichs scheme

We must make sure that the approximation solution obtained form the Lax-Friedrichs scheme shares similar properties as the entropy solution. Since the Lax-Friedrichs scheme may devolve undesired oscillations, we must impose a condition on the parameters Δx, Δt and α. So, we choose the parameters such that

$$\Delta t \leq \frac{\Delta x}{\alpha} \tag{16}$$

in order to obtain stability.

4 Numerical Experiments

In the previous section we have devolved the Lax-Friedrichs scheme and described the complete algorithm on how to implement the scheme. In this section we consider if the numerical solutions converge to the correct solution numerically.

4.1 Comparison of Left endpoint and Normalized Left endpoint

We start by comparing the quadrature weights obtained from the left endpoint and the normalized left endpoint . 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:

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

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 and the viscosity coefficient β is set to 2.

Numerical experiment 1a:

We begin by implementing the finite volume scheme designed in the previous section for $\epsilon = 0.1$ and $\Delta x = 0.02$

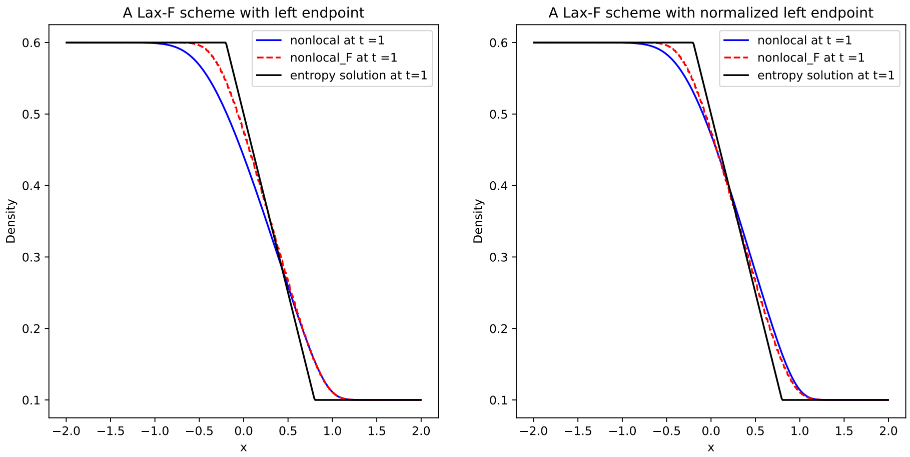

Figure 1: Numerical solutions and entropy solution at t = 1 corresponding to the left endpoint numerical quadrature weights (left) and the normalized left endpoint numerical quadrature weights (right).

As shown in Figure 1, the choice of numerical quadrature weights does not lead to a significant difference between the numerical solutions and the entropy solution. Nevertheless, a low level of accuracy remains at the shocks, where the numerical solutions deviate noticeably from the entropy solution. We aim to achieve convergence of the numerical solutions to the entropy solution as Δx and ϵ tend to zero simultaneously.

Numerical experiment 1b:
Next we examine the implementation for $\epsilon = 0.01$ and $\Delta x = 0.01$

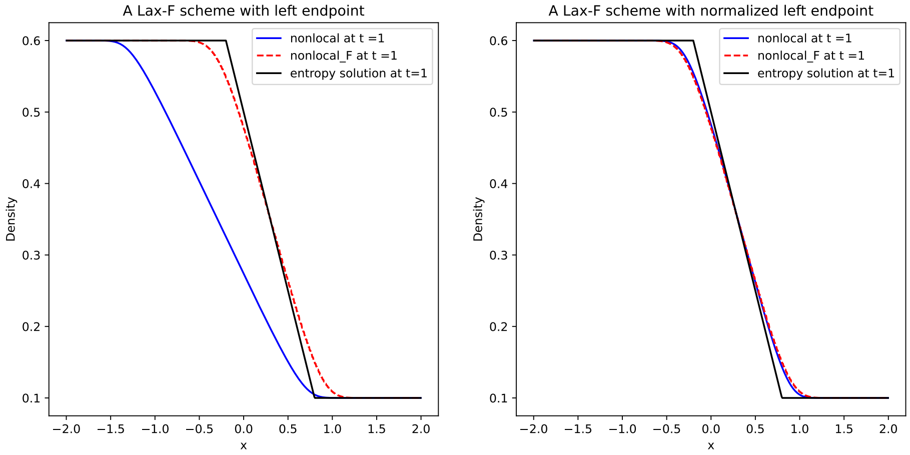

Figure 2: Numerical solutions and entropy solution at t = 1 corresponding to the left endpoint numerical quadrature weights (left) and the normalized left endpoint numerical quadrature weights (right).

As observed in Figure 2, contrary to our expectations, the left endpoint numerical quadrature weights cause the numerical solution of the nonlocal model to deviate further from both the numerical solution of the local model and the entropy solution. In contrast, the normalized left endpoint quadrature weights lead to improved accuracy and better alignment between the numerical solutions and the entropy solution.

Numerical experiment 1b:

Even when taking smaller parameter values, $\epsilon = 0.001$ and $\Delta x = 0.004$, the numerical solution of the nonlocal model obtained using the left endpoint numerical quadrature weights completely fails, in the sense that it does not even appear in Figure 3. By contrast, significantly better accuracy is observed when using the normalized left endpoint numerical quadrature weights.

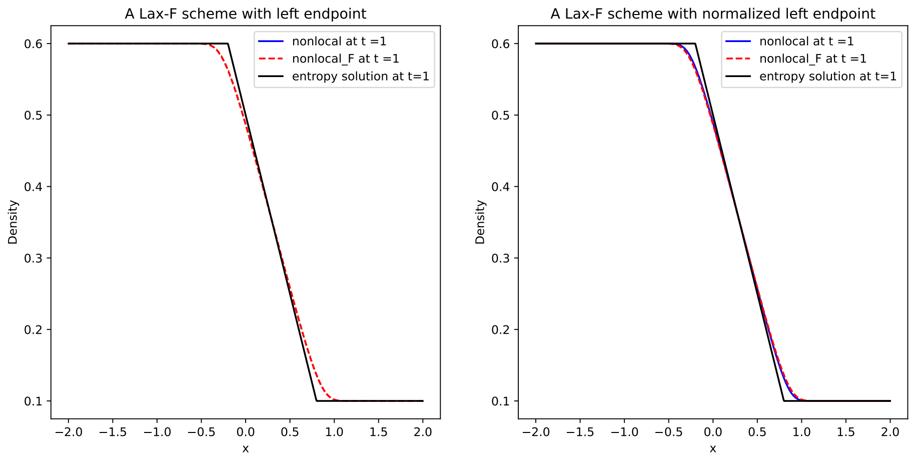

Figure 3: Numerical solutions and entropy solution at t = 1 corresponding to the left endpoint numerical quadrature weights (left) and the normalized left endpoint numerical quadrature weights (right).

Summary.
The numerical experiments above demonstrate that, as Δx and ϵ tend to zero simultaneously, while the finite volume scheme based on left endpoint quadrature weights becomes increasingly inaccurate and eventually fails to produce a meaningful numerical solution, the normalized left endpoint quadrature weights consistently maintain stability and improved alignment with the entropy solution. These results indicate that the choice of numerical quadrature weights plays a crucial role in ensuring consistency between the Lax-Friedrichs scheme and the local model. For the sake of consistency it's required that the numerical quadrature weights satisfy the normalized condition $\sum_{k=0}^{m-1} w_k = 1$.

4.2 Convergence Analysis

Based on the numerical experiments above, the Lax-Friedrichs scheme seems consistent and stable when using the normalized left endpoint quadrature weights.

Let us continue the numerical experiment to determine the convergence and order of convergence. Denote U^{C_j} and $U_e^{C_j}$ as the approximate solution and the exact solution at cell C_j, respectively. Let the error at C_j be $e^{C_j} = U_e^{C_j} - U^{C_j}$, and we wish to get it's size as small as possible.

We measure the size of the global error $e = \sum_{j=0}^{K-1} e^{C_j}$ using the discrete ℓ^1 norm

$$\|e\|_{\ell^1} = \Delta x \sum_{k=0}^{K-1} |e^{C_j}|$$

and say that the Lax-Friedrichs scheme converges to the entropy (exact) solution if the global error $\|e\|_{\ell^1} \to 0$ as $\Delta x \to 0$.

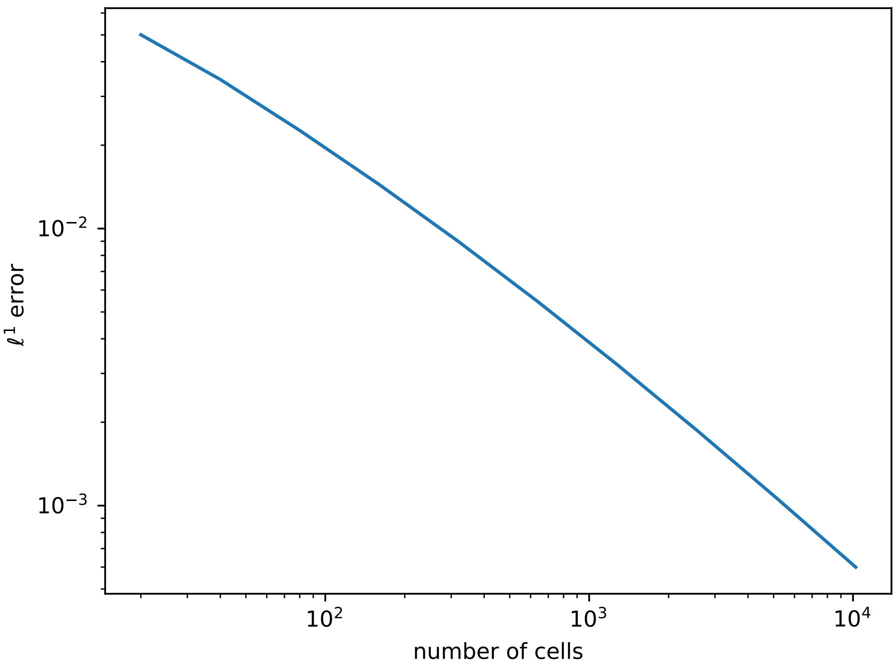

Figure 4: ℓ^1 error vs. number of cells

Figure 4 shows that the discrete ℓ^1-error tends to zero as the number of cells increases. In other words, the Lax–Friedrichs scheme converges to the entropy solution as $\Delta x \to 0$. A natural question is: how fast does it converge? To answer this, we study the order of convergence r.

We assume that the ℓ^1 error on a mesh of size Δx_i can be written as

$$E_i = C\,(\Delta x_i)^r,$$

for some constant $C > 0$. If we have the error at two consecutive refinement levels (E_{i-1}, E_i), then

$$\frac{E_{i-1}}{E_i} = \left(\frac{\Delta x_{i-1}}{\Delta x_i}\right)^r.$$

Taking logarithms, we obtain the standard expression for the order of convergence r:

$$r = \frac{\log(E_{i-1}) - \log(E_i)}{\log(\Delta x_{i-1}) - \log(\Delta x_i)}.$$

Table **??** shows the global error e and the observed order of convergence r for the Riemann problem with $u_L > u_R$. The corresponding entropy solution is

Number of cells	**Global error** e	**Order of convergence** r
20	0.050	0.536
40	0.035	0.611
80	0.023	0.648
160	0.014	0.685
320	0.009	0.722
640	0.005	0.755
1280	0.003	0.785
2560	0.002	0.810
5120	0.001	0.831

Table 1: The global error e and the order of convergence r for the Riemann problem with $u_L > u_R$.

a rarefaction wave (see (18)). We observe that the order of convergence remains below 1, indicating relatively slow convergence. The Lax-Friedrichs scheme is expected to be at most first order accurate because it is based on a piecewise constant approximation.

If instead we choose $u_L = 0.1$ and $u_R = 0.6$ for the Riemann problem (5), the entropy solution is

$$u(x,t) = \begin{cases} 0.1 & \text{if } x < 0.3t \\ 0.6 & \text{if } x > 0.3t \end{cases}$$

which contains a single shock. Table 2 reports the corresponding convergence results. In this case, the observed order of convergence reaches 1, indicating faster convergence.

Number of cells	**Global error** e	**Order of convergence** r
20	0.046	0.746
40	0.027	0.802
80	0.016	0.903
160	0.008	0.976
320	0.004	0.999
640	0.002	1.000
1280	0.001	1.000
2560	0.001	1.000
5120	0.000	1.000

Table 2: The global error e and the order of convergence r for the Riemann problem with $u_L < u_R$.

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.