Journal of Scientific Computing (2019) 81:1240–1265
https://doi.org/10.1007/s10915-019-00982-7

Numerical Analysis for Conservation Laws Using l_1 Minimization

Anne Gelb[1] · **X. Hou**[2] · **Q. Li**[2]

Received: 20 December 2018 / Revised: 17 May 2019 / Accepted: 28 May 2019 / Published online: 3 June 2019

Abstract
This paper develops and analyzes a new numerical scheme for solving hyperbolic conservation laws that combines the Lax Wendroff method with l_1 regularization. While prior investigations constructed similar algorithms, the method developed here adds a new critical conservation constraint. We demonstrate that the resulting method is equivalent to the well known lasso problem, guaranteeing both existence and uniqueness of the numerical solution. We further prove consistency, convergence, and conservation of our scheme, and also show that it is TVD and satisfies the weak entropy condition for conservation laws. Numerical solutions to Burgers' and Euler's equation validate our analytical results.

Keywords Conservation laws · l_1 regularization · Polynomial annihilation

Mathematics Subject Classification 65M08 · 65M15 · 65K10

1 Introduction

Hyperbolic conservation laws define a set of partial differential equations of hyperbolic type that describe the dynamics of some conserved quantities, such as mass and energy, in the physical domain. They are largely used in gas dynamics, acoustics, elastodynamics, optics

In memory of Saul Abarbanel, a great scholar, mentor, and friend.

The work of X. H. and Q. L. is supported in part by NSF-DMS 1619778, KI-Net RNMS-1107291 and NSF-TRIPODS 1740707. The work of A.G. is supported in part by National Science Foundation under the Grant NSF-DMS 1502640, NSF-DMS 1732434, and AFOSR FA9550-18-1-0316.

✉ Anne Gelb
annegelb@math.dartmouth.edu

X. Hou
xhou9@wisc.edu

Q. Li
qinli@math.wisc.edu

[1] Department of Mathematics, Dartmouth College, Hanover, NH, USA

[2] Department of Mathematics, UW-Madison, Madison, WI, USA

and geophysics. Even with smooth initial data, solutions to hyperbolic PDEs often contain discontinuities, or shocks, such as wave fronts that develop in finite time. To interpret the meaning of solutions at the discontinuities where the differential operators are invalid, the concept of *weak solutions* was developed. There are corresponding numerical challenges, as most numerical methods rely on some regularity assumptions of the solutions. A broadly investigated topic in numerical conservation laws is how to modify these methods so that they are suitable for problems with non-smooth solutions. While high order numerical solvers usually encode artificial dispersion that brings non-physical oscillations, low order numerical solvers, such as upwinding schemes, yield too much artificial viscosity that brings non-physical smoothing effects. *Slope limiters* were introduced to provide the right balance of artificial dispersion and artificial viscosity, in particular by accentuating the higher order methods more in smooth regions, in order to preserve a fast convergence rate, and less in regions with discontinuities, to prevent oscillations that lead to numerical instability, [5,10, 13–15,20–22]. A hybrid, spatially-adaptive, weighted, essentially non-oscillatory (WENO) scheme was developed in [5]. At each iteration, shocks are detected by comparing solution values in a neighborhood of 5 to 7 grid cells, and a weighted combination of several difference operators is then used to approximate the differential operator in the equation. Slope limiting methods and other spatially varying diffusion techniques are also used in global spectral methods. For example, in [22] the spectral viscosity (SV) method was adapted to include a step that locates the region containing the shock location. By doing so, less viscosity is enforced in the smooth regions of the solution. High order post-processing, which typically requires knowledge of each shock location, is required to recover spectral accuracy from the SV solution [7,19]. Finally, a scheme that combines non-uniform and adaptively re-defined spatial meshes with entropy conservative schemes to compute the shock was proposed in [3].

Regardless of the underlying type of numerical method, the key to its success for hyperbolic conservation laws lies in its ability to capture shocks. The classical solvers discussed above all resolve the issue by carefully monitoring the solution's behavior at every grid point. They are all highly nonlinear, and heavily rely on a very delicate local manipulation of the local numerical solutions, bringing obvious computational complexity, as well as high numerical cost. A natural question, then, is to look for a systematic way of "detecting" shocks.

The recent advances in l_1 minimization may provide an answer. Specifically, l_1 minimization, as a surrogate of l_0 minimization, promotes sparsity in solutions. For conservation law equations in particular, since singular points are indeed scattered sparsely on the domain, the application of l_1 regularization could be potentially useful for extracting such information. Several approaches in this vein were taken in [8,10–12]. In [11,12], an l_1 minimization technique was applied for computing steady-state conservation laws in one and two dimensions respectively. In particular, the finite volume approximation of the corresponding non-singularly perturbed problem was written as an overdetermined system and then solved by minimizing the l_1 norm of the system residual. However, the method did not include time integration, and it is not apparent how such techniques could be adapted to time dependent problems when l_1 solutions are not explicitly available. In [10], the solution was divided into two components: while the discontinuous component can be explicitly obtained, the smooth part can be computed with standard high order methods. In [18], the authors assumed that the solutions are sparse in the Fourier space and included the l_1 constraints for updating the dynamics. However, the validity of the assumption was not justified and singularities in the physical domain were not considered. Moreover, the algorithm consists of advancing the PDE forward in time and then projecting the updated solution onto a sparse subset. This requires additional transformations between spatial and coefficient domains at each iteration, thereby adding another layer of complexity.

The approach we take in this paper was initially proposed in [17], where a two-stage procedure was developed for solving numerical conservation laws. The first stage applies a standard high order numerical PDE solver. Such a solver, while obtaining a high order of accuracy, may lead to wild artificial oscillations. Hence in the second stage, an l_1 minimization problem is solved to eliminate the oscillations and enforce the sparsity of discontinuities. A related technique developed in [16] combined several time steps of Lax–Wendroff scheme followed by a Lax–Friedrichs step, which ostensibly keeps the desired features of both methods. No theoretical analysis was provided, however. In [17], the "discontinuities" are captured through a polynomial annihilation (PA) operator, which provides a systematic way to detect shock locations. The difference between the solution sought and the standard numerical solution is used as the fidelity term. The primary advantage of the method is that it does not require the delicate monitoring of local solution behavior, as the PA operator could automatically extract discontinuities. Depending on the order of PA operator, the final scheme can potentially achieve high order of accuracy. It was also numerically demonstrated in [17] that the algorithm has a less restrictive time step than that imposed by the usual CFL condition, resulting in a fewer number of time step iterations.

While numerical examples in [17] demonstrated the power of the method, no theoretical results were provided. The well-posedness of the minimization problem, as well as its convergence properties, were left open, similar to most other works that employ l_1 minimization. The main goal of the current paper to fill this gap—first to improve the algorithm by adding a mass conservation constraint to the minimization procedure, and further to study the theoretical aspects of solver. We will demonstrate that the problem is equivalent to a lasso problem, and the existence and uniqueness of a numerical solution is guaranteed. We furthermore present that the solution to the minimization problem converges to the true solution. Properties such as convergence to the weak solution and total-variation-diminishing (TVD) will also be shown. Finally, we will discuss how physical intuition can be used to optimally tune the algorithm parameters.

The rest of this paper is organized as follows: in Sect. 2 we introduce the two-stage algorithm incorporating the PA operator that approximates the jump function. The l_1 minimization problem will be reduced to a lasso problem, and we employ the LARS algorithm to solve it. Some modifications to the algorithm are also illustrated in the same section. In Sect. 3 we present all theoretical justification, including the existence and uniqueness of the minimization problem, and some properties the scheme satisfies, such as mass conservation, TVD and l_1 contraction. In Sect. 4, we present numerical results on four examples using our scheme together with modifications. Concluding remarks are provided in Sect. 5.

2 Proposed Algorithm

A model hyperbolic conservation law equation with a periodic boundary condition is given by

$$\begin{cases} \partial_t u + \partial_x f(u) = 0, \quad (t, x) \in \mathbb{R}^+ \times [a, b] \\ u(t = 0, x) = u_{\text{in}}(x), \quad u(a) = u(b) \end{cases} . \tag{1}$$

Here u is the conserved quantity and $f(u)$ is the flux. In the linear case, $f(u)$ linearly depends on u and one obtains the simple advection equation. But typically $f(u)$ is a nonlinear function of u, which could potentially introduce singularities and shock fronts.

The numerical method developed in this section is designed to approximate u in the smooth regions with high accuracy while also capturing the sparsely located jump discontinuities.

The idea is to separate these goals and tackle them in two stages. In the first stage a higher order numerical method is applied to approximate $u(x,t)$. Since this may trigger artificial oscillations at the discontinuous points, in the second stage an l_1 minimization is encoded. The l_1 minimization term is applied on the polynomial annihilation (PA) operator, [2], which closely approximates the jump function of the underlying solution. By minimizing this term we are able to reduce artificial oscillations and sharpen shocks without compromising the accuracy of the solution in the smooth regions.

2.1 Proposed Algorithm

We first introduce some unifying notation. Although not required for our algorithmic development, for ease of presentation we consider only uniform grids, with

$$x_j = a + j\Delta x\,, \quad j = 0, 1, \ldots, N, \quad \text{and} \quad \Delta x = \frac{b-a}{N},$$

in space, and time discretization denoted by Δt. The final time T for solving (1) is given by $T = M\Delta t$. We denote U_j^n as the numerical approximation to $u(t_n, x_j)$, and $\mathsf{U}^n = [U_1^n \ \ldots, U_N^n]$ as the vector solution at time t_n.

As described above, there are two stages in the method for updating numerical solution from U^n to U^{n+1}. In Stage 1, one applies the standard Lax–Wendroff (LxW) method. It is well known that artificial oscillations will occur near the shock locations, and eventually lead to numerical instability unless otherwise mitigated. Hence in Stage 2 an l_1 penalty term is added, yielding a numerical solution that is close to the LxW solution but with artificial oscillations eliminated. This method, which we will refer to as the l_1-Modified-Lax–Wendroff Algorithm, is detailed in Algorithm 1.

Algorithm 1: l_1 modified Lax–Wendroff scheme

Input: Initial value at each grid point $U_j^0 = u_{\text{in}}(x_j)$, $j = 1, 2, \ldots, N$

Output: Numerical solution at final time: U_j^M, $j = 1, 2, \ldots, N$

for $n = 0, \cdots, M-1$ **do**

Stage 1 updates the numerical solution according to classical higher order PDE solvers, such as the Lax-Wendroff (LxW) method:

$$U_j^{n+1/2} = U_j^n - \frac{\Delta t}{\Delta x}\Big[f(U_{j+1/2}^n) - f(U_{j-1/2}^n)\Big]\,, \tag{2}$$

where the fluxes are evaluated at:

$$U_{j+1/2}^n = \frac{1}{2}(U_j^n + U_{j+1}^n) - \frac{\Delta t}{2\Delta x}(f(U_{j+1}^n) - f(U_j^n))\,.$$

Stage 2 improves the numerical solution given by the LxW by encoding l_1 regularizer on the PA operator to eliminate artificial oscillations:

$$\begin{aligned} \mathsf{U}^{n+1}(\lambda) = \arg\min_{\mathsf{V}} \Big(&\frac{1}{2}\left\|\mathsf{U}^{n+1/2} - \mathsf{V}\right\|_2^2 + \lambda \left\|\mathcal{L}^m \mathsf{V}\right\|_1\Big) \\ &\text{s.t.} \quad \mathsf{e}^\top(\mathsf{V} - \mathsf{U}^{n+1/2}) = 0\,. \end{aligned} \tag{3}$$

end

In (3) of Algorithm 1, $\mathcal{L}^m$ is the PA operator so that

$$(\mathcal{L}^m \mathsf{U})_i \sim [u](x_i)\,,$$

where $[u]$ is the jump function defined as

$$[u]\,(y) := u(y^+) - u(y^-)\,. \tag{4}$$

Observe that (4) takes value 0 when the function is smooth but records the jump value at the discontinuities. Clearly then $\{u(t_n, x_j) : j = 0, \ldots, N\}$, for all $n = 0, \ldots, M$, is *sparse* in its jump function domain. Hence it is appropriate to use $(\mathcal{L}^m \mathsf{U}^{\mathsf{n}})_j$ to approximate $[u](t_n, x_j)$ in the l_1 regularization term.

Note that Algorithm 1 differs from the method introduced in [17] in one important way—namely *conservation* of (1) is added as a constraint:

$$\mathsf{e}^\top(\mathsf{U}^n - \mathsf{U}^{n+1/2}) = 0 \quad \text{with} \quad \mathsf{e} = \frac{1}{\sqrt{N}}[1, 1, \ldots, 1]^\top\,,$$

which imposes

$$\sum U_j^{n+1}(\lambda) = \sum U_j^{n+1/2}(\lambda)\,.$$

We emphasize the dependence in $\mathsf{U}^{n+1}(\lambda)$ since the final solution relies on the choice of λ. As will be demonstrated later, the mass conservation constraint is crucial. The conservation property, combined with the consistency of the algorithm, leads to the convergence to the weak solution.

2.2 PA Operator

A good approximation to the jump function is critical to the success of Algorithm 1. Solutions to hyperbolic conservation laws have sparse discontinuities, and it is imperative that the l_1 operator is applied to accurately capture this sparsity. The polynomial annihilation (PA) operator $\mathcal{L}^m$, introduced in [2], yields an m-th order approximation to the jump function of its corresponding piecewise smooth function from a (local) set of grid points. The general definition of the PA operator is given by

$$\mathcal{L}^m f(y) = \frac{1}{q^m(y)} \sum_{x_j \in S} c_j(y) f(x_j)\,, \tag{5}$$

where m is the order of approximation to the jump function $[f]\,(y)$, and $S = \{x_j, j = 1, \ldots, m+1\}$ is any given local set of $m+1$ grid points. The annihilation coefficients c_j are computed from

$$\sum_{x_j} c_j(y) p_l(x_j) = p_l^{(m)}(y), \quad j = 1, \ldots, m+1\,,$$

where $\{p_l : l = 0, \ldots, m\}$ is a basis for the space of polynomials of degree less than or equal to m. The normalization factor q^m is given by

$$q^m(y) = \sum_{x_j \in S^+} c_j(y)\,,$$

where $S^+ = \{x_j \in S : x_j \geq y\}$. Suppose f has m continuous derivatives in smooth regions, then the operator $\mathcal{L}^m$ approximates the jump function with m-th order accuracy, namely, it

captures the discontinuities at singular points and yields an m-th order accurate approximation to 0 in the smooth regions, as summarized in the following theorem:

Theorem 1 (Theorem 3.1 in [2]) *Let $m \in \mathbb{N}$ and $\mathcal{L}^m$ be defined as in (5) using a local set S_x with $m+1$ elements. Then we have*

$$\mathcal{L}^m f(x) = \begin{cases} [f](\xi) + O(h(x)) & \text{if } x_{j-1} \le \xi, x \le x_j \\ O(h^{\min(m,k)}(x)) & \text{if } f \in C^k(I_x) \text{ for } k > 0 \end{cases},$$

where $h(x) = \max\{|x_i - x_{i-1}| : x_{i-1}, x_i \in S_x\}$ and I_x is the smallest closed interval such that $S_x \subset I_x$.

Furthermore, when the grids in S are uniform and f has periodic boundaries, the PA operators have explicit form:

$$\mathcal{L}^1 = \begin{bmatrix} 1 & -1 & & & \\ & 1 & -1 & & \\ & & \ddots & \ddots & \\ & & & 1 & -1 \\ -1 & & & & 1 \end{bmatrix}, \quad \mathcal{L}^3 = \frac{1}{2}\begin{bmatrix} 3 & -3 & 1 & & & -1 \\ -1 & 3 & -3 & 1 & & \\ & \ddots & \ddots & \ddots & \ddots & \\ & & -1 & 3 & -3 & 1 \\ 1 & & & -1 & 3 & -3 \\ -3 & 1 & & & -1 & 3 \end{bmatrix}. \tag{6}$$

It was shown in [17] that using the PA operator instead of the standard TV regularization (which is equivalent to $m = 1$) better captures the variation in the smooth regions. This is especially true when problems are not well resolved. Using $m = 1$ in this case will cause a staircasing effect since the solution space is comprised of piecewise constant functions.

Remark 1 We note that the PA operators were originally designed to detect edges in images, where a similar l_1 regularization is applied to promote the sparsity of edges:

$$f = \arg\min_g \left(\left\| \mathcal{L}^m g \right\|_1 + \frac{\lambda}{2} \left\| \mathcal{F} g - \hat{f} \right\|_2^2 \right).$$

Here f is the reconstructed signal or image, $\hat{f}$ are the acquired data measurements, and $\mathcal{F}$ is a model that projects f to $\hat{f}$. Applications for when the acquired data are Fourier samples are thoroughly discussed in [1].

2.3 LARS Algorithm

The minimization problem, (3) in Stage 2 of Algorithm 1, contains mass conservation as a separate constraint. However, through change of variables and by the properties of the PA transform operator $\mathcal{L}^m$, it could be reduced to the unconstrained minimization problem given by

$$\arg\min_{\mathsf{b}} \left(\frac{1}{2} \|\gamma - \mathsf{X}\mathsf{b}\|_2^2 + \lambda \|\mathsf{b}\|_1 \right), \tag{7}$$

which has the form of the standard lasso (least absolute shrinkage and selection operator) problem. The translation from $(\mathcal{L}, \mathsf{U}, \mathsf{V})$ to $(\mathsf{X}, \gamma, \mathsf{b})$ will be presented in Theorem 2. In the new formulation, γ is the data to fit, b is the parameter whose linear transform is expected to

be close to γ, while its sparsity is promoted through the l_1 term, and λ is the regularization parameter.

Several algorithms are widely accepted as good candidates for solving lasso problems. For example, the alternating direction method of multipliers (ADMM) translates an unconstrained problem into a constrained one and applies an augmented Lagrangian scheme that uses partial updates for the dual variables. The *glmnet* algorithm was developed in [6] to compute the solution path for generalized linear models with convex penalties, including the lasso problem, by using the coordinate descent method for convex problems. In the solver SPGL1 [24], an iterative method is used to solve the lasso problem by applying the spectral gradient-projection method at each iteration [25].

Here we use the well-known LARS (least-angle regression) algorithm, which we choose for two reasons. First, LARS provides not only the solution to any prior chosen λ, but also a solution path that characterizes its dependence on λ. As will be discussed in Sect. 3, the choice of λ is critical, with only a small range of suitable λ yielding both convergence and total-variation-diminishing (TVD). Although this range of λ is not known a-priori, the LARS algorithm reveals the solution's dependence on λ which in turn provides guidance on its tuning. Second, the method gives the "exact" solution, meaning that it obtains up to machine precision accuracy. Numerical solutions to hyperbolic conservation laws are typically very sensitive to errors. Indeed, small perturbations in the solution profile, especially close to shock locations, can significantly change the shock speed and/or location, inevitably leading to large errors. Most other algorithms used for solving (7) follow some variation of gradient-descent and stop at certain preset thresholds. In these cases, the accuracy depends heavily on the conditioning of the problem (determined by X). On the other hand, as noted above, LARS provides exact numerical solutions (up to machine precision). We stress that we are not proposing that LARS generally be used for solving (3), as it is computationally inefficient. Rather we are using it to demonstrate the suitability of our approach. ADMM was used in [17], and we anticipate other algorithms will also provide similar results, although theoretical justification will not be readily available.

The general form of lasso problem is the following: given the outcome vector $\gamma \in \mathbb{R}^n$, and the matrix $\mathsf{X} \in \mathbb{R}^{n\times p}$ of predictor variables, one looks for $\mathsf{b} \in \mathbb{R}^p$ so that Xb fits γ in l_2 norm with b expected to have as few non-trivial entries as possible. More specifically, the minimization formulation is

$$\hat{\mathsf{b}} = \arg\min_b \frac{1}{2}\|\gamma - \mathsf{Xb}\|_2^2 + \lambda \|\mathsf{b}\|_1 \,. \tag{8}$$

Clearly (8) reduces to the simple least square problem when $\lambda = 0$. Here we only consider $\lambda > 0$ to exploit the sparsity in b. Since (8) is convex, it is optimized when its gradient is 0. While the gradient of the l_2 term is simply determined as $\mathsf{X}^\top(\mathsf{Xb} - \gamma)$, for the l_1 term, which is not differentiable, we look at the subgradient η with components

$$\eta_i \in \begin{cases} \{\text{sign}(\mathsf{b}_i\} & \text{if } \mathsf{b}_i \neq 0 \\ [-1, 1] & \text{if } \mathsf{b}_i = 0 \end{cases}.$$

Thus the optimal solution $\hat{\mathsf{b}}$ satisfies

$$\mathsf{X}^\top(\gamma - \mathsf{X}\hat{\mathsf{b}}) = \lambda\eta \,.$$

Let us now denote $\mathcal{E}$ as the equi-correlation set by

$$\mathcal{E} = \left\{ i \in \{1, \ldots, p\} : \left| \mathsf{X}_i^\top(\gamma - \mathsf{X}\hat{\mathsf{b}}) \right| = \lambda \right\} .$$

That is, $\mathcal{E}$ is the set of indices having equal (and maximal) absolute correlation with the residual $\gamma - \mathsf{X}\hat{\mathsf{b}}$. We further denote the equi-correlation signs by

$$\mathsf{s} = \text{sign}(\mathsf{X}_{\mathcal{E}}^{\top}(\gamma - \mathsf{X}\hat{\mathsf{b}}))\,,$$

where $\mathsf{X}_{\mathcal{E}}$ comprises the columns of X with indices in $\mathcal{E}$. Applying the KKT condition, in [23] it was proven that any lasso solution $\hat{\mathsf{b}}$ has the form

$$\hat{\mathsf{b}}_{\mathcal{E}^c} = 0\,, \quad \text{and} \quad \hat{\mathsf{b}}_{\mathcal{E}} = (\mathsf{X}_{\mathcal{E}})^{\dagger}(\gamma - (\mathsf{X}_{\mathcal{E}}^{\top})^{\dagger}\lambda\mathsf{s}) + \mathsf{c}\,,$$

where $\mathcal{E}^c$ denotes the complement of $\mathcal{E}$ and $\mathsf{c} \in \text{null}(\mathsf{X}_{\mathcal{E}})$. There are two immediate takeaways:

1: If $\mathcal{E}$ can be determined a-priori, the lasso solution can be viewed as a linear function of λ. In particular, the solution's dependence on λ is explicit.
2: If $\mathsf{X}_{\mathcal{E}}$ is not full rank, $\text{null}(\mathsf{X}_{\mathcal{E}})$ is not empty, and there are infinitely many choices for c. This leads to the non-uniqueness of the solution.

It is apparent that finding the equi-correlation set $\mathcal{E}$ is the key to solving the problem, and finding the optimizer space is equivalent to finding $\mathcal{E}$. Determining $\mathcal{E}$, however, is not straightforward, except in the case where $\lambda = \infty$, which yields $\mathcal{E} = \emptyset$. More precisely, at $\lambda = \infty$, the sparsity term dominates, and the optimizer has the trivial solution and all entries are zeros.

LARS fully makes use of the fact that $\mathcal{E}$ determines the solution space (including that $\mathcal{E} = \emptyset$ at $\lambda = \infty$) by tracing $\mathcal{E}$'s dependence on λ and gradually adding indices into it. The solution path is provided in the sense that a solution is determined for each λ as λ decreases from ∞ [or a big enough value so that $\mathsf{b}(\lambda)$ is trivial] to 0. By construction, as λ decreases, the l_1 penalty term increasingly makes room for the mismatch (fidelity) term to become more significant, allowing more non-zero entries in the (sparse) vector solution. Indices are correspondingly added to or deleted from $\mathcal{E}$. To determine at which λ indices are added to or deleted from $\mathcal{E}$, we use the explicit solution in (9). As suggested by the formula, each entry in the solution path is a piecewise linear function, and the "joining" and "crossing" times can be precomputed simply by setting $\hat{\mathsf{b}}_i = 0$ for $i \in \mathcal{E}$. Details for computing the joining and crossing times can be found in [23]. Algorithm 2 summarizes the method.

As λ decreases, λ_k records all the times where adding (deleting) is taking place. As is evident in (9), the entries of the lasso solution $\hat{\mathsf{b}}$ are linear functions of λ_k. This allows us to explicitly compute the next joining time and crossing time as in (10) and (11), where t_i^{join} is the joining time of the ith entry and t_j^{cross} is the crossing time of the jth entry. We update λ_{k+1} as the closest time when a variable joins or leaves the equi-correlation set $\mathcal{E}$.

The LARS algorithm enjoys a number of good properties that will be heavily used in our later applications. We summarize them below.

Proposition 1 *Denote $\hat{b}$ the LARS lasso solution to (8), then $\hat{b}$, $X\hat{b}$, and $\left\|\hat{b}\right\|_1$ all continuously depend on λ.*

Proof By construction of the LARS algorithm and LARS lasso solution in (9), we can easily determine that each entry of the solution $\hat{\mathsf{b}}$ is a continuous piecewise linear function of λ (piecewise linear or piecewise 0). Furthermore, both matrix multiplication and l_1 norms are continuous functions. Hence $\hat{\mathsf{b}}$, $\mathsf{X}\hat{\mathsf{b}}$, and $\left\|\hat{\mathsf{b}}\right\|_1$ are all continuous functions of λ. □

There is also a loose upper bound of the computational cost which depends on the size of matrix X [23]:

Algorithm 2: LARS algorithm

Input: γ, X
Output: Solution path of the lasso problem

1 Start with $k = 0$, $\lambda_0 = \infty$, $\mathcal{E} = \varnothing$, and $\mathsf{s} = \varnothing$;
2 **while** $\lambda_k > 0$ **do**
3 Compute the LARS lasso solution at λ_k by least squares:

$$\hat{\mathsf{b}}_{\mathcal{E}^c} = 0\,, \quad \text{and} \quad \hat{\mathsf{b}}_{\mathcal{E}}(\lambda_k) = (\mathsf{X}_{\mathcal{E}})^{\dagger}(\gamma - (\mathsf{X}_{\mathcal{E}}^{\top})^{\dagger}\lambda_k \mathsf{s})\,, \tag{9}$$

and continue in a linear direction from the solution for $\lambda \leq \lambda_k$, that is, compute the linear solution path between λ_k and λ_{k+1};
4 Compute the next joining time $\lambda_{k+1}^{\text{join}}$, i.e. when a variable outside the equi-correlation set achieves the maximal absolute inner product with the residual:

$$\lambda_{k+1}^{\text{join}} = \max_{i \notin \mathcal{E}} t_i^{\text{join}}\,, \quad \text{where} \quad t_i^{\text{join}} = \frac{\mathsf{X}_i^{\top}(\mathsf{I} - \mathsf{X}_{\mathcal{E}}(\mathsf{X}_{\mathcal{E}})^{\dagger})\gamma}{\pm 1 - \mathsf{X}_i^{\top}(\mathsf{X}_{\mathcal{E}}^{\top})^{\dagger}\mathsf{s}}. \tag{10}$$

Note that exactly one of $+1$ and -1 will lead to $t_i^{\text{join}} \in [0, \lambda_k]$;
5 Compute the next crossing time $\lambda_{k+1}^{\text{cross}}$, when the coefficient path of an equi-correlation variable crosses through zero:

$$\lambda_{k+1}^{\text{cross}} = \max_{i \in \mathcal{E}} t_i^{\text{cross}}\,, \quad \text{where} \quad t_i^{\text{cross}} = \frac{\left[(\mathsf{X}_{\mathcal{E}})^{\dagger}\gamma\right]_i}{\left[(\mathsf{X}_{\mathcal{E}}^{\top}\mathsf{X}_{\mathcal{E}})^{\dagger}\mathsf{s}\right]_i} \cdot 1\left\{\frac{\left[(\mathsf{X}_{\mathcal{E}})^{\dagger}\gamma\right]_i}{\left[(\mathsf{X}_{\mathcal{E}}^{\top}\mathsf{X}_{\mathcal{E}})^{\dagger}\mathsf{s}\right]_i} \leq \lambda_k\right\}; \tag{11}$$

6 Set $\lambda_{k+1} = \max\left\{\lambda_{k+1}^{\text{join}}, \lambda_{k+1}^{\text{cross}}\right\}$. If $\lambda_{k+1}^{\text{join}} > \lambda_{k+1}^{\text{cross}}$, then add the joining variable to $\mathcal{E}$ and its sign to s; otherwise, remove the crossing variable from $\mathcal{E}$ and its sign from s;
7 Update $k = k + 1$;
8 **end**

Proposition 2 *For any γ, X, the LARS algorithm for the lasso path performs at most $\sum_{k=1}^{p}\binom{p}{k}2^k = 3^p$ iterations before termination.*

The LARS algorithm is used as a black box to treat (8). Many of the useful properties of LARS algorithm described in Propositions 1 and 2 are maintained when applied to our numerical conservation law problem. This will be discussed in Sect. 3.2.

2.4 Modification

While Algorithm 1 provides the general blue print for solving a conservation law exploiting the sparsity of the shocks and ensuring conservation, there are several issues that must be addressed to ensure its efficient and accurate implementation, namely how the regularization parameter λ is chosen, how to ensure computational efficiency, and what properties are needed for the ℓ_1 operator.

2.4.1 Choice of λ

Choosing λ is critical for the minimization problem in the second stage of the algorithm, as it balances the weight on the fidelity function in l_2 norm and the sparsity of the discontinuities. Large λ enforces strong sparsity but drives the solution away from the LxW result, while

small λ is not enough to eliminate oscillations. As λ goes from ∞ to 0, the oscillations gradually becomes stronger, until the scheme eventually recovers the LxW solution at $\lambda = 0$.

One way to choose λ is to enforce the total variation diminishing (TVD) property. It is widely accepted that a meaningful numerical solution should preserve the properties satisfied by the true solution. For instance, since analytic solutions to hyperbolic conservation laws are total-variation diminishing in time, the corresponding numerical solution should be as well. The standard LxW method is not TVD, however, since the artificial oscillations introduced by LxW increase the total-variation of the solution. Therefore, in Stage 2, when the lasso problem is applied to eliminate the oscillations, a natural criterion is to set λ bigger than λ_{TV}, that is, the minimum λ to ensure a TVD solution. We therefore modify the LARS algorithm to trace the TV norm of the numerical solution and stop at the biggest λ where the TV norm is smaller than the previous time step. This will be further explained in Sect. 3.

2.4.2 Reducing Number of Minimization Problems

Many efficient methods exist for solving the lasso problem, see e.g. [6,23–25]. Nevertheless solving the lasso problem as part of the numerical PDE solver adds another layer of computation. To avoid excessive numerical costs and complexity, it is natural to ask if the minimization problem truly needs to be solved at each time step. One possible way of reducing numerical cost is to implement the minimization stage every few time steps, or even only as a post-processing final time step if stability can be maintained.

2.4.3 Choosing $\mathcal{L}^m$

In this investigation we use the PA operator, which can be regarded as high order total variation (HOTV), to approximate the jump function. The PA order m can significantly affect the results. Observe that $\mathcal{L}^1$, when applied to a function, is numerically equivalent to taking the TV norm of the discrete version of that function. By contrast, $\mathcal{L}^3$ provides a higher (third) order approximation to the jump function.[1] Neither has full rank, however. Due to the periodic boundary condition, the rank of the matrix is one smaller than the size, which leads to some difficulties when translating to the lasso problem. This is easily overcome by simply removing the last row, meaning that the TV norm is taken only in the interior. In this case the matrix becomes full rank with the following form:

$$\mathcal{L}^1 = \begin{bmatrix} 1 & -1 & & & \\ & 1 & -1 & & \\ & & \ddots & \ddots & \\ & & & 1 & -1 \end{bmatrix}. \tag{12}$$

Without any theoretical justification of the algorithm's convergence, it is difficult to decide in advance how these three issues—parameter tuning, cost of implementation, and choosing the regularization operator—should be addressed, nor understand what the trade offs may be. Hence in Sect. 3 we will establish some theoretical results and in Sect. 4 we extensively test our numerical algorithm to determine what these trade offs are. In so doing we provide the intuition needed to associate certain types of solution behaviors with the different choices we make. This will allow practitioners to develop schemes based on Algorithm 1 for their particular application.

[1] For simplicity we consider only odd orders which yield symmetric stencil for uniformly distributed grid points.

3 Theoretical Results

As noted above, the l_1-Modified-Lax–Wendroff scheme provided in Algorithm 1 is composed of two stages: the standard LxW method is used in Stage 1, and the mass-conservation-constraint minimization problem with an l_1 penalty term is solved in Stage 2.

To understand why our algorithm works, we separate the discussion of the two stages. Convergence for the LxW method has been established for conservation laws, see e.g. [9,14,15]. Hence in this paper we mainly focus on analyzing Stage 2. To this end in Sect. 3.1 we first reduce the problem to a standard lasso format, and utilize the well-posedness result for lasso to show our minimization problem has a unique solution for every pre-chosen regularization coefficient λ. We demonstrate that this unique solution, in the zero limit of λ, converges to the LxW solution. The result is summarized in Theorem 4.

Numerical methods for hyperbolic conservation laws should satisfy properties such as conservation and total-variation-diminishing (TVD). We discuss these properties in Sect. 3.2. As will be established in the corresponding proof, the PA operators have beneficial properties that are key to the success of our scheme.

3.1 Well-Posedness

A natural question to ask in Stage 2 is whether or not the minimization formulation (3) has a unique solution. We investigate this question by first writing (3) in a lasso formulation, for which well-posedness has been frequently analyzed. To do so, we first observe that the constraint-minimization problem in Stage 2 can be summarized as

$$\begin{aligned} \hat{\mathsf{V}}(\lambda) = \arg\min_{\mathsf{V}} \left(\frac{1}{2} \|\mathsf{U} - \mathsf{V}\|_2^2 + \lambda \, \|\mathcal{L}\mathsf{V}\|_1 \right), \\ \text{s.t.} \quad \mathsf{e}^\top (\mathsf{V} - \mathsf{U}) = 0 \end{aligned} \tag{13}$$

where we have omitted the super-indices of U and $\mathcal{L}$ for simplicity. Although (13) is not a standard lasso formulation, below we demonstrate that with some manipulation it is possible to reduce it to one, and then apply well-posedness results for lasso problems. To do so, we first must write (13) as an unconstrained problem, which is accomplished in Lemma 1.

Lemma 1 *Let $\hat{\alpha}$ be the minimizer of the following unconstrained optimization problem*

$$\hat{\alpha} = \arg\min_{\alpha} \left(\frac{1}{2} \|\gamma - \alpha\|_2^2 + \lambda \, \|\mathcal{L}E\alpha\|_1 \right), \tag{14}$$

where E is any $n \times (n-1)$ matrix with orthonormal columns that satisfies $e^\top E = 0$, and $\gamma = E^\top U$. Assuming U is known, the optimization problem (14) is equivalent to (13) in the following sense:

$$\hat{V} = ee^\top U + E\hat{\alpha}\,, \quad \hat{\alpha} = E^\top (\hat{V} - ee^\top U)\,. \tag{15}$$

Proof Suppose $\hat{\mathsf{V}}$ is the solution to the minimization problem (13). Since E has orthonormal columns with $\mathsf{e}^T\mathsf{E} = 0$, $\mathbb{R}^n = \text{span}\{\mathsf{e}, \mathsf{E}\}$, and there exists $a \in \mathbb{R}$ and $\hat{\alpha} \in \mathbb{R}^{(n-1)\times 1}$ such that

$$\hat{\mathsf{V}} = a\mathsf{e} + \mathsf{E}\hat{\alpha}\,.$$

Taking the inner product with e and utilizing the fact that $\mathsf{e}^\top\mathsf{U} = \mathsf{e}^\top\hat{\mathsf{V}}$, we see that $a = \mathsf{e}^\top\mathsf{U}$ and $\hat{\alpha} = \mathsf{E}^\top(\hat{\mathsf{V}} - \mathsf{e}\mathsf{e}^\top\mathsf{U})$. Thus, for $\gamma = \mathsf{E}^\top\mathsf{U}$, we have

$$\|\hat{\mathsf{V}} - \mathsf{U}\|_2^2 = \|\mathsf{E}\hat{\alpha} - \mathsf{E}\gamma\|_2^2 = \|\hat{\alpha} - \gamma\|_2^2\,, \tag{16}$$

where we have used the orthonormality of E to obtain the result.

Observe now that $\mathcal{L} \in \mathbb{R}^{n\times n}$ is a rank $n-1$ finite difference matrix. The null space and the space perpendicular to the range are explicitly determined as

$$\mathcal{L}\mathsf{e} = \mathcal{L}^\top \mathsf{e} = 0\,, \tag{17}$$

which means $\mathcal{L}\hat{\mathsf{V}} = \mathcal{L}\mathsf{E}\hat{\alpha}$. For any α, we can define $\mathsf{V} = a\mathsf{e} + \mathsf{E}\alpha$. Similar to the analysis leading to (16), we have $\|\mathsf{V}-\mathsf{U}\|_2^2 = \|\alpha-\gamma\|_2^2$ and $\mathcal{L}\mathsf{V} = \mathcal{L}\mathsf{E}\alpha$. Since $\hat{\mathsf{V}}$ is the minimizer of (13), we immediately have

$$\begin{aligned}\frac{1}{2}\left\|\gamma-\hat{\alpha}\right\|_2^2 + \lambda\left\|\mathcal{L}\mathsf{E}\hat{\alpha}\right\|_1 &= \frac{1}{2}\left\|\mathsf{U}-\hat{\mathsf{V}}\right\|_2^2 + \lambda\left\|\mathcal{L}\hat{\mathsf{V}}\right\|_1 \\ &\le \frac{1}{2}\left\|\mathsf{U}-\mathsf{V}\right\|_2^2 + \lambda\left\|\mathcal{L}\mathsf{V}\right\|_1 \\ &= \frac{1}{2}\left\|\gamma-\alpha\right\|_2^2 + \lambda\left\|\mathcal{L}\mathsf{E}\alpha\right\|_1\,,\end{aligned}$$

which means $\hat{\alpha}$ is a solution to the unconstrained problem (14).

To prove the other direction, we start with the solution $\hat{\alpha}$ to the unconstrained problem (14) and let $\hat{\mathsf{V}} = \mathsf{e}\mathsf{e}^\top\mathsf{U} + \mathsf{E}\hat{\alpha}$. Since

$$\mathsf{I} = \mathsf{e}\mathsf{e}^\top + \mathsf{E}\mathsf{E}^\top\,,$$

then

$$\mathsf{E}\gamma = \mathsf{E}\mathsf{E}^\top\mathsf{U} = (\mathsf{I}-\mathsf{e}\mathsf{e}^\top)\mathsf{U}\,.$$

Utilizing this fact, two terms in the minimization problem can be rewritten:

$$\left\|\hat{\alpha}-\gamma\right\|_2^2 = \left\|\mathsf{E}\hat{\alpha}-\mathsf{E}\gamma\right\|_2^2 = \left\|(\hat{\mathsf{V}}-\mathsf{e}\mathsf{e}^\top\mathsf{U})-(\mathsf{I}-\mathsf{e}\mathsf{e}^\top)\mathsf{U}\right\|_2^2 = \left\|\hat{\mathsf{V}}-\mathsf{U}\right\|_2^2\,,$$

and

$$\left\|\mathcal{L}\mathsf{E}\hat{\alpha}\right\|_1 = \left\|\mathcal{L}(\hat{\mathsf{V}}-\mathsf{e}\mathsf{e}^\top\mathsf{U})\right\|_1 = \left\|\mathcal{L}\hat{\mathsf{V}}\right\|_1\,.$$

For any V, there exists $\alpha = \mathsf{E}^\top(\mathsf{V}-\mathsf{e}\mathsf{e}^\top\mathsf{U})$ with similar equivalence of l_2 norms and l_1 norms. Therefore, since $\hat{\alpha}$ is a minimizer to (14),

$$\begin{aligned}\frac{1}{2}\left\|\mathsf{U}-\hat{\mathsf{V}}\right\|_2^2 + \lambda\left\|\mathcal{L}\hat{\mathsf{V}}\right\|_1 &= \frac{1}{2}\left\|\gamma-\hat{\alpha}\right\|_2^2 + \lambda\left\|\mathcal{L}\mathsf{E}\hat{\alpha}\right\|_1 \\ &\le \frac{1}{2}\left\|\gamma-\alpha\right\|_2^2 + \lambda\left\|\mathcal{L}\mathsf{E}\alpha\right\|_1 \\ &= \frac{1}{2}\left\|\mathsf{U}-\mathsf{V}\right\|_2^2 + \lambda\left\|\mathcal{L}\mathsf{V}\right\|_1\,.\end{aligned}$$

To complete the proof, we derive the constraint

$$\mathsf{e}^\top(\hat{\mathsf{V}}-\mathsf{U}) = \mathsf{e}^\top(\mathsf{e}\mathsf{e}^\top\mathsf{U} + \mathsf{E}\hat{\alpha} - \mathsf{e}\mathsf{e}^\top\mathsf{U} - \mathsf{E}\gamma) = \mathsf{e}^\top\mathsf{E}(\hat{\alpha}-\gamma) = 0\,.$$

□

To further reduce it to the formulation of the standard lasso problem, the matrix $\mathcal{L}\mathsf{E}$ must be moved to the fidelity term. However, since $\mathcal{L}\mathsf{E}$ is not guaranteed to be of full rank, the inverse may not exist, depending on which form of $\mathcal{L}$ is taken. To address this issue, we will use the pseudoinverse of $\mathcal{L}\mathsf{E}$, as described in the following theorem.

Theorem 2 *Let E be any $n \times (n-1)$ orthonormal matrix satisfying $e^\top E = 0$, $\gamma = E^\top U$, and define X as the $(n-1) \times n$ matrix $X = (\mathcal{L}E)^\dagger$, the pseudoinverse of $\mathcal{L}E$. Then the optimization problem (13) is equivalent to the lasso problem,*

$$\hat{b} = \arg\min_b \left(\frac{1}{2} \|\gamma - Xb\|_2^2 + \lambda \|b\|_1 \right), \tag{18}$$

where equivalence means that the optimizers are connected through

$$\hat{V} = ee^\top U + EX\hat{b}$$
$$\hat{b} = \mathcal{L}EE^\top (\hat{V} - ee^\top U)$$

while the corresponding l_2 norms and l_1 norms are equivalent with

$$\|U - \hat{V}\|_2 = \|\gamma - X\hat{b}\|_2 ,$$
$$\|\mathcal{L}\hat{V}\|_1 = \|\hat{b}\|_1 .$$

Proof From Lemma 1 we know that (13) is equivalent to (14). Hence here it suffices to show that (18) is equivalent to (14) in the sense that $\hat{\mathsf{b}} = \mathcal{L}\mathsf{E}\hat{\alpha}$ and $\hat{\alpha} = \mathsf{X}\hat{\mathsf{b}}$.

We first notice that the map between b in (18) and α in (14) is one-to-one. For any given α, define $b = \mathcal{L}\mathsf{E}\alpha$, then

$$\mathcal{L}\mathsf{E}\hat{\alpha} = \mathcal{L}\mathsf{E}(\mathcal{L}\mathsf{E})^\dagger \mathcal{L}\mathsf{E}\hat{\alpha} = \mathcal{L}\mathsf{E}(\mathcal{L}\mathsf{E})^\dagger \hat{\mathsf{b}} = \mathcal{L}\mathsf{E}\mathsf{X}\hat{\mathsf{b}} ,$$

meaning that $\alpha - \mathsf{X}\mathsf{b}$ belongs to the nullspace of $\mathcal{L}\mathsf{E}$. From (17) we have $\mathrm{Null}(\mathcal{L}) = \mathrm{span}\{\mathsf{e}\}$, and since $\mathsf{e}^\top \mathsf{E} = 0$, we can conclude that $\mathrm{Null}(\mathcal{L}\mathsf{E}) = \{0\}$ and thus $\alpha = \mathsf{X}\mathsf{b}$. Similarly, for any given b, define $\alpha = \mathsf{X}\mathsf{b}$, then

$$(\mathcal{L}\mathsf{E})\mathsf{X}(\mathcal{L}\mathsf{E})\hat{\alpha} = \mathcal{L}\mathsf{E}\hat{\alpha} = (\mathcal{L}\mathsf{E})\mathsf{X}\hat{\mathsf{b}}.$$

meaning $\alpha - \mathsf{X}\mathsf{b}$ is in the nullspace of $\mathcal{L}\mathsf{E}$ and thus $\mathsf{b} = \mathcal{L}\mathsf{E}\alpha$.

The equivalence between α and b then quickly yields

$$\|\mathsf{b}\|_1 = \|\mathcal{L}\mathsf{E}\alpha\|_1 , \quad \text{and} \quad \|\gamma - \mathsf{X}\mathsf{b}\|_2 = \|\gamma - \alpha\|_2 ,$$

naturally leading to the conclusion. □

Note that the last row of operator $\mathcal{L}$ defined in (12), which contains the periodic boundary conditions, is deleted. Thus the $(n-1)\times(n-1)$ matrix $\mathcal{L}\mathsf{E}$ is full rank, since E is perpendicular to e. Hence the pseudoinverse of $\mathcal{L}\mathsf{E}$ is the true inverse and is given by

$$\mathsf{X} = (\mathcal{L}\mathsf{E})^\dagger = (\mathcal{L}\mathsf{E})^{-1} . \tag{19}$$

By successfully reducing the optimization problem in Algorithm 1 to a standard lasso problem (13), we are now able to apply the analytical results for the lasso problem directly to our technique. Two results pertaining to the existence and uniqueness of the lasso problem in a general setting will be particularly useful:

Theorem 3 (Well-posedness of the lasso problem [23]) *For any γ, X, and fixed $\lambda \geq 0$, the lasso problem*

$$\hat{b} \in \arg\min_b \left(\frac{1}{2} \|\gamma - Xb\|_2^2 + \lambda \|b\|_1 \right)$$

has the following properties:

1. *There is either a unique solution or an (uncountably) infinite number of solutions.*
2. *Every solution* $\hat{b}$ *gives the same fitted value* $X\hat{b}$.
3. *If* $\lambda > 0$, *then every solution has the same* l_1 *norm* $\left\|\hat{b}\right\|_1$.

Proposition 3 (Convergence of LARS [23]) *For any* γ, X, *the numerical solution to Problem* (8) *by LARS converges to the least square solution that has the minimum* l_1 *norm as* $\lambda \to 0^+$, *that is,*

$$\lim_{\lambda \to 0^+} \hat{b}^{LARS}(\lambda) = \hat{b}^{LS,l_1},$$

where $\hat{b}^{LS,l_1} \in \arg\min_b \|\gamma - Xb\|_2^2$ *is a least square solution that achieves the minimum* l_1 *norm.*

Theorem 3 shows the well-posedness of standard lasso, while Proposition 3 shows that the LARS solution, in the zero limit of λ, recovers the least square solution, and furthermore, when the least square problem does not have unique solutions, LARS picks the one that has the minimum l_1 norm. Combining the Lemma 1, Theorems 2, 3, and Proposition 3 we are ready to show Theorem 4.

Theorem 4 *The optimization problem* (13) *has a unique solution with* $\mathcal{L}$ *being a PA operator. Moreover, the optimizer* $\hat{V}(\lambda)$ *converges to* U *as* $\lambda \to 0^+$.

Proof According to Theorem 2, one needs the uniqueness of $\hat{\mathsf{V}}$ which can be computed as

$$\hat{\mathsf{V}} = \mathsf{e}\mathsf{e}^\top \mathsf{U} + \mathsf{E}\mathsf{X}\hat{\mathsf{b}},$$

where $\hat{\mathsf{b}}$ is the solution the lasso problem (18). By the second property in Theorem 3, although $\hat{\mathsf{b}}$ is not unique, the linear combination $\mathsf{X}\hat{\mathsf{b}}$, however, is. This gives the unique evaluation of $\hat{\mathsf{V}}$.

To show convergence, we can simply apply Proposition 3 on the equivalence of (13), as shown in Theorem 2:

$$\lim_{\lambda \to 0^+} \hat{\mathsf{b}}^{\mathrm{LARS}}(\lambda) = \hat{\mathsf{b}}^{\mathrm{LS},l_1},$$

where $\hat{\mathsf{b}}^{\mathrm{LS},l_1} \in \arg\min_{\mathsf{b}} \|\gamma - \mathsf{X}\mathsf{b}\|_2^2$ is the least square solution that contains the minimum l_1 norm. Equivalently, $\hat{\mathsf{b}}^{\mathrm{LS},l_1}$ is the solution to the minimization problem (18) when $\lambda = 0$. By Theorem 2, $\hat{\mathsf{b}}^{\mathrm{LS},l_1}$ can be computed using the constrained problem (13), which has the minimizer $\hat{\mathsf{V}}(\lambda) = \mathsf{U}$ when $\lambda = 0$ given by

$$\hat{\mathsf{b}}^{\mathrm{LS},l_1} = \mathcal{L}\mathsf{E}\mathsf{E}^\top(\mathsf{U} - \mathsf{e}\mathsf{e}^\top \mathsf{U}).$$

From Theorem 2 we have

$$\hat{\mathsf{V}}(\lambda) = \mathsf{e}\mathsf{e}^\top \mathsf{U} + \mathsf{E}\mathsf{X}\hat{\mathsf{b}}^{\mathrm{LARS}}(\lambda). \tag{20}$$

Finally, by the equivalence conditions in Lemma 1 and Theorem 2, we can conclude that

$$\lim_{\lambda \to 0^+} \hat{\mathsf{V}}(\lambda) = \mathsf{U}.$$

□

3.2 Scheme Properties

Upon showing the uniqueness of the minimization problem in Stage 2, we can continue to discuss the properties of the l_1-Modified-Lax–Wendroff scheme. Generally speaking, for a numerical conservation law method to be reliable, the following numerical properties must hold:

- **Property 1** The numerical solution is conservative.
- **Property 2** The numerical solution is consistent.
- **Property 3** The numerical solution converges to a weak solution as $h \to 0$.
- **Property 4** The numerical solution satisfies entropy condition.
- **Property 5** The numerical solution is total variation diminishing (TVD).

It is relatively straightforward to demonstrate **Property 1**. Indeed, in Stage 1, a finite volume type method was used and U_j^n is updated through the flux term, and in Stage 2, the mass conservation is encoded in the optimization constraint. Since mass is conserved in both stages, the whole scheme is automatically conservative. To address **Property 2**, we know that the LxW method is second order and is therefore consistent. New errors may be introduced into the numerical solution in Stage 2, however. In what follows we show that the LARS algorithm keeps the numerical error small enough to maintain consistency. Since the method is conservative and consistent it automatically converges to a weak solutions, yielding **Property 3**. These properties are proved in Theorem 5, while the TVD properties and entropy condition are demonstrated in Propositions 4 and 5 respectively.

Theorem 5 (Consistency, conservative and convergence) *The numerical solution of the l_1 modified Lax–Wendroff scheme converges to a weak solution of the hyperbolic conservation law* (1)*. In particular:*

(a) There exists λ_0 such that for any $\lambda < \lambda_0$ the minimization solution (13) *is of $\mathcal{O}(\Delta x^2)$ away from the LxW solution.*
(b) The l_1 modified Lax–Wendroff scheme is a second order consistent method for $\lambda < \lambda_0$.
(c) The l_1 modified Lax–Wendroff scheme is conservative.
(d) The l_1 modified Lax–Wendroff scheme converges to a weak solution to (1) *in the zero limit of Δx.*

Proof Without loss of generality, we consider the solution at time step t_n. To show (a), we can simply utilize the convergence result in Theorem 4, namely,

$$\lim_{\lambda \to 0^+} \mathsf{U}^{n+1}(\lambda) = \mathsf{U}^{n+1/2},$$

which means that there exists λ_0 such that for any $\lambda < \lambda_0$ we have

$$\|\mathsf{U}^{n+1}(\lambda) - \mathsf{U}^{n+1/2}\|_2 \leq \mathcal{O}(\Delta x^2)\,.$$

To show (b) we simply use the triangle inequality:

$$\|\mathsf{U}^{n+1}(\lambda) - \mathsf{U}(t_{n+1})\| \leq \|\mathsf{U}^{n+1}(\lambda) - \mathsf{U}^{n+1/2}\| + \|\mathsf{U}^{n+1/2} - \mathsf{U}(t_{n+1})\| = \mathcal{O}(\Delta x^2)\,,$$

where $\mathsf{U}(t_{n+1})$ denotes the true solution at t_{n+1}. Here we have used the fact that the LxW is second order, and that an appropriate $\lambda < \lambda_0$ is chosen.

Observe that (c) is a natural consequence from the design of the problem since LxW method is a standard finite element method with conservation of mass encoded, and the mass-conservation constraint is added in Stage 2.

To demonstrate (d) and conclude the proof, recall that according to Lax–Wendroff Theorem, a conservative and consistent numerical scheme naturally converges to the weak solutions of a hyperbolic conservation law. □

Proposition 4 (Total Variation Diminishing (TVD)) *There exists λ_{TV} so that for all $\lambda > \lambda_{TV}$, the l_1-Modified-Lax–Wendroff scheme is Total Variation Diminishing (TVD), that is*

$$TV(U^{n+1}(\lambda)) \leq TV(U^n)\,. \tag{21}$$

Proof We first show that the method is TVD in Stage 2, namely that

$$\mathsf{TV}(\mathsf{U}^{n+1}) \leq \mathsf{TV}(\mathsf{U}^{n+1/2})\,.$$

To do so, we first observe that if $\mathcal{L} = \mathcal{L}^1$, then by definition, $\mathsf{TV}(\mathsf{U}) = \|\mathcal{L}\mathsf{U}\|_1$. Thus

$$\begin{aligned}\lambda\mathsf{TV}(\mathsf{U}^{n+1}) = \lambda\left\|\mathcal{L}\mathsf{U}^{n+1}\right\|_1 &\leq \frac{1}{2}\left\|\mathsf{U}^{n+1/2}-\mathsf{U}^{n+1}\right\|_2 + \lambda\left\|\mathcal{L}\mathsf{U}^{n+1}\right\|_1\\ &\leq \frac{1}{2}\left\|\mathsf{U}^{n+1/2}-\mathsf{U}^{n+1/2}\right\|_2 + \lambda\left\|\mathcal{L}\mathsf{U}^{n+1/2}\right\|_1\\ &= \lambda\left\|\mathcal{L}\mathsf{U}^{n+1/2}\right\|_1 = \lambda\mathsf{TV}(\mathsf{U}^{n+1/2})\,.\end{aligned}$$

The inequality holds for all λ, meaning the solution has its TV norm suppressed regardless of the choice of the regularization parameter. To show (21), however, some constraints on λ must be imposed. In fact LxW increases the TV norm of the solution and thus $\mathsf{TV}(\mathsf{U}^{n+1/2}) \geq \mathsf{TV}(\mathsf{U}^n)$. To compensate for this, let us first denote $\hat{\mathsf{b}}(\lambda)$ as the solution to the corresponding lasso problem (18). Then according to Proposition 1, $\|\hat{\mathsf{b}}(\lambda)\|_1$ continuously depends on λ. Furthermore, by Theorem 2, we have $\|\hat{\mathsf{b}}(\lambda)\|_1 = \|\mathcal{L}\mathsf{U}^{n+1}(\lambda)\|_1$, meaning $\mathsf{TV}(\mathsf{U}^{n+1}(\lambda)) = \|\mathcal{L}\mathsf{U}^{n+1}(\lambda)\|_1$ continuously depends on λ. Notice that

(1) $\mathsf{TV}(\mathsf{U}^{n+1}(\lambda)) = 0$ for $\lambda = \infty$;
(2) $\mathsf{TV}(\mathsf{U}^{n+1}(\lambda)) = \mathsf{TV}(\mathsf{U}^{n+1/2}) \geq \mathsf{TV}(\mathsf{U}^n)$ for $\lambda = 0$.

Hence the TV norm of $\mathsf{U}^{n+1}(\lambda)$ continuously goes from 0 to a value bigger than $\mathsf{TV}(\mathsf{U}^n)$ as λ decreases, implying that there must be a λ_{TV} so that for all $\lambda > \lambda_{\mathrm{TV}}$, the TV norm of the new solution is smaller than that from the previous step, which finishes the proof. □

Proposition 5 *The l_1-modified-Lax–Wendroff scheme satisfies the weak entropy condition. More specifically, for every fixed convex function ϕ, there exists a constant α such that*

$$\phi(U^{n+1}(\lambda)) \leq \alpha\phi(U^{n+1/2}) + \mathcal{O}(\lambda)\,.$$

Here $U^{n+1}(\lambda)$ is the optimizer of (13) with λ being the regularizer coefficient. In particular, for $\phi(U) = \|U\|_2^2$, $\alpha = 1$.

Proof We first note that $\mathsf{U}^{n+1}(0) = \mathsf{U}^{n+1/2}$ and the entropy condition is automatically satisfied. For $\lambda \neq 0$, we first set $\phi(\mathsf{x}) = \frac{1}{2}\|\mathsf{x}\|_2^2$. Then

$$\begin{aligned}\phi(\mathsf{U}^{n+1}(\lambda)) &\leq \phi(\mathsf{U}^{n+1}(\lambda)) + \lambda\|\mathcal{L}\mathsf{U}^{n+1}(\lambda)\|_1\\ &\leq \phi(\mathsf{U}^{n+1}(\lambda) - \mathsf{U}^{n+1/2}) + \phi(\mathsf{U}^{n+1/2}) + \lambda\|\mathcal{L}\mathsf{U}^{n+1}(\lambda)\|_1\\ &= \frac{1}{2}\|\mathsf{U}^{n+1}(\lambda) - \mathsf{U}^{n+1/2}\|_2^2 + \phi(\mathsf{U}^{n+1/2}) + \lambda\|\mathcal{L}\mathsf{U}^{n+1}(\lambda)\|_1\end{aligned}$$

$$\leq \frac{1}{2}\|\mathsf{U}^{n+1/2} - \mathsf{U}^{n+1/2}\|_2^2 + \phi(\mathsf{U}^{n+1/2}) + \lambda\|\mathcal{L}\mathsf{U}^{n+1/2}\|_1$$
$$= \phi(\mathsf{U}^{n+1/2}) + \lambda\|\mathcal{L}\mathsf{U}^{n+1/2}\|_1 .$$

In the derivation, the second inequality comes from Jensen's inequality, and the third comes from the fact that $\mathsf{U}^{n+1}(\lambda)$ is the minimizer of the lasso problem. Note that $\|\mathcal{L}\mathsf{U}^{n+1/2}\|_1$ is pre-determined by the LxW solution. If ϕ is not a quadratic function, given it is a convex function, one can always find two constants c and C such that

$$c\|\mathsf{x}\|^2 \leq \phi(\mathsf{x}) \leq C\|\mathsf{x}\|^2 ,$$

then considering the optimizer $\mathsf{U}^{n+1}(\frac{\lambda}{2C})$

$$\begin{aligned}
\phi\left(\mathsf{U}^{n+1}(\frac{\lambda}{2C})\right) &\leq \phi\left(\mathsf{U}^{n+1}(\frac{\lambda}{2C})\right) + \lambda\|\mathcal{L}\mathsf{U}^{n+1}\left(\frac{\lambda}{2C}\right)\|_1 \\
&\leq C\|\mathsf{U}^{n+1}\left(\frac{\lambda}{2C}\right)\|_2^2 + \lambda\|\mathcal{L}\mathsf{U}^{n+1}\left(\frac{\lambda}{2C}\right)\|_1 \\
&\leq C\|\mathsf{U}^{n+1/2} - \mathsf{U}^{n+1}\left(\frac{\lambda}{2C}\right)\|_2^2 + C\|\mathsf{U}^{n+1/2}\|_2^2 + \lambda\|\mathcal{L}\mathsf{U}^{n+1}\left(\frac{\lambda}{2C}\right)\|_1 \\
&\leq 2C\left(\frac{1}{2}\|\mathsf{U}^{n+1/2} - \mathsf{U}^{n+1}(\frac{\lambda}{2C})\|_2^2 + \frac{\lambda}{2C}\|\mathcal{L}\mathsf{U}^{n+1}(\frac{\lambda}{2C})\|_1\right) + C\|\mathsf{U}^{n+1/2}\|_2^2 \\
&\leq \lambda\|\mathcal{L}\mathsf{U}^{n+1/2}\|_1 + C\|\mathsf{U}^{n+1/2}\|_2^2 \\
&\leq \lambda\|\mathcal{L}\mathsf{U}^{n+1/2}\|_1 + \frac{C}{c}\phi(\mathsf{U}^{n+1/2}) .
\end{aligned}$$

Once again we used the Jensen's inequality and the fact that $\mathsf{U}^{n+1}(\frac{\lambda}{2C})$ is the optimizer of $\frac{1}{2}\|\mathsf{U}^{n+1/2} - \mathsf{U}^{n+1}\|_2^2 + \frac{\lambda}{2C}\|\mathcal{L}\mathsf{U}^{n+1}\|_1$ in the third and the forth inequalities. The proof is complete by setting $\alpha = \frac{C}{c}$. □

4 Numerical Experiments

In this section we discuss how to choose regularization parameter λ and PA operator $\mathcal{L}^m$ order parameter m for a few problems. We start with a simple example to gain physical intuition for the behavior of $\mathcal{L}^m$. Various choices of λ and $\mathcal{L}^m$ will then be compared for Burgers' equation with a stationary shock, Burgers' equation with a moving shock, and finally Euler's equations.

4.1 Effects of Adding an l_1 Penalty Term

To demonstrate how adding the l_1 penalty term affects a numerical solution, we first consider the 6-grid toy example displayed in Fig. 1. In each subfigure, the blue line represents the true solution which contains a shock between the fourth and the fifth grid point, while the green line shows the initial guess containing oscillations that resembles those resulting from the artificial numerical dispersion term in the LxW solution. The red lines in each subfigure are the resulting solution using the LARS algorithm of the corresponding lasso problem, each using different values of λ, aimed to mitigate the oscillations. It is immediately evident that while using a small λ ensures that the lasso solution remains close to the initial guess, as λ

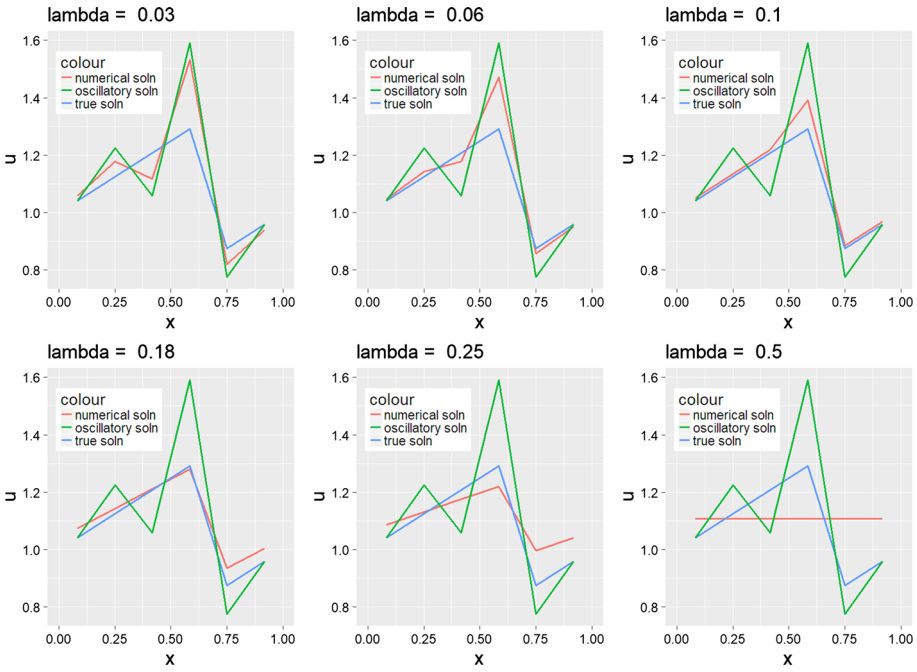

Fig. 1 Toy model used to demonstrate how the l_1 penalty term affects an oscillatory solution. (blue) true solution; (green) oscillatory solution; (red) solution using the LARS algorithm to the corresponding lasso problem. Observe that as λ increase, the solution tends to the constant function (Color figure online)

gradually increases, the oscillations gradually do get eliminated. Finally, for large enough λ, the solution becomes a constant function.

4.2 Burgers' Equation with a Stationary Shock

Burgers' equation provides the simplest model in the study of nonlinear hyperbolic conservation laws. In this first example we choose the initial data so that the solution contains a stationary shock:

$$\partial_t u + \partial_x \left(\frac{u^2}{2}\right) = 0\,, \quad (t,x) \in \mathbb{R}^+ \times [0,1] \quad \text{with} \quad u(t=0,x) = \begin{cases} x & x < 0.5 \\ x-1 & x > 0.5 \end{cases}, \tag{22}$$

and apply periodic boundary conditions. For this particular example, the analytical solution is explicitly given by

$$u(t,x) = \begin{cases} \dfrac{x}{t+1} & x < 0.5 \\ \dfrac{x-1}{t+1} & x > 0.5 \end{cases}.$$

Thus we see that the solution has a stationary shock at $x = 0.5$, but that the shock strength decays in time. To numerically calculate this solution, we use $\Delta x = 0.01$, $\Delta t = \frac{\Delta x}{10}$, and set the final time to be $T = 0.1$ (i.e. $M = 100$ time steps). The solutions for $m = 1$ and 3 in $\mathcal{L}^m$ for various choices of λ are discussed in Sects. 4.2.1 and 4.2.2 respectively.

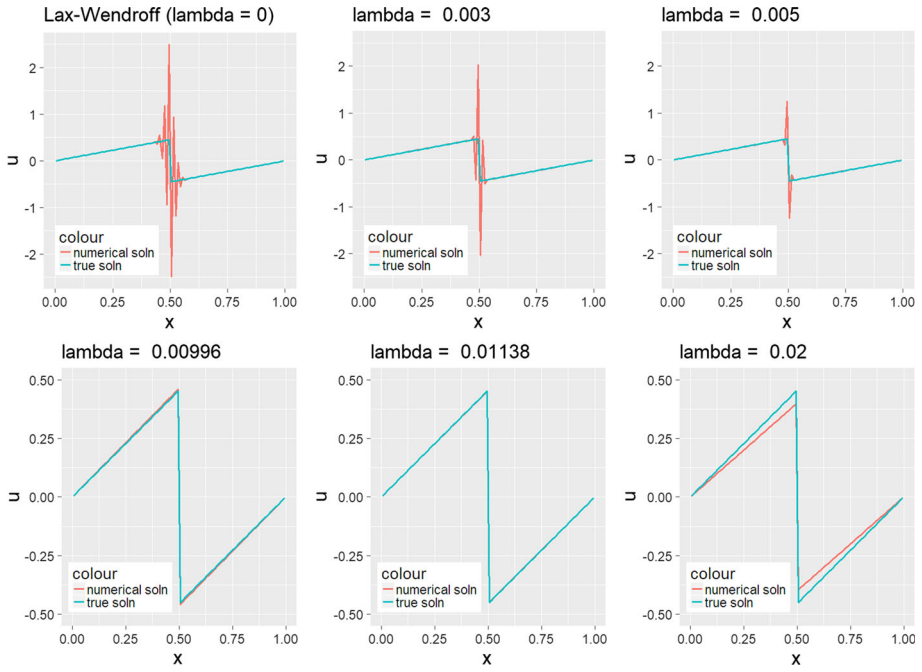

Fig. 2 Numerical solution from Algorithm 1 using $\mathcal{L}^1$ with different choices of λ to (22) at time $T = .1$ with $\Delta x = .01$, and $\Delta t = \frac{\Delta x}{10}$

4.2.1 Setting $\mathcal{L} = \mathcal{L}^1$

We first consider using the first order PA operator, $\mathcal{L}^1$, in Stage 2 of Algorithm 1 for solving (22). Figure 2 compares the solution at final time $T = .1$ for various choices of λ. It is evident that larger values of λ yield greater "smoothing" effects, with the best (visual) solution occurring when $\lambda \sim .01$.

Applying l_1 minimization at each time step is clearly cost prohibitive for most problems. Thus it is natural to ask if it is possible to apply it less often, or even just one time after the final time of $T = .1$, as a way to post-process the solution. Figure 3 demonstrates that using the l_1 regularization as a post-processor performs comparably to the method provided in Algorithm 1. In both cases larger values for λ yield greater smoothing effects, and for some critical threshold $\lambda > \lambda_{CT}$, the smoothing is so strong that the numerical solution no longer can recover the true solution. It is interesting to observe that the λ used for the post-processing l_1 regularization is nearly 100 (M) times the threshold for the case when the minimization is conducted at each time step.

From the results in Figs. 2 and 3, it is apparent that at least in the case of a stationary shock, the l_1 minimization should be used only as a post-processing step. We could still gain more insight into choosing λ, by asking if it is possible to choose λ a-priori based on the desirable properties of numerical conservation laws. Specifically, as shown in Proposition 4 we know that there does exist a threshold λ_{TV} so that the scheme is TVD for all $\lambda > \lambda_{\text{TV}}$. Determining λ_{TV} is not trivial. Fortunately, however, the LARS algorithm provides the entire solution path, which allows us to record the TV norm of the solution as a function of λ. Hence to shed more light on how to choose λ for the stationary shock problem, we traced

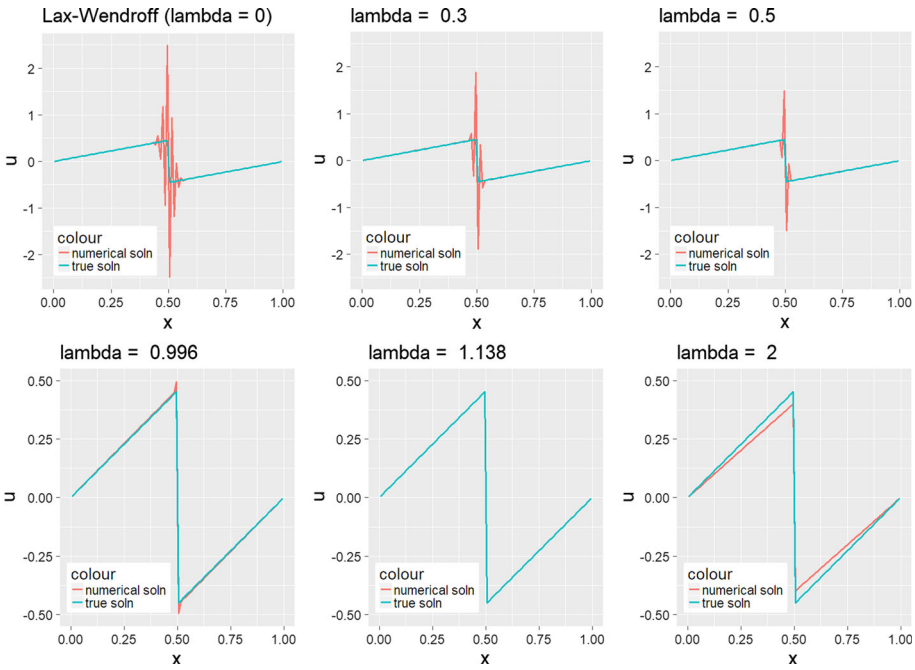

Fig. 3 Numerical solution using different choices for λ obtained by applying l_1 regularization as a single post processing step on the LxW solution for (22) at time $T = .1$ with $\mathcal{L}^1$, $\Delta x = .01$, and $\Delta t = \frac{\Delta x}{10}$

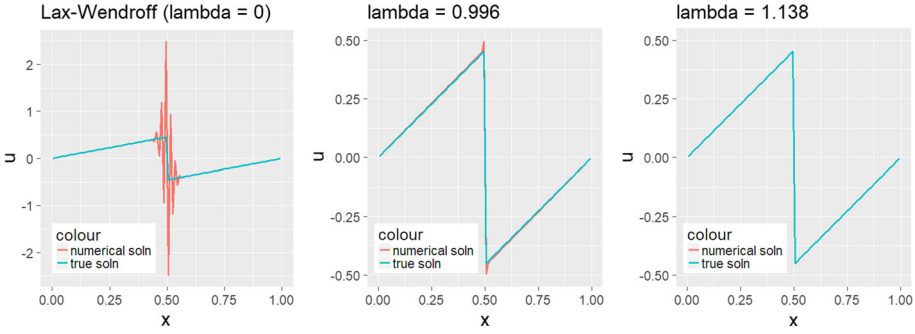

Fig. 4 Solution to (22) using regularization at the final time step with different λ. (left) $\lambda = 0$, or LxW solution. (middle) Solution using the smallest λ that guarantees the TVD property as obtained by the LARS algorithm at the final time step. (right) Solution using the smallest λ that guarantees the TVD property as obtained from the analytical solution at $T = .1$. Here we use $\mathcal{L}^1$

the solution's TV norm by following as λ decayed from ∞ to 0, while stopping the LARS algorithm whenever the TVD property was achieved. Note that while it is technically possible to calculate a different $\lambda = \lambda(n)$ at each time step in Stage 2 in Algorithm 1, it would be extremely costly. Hence to obtain a TVD approximation, we simply compare the TV norm of the final time step solution, $TV(\mathsf{U}^M)$, to that of the initial value, $TV(\mathsf{U}^0)$. In this way, we found that $\lambda_{\text{TV}} = 0.996$ provided the smallest λ guaranteeing TVD. The corresponding solution is shown in Fig. 4. Finally we note that for the stationary shock problem we have the exact solution, with the TV norm of $u(.1, x)$ being smaller than that of $u(0, x)$. By using

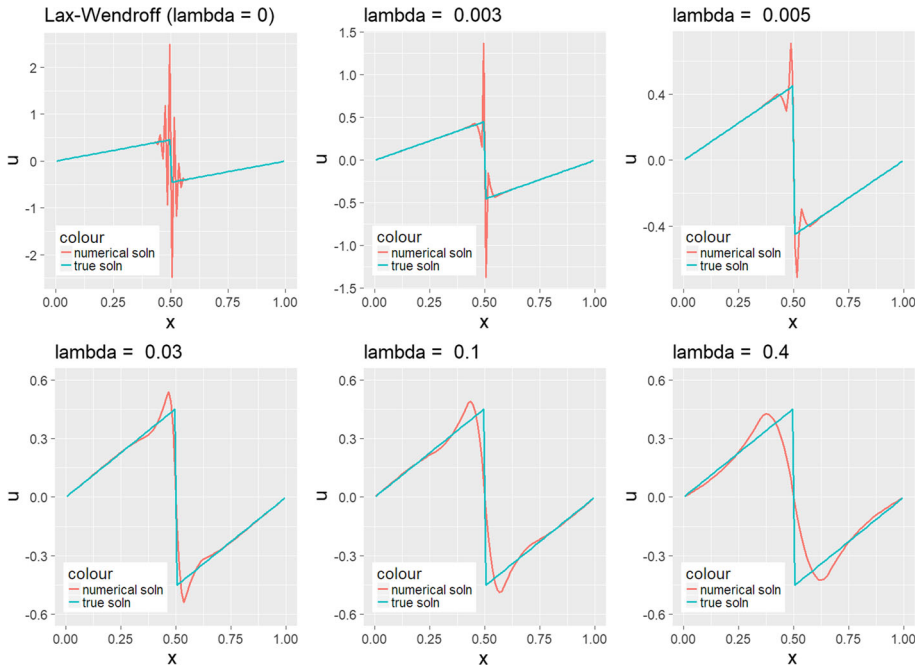

Fig. 5 Numerical solution from Algorithm 1 using $\mathcal{L}^3$ with different choices of λ to (22) at time $T = .1$ with $\Delta x = .01$, and $\Delta t = \frac{\Delta x}{10}$

$TV(u(.1, x))$ as a threshold, we obtain $\lambda_{\text{TV}} = 1.138$. The corresponding solution is plotted in Fig. 4 (right).

To conclude our remarks for solving the stationary shock problem with $\mathcal{L}^1$, we note that the accuracy of the solution heavily depends on how λ is chosen. If λ is too small, there will still be artificial oscillations. On the other hand, if λ is too large, the solution will dissipate to strongly. As noted previously, the appropriate λ will balances the accuracy and the smoothness, or equivalently the fidelity and regularization terms. To accomplish this, we apply the LARS algorithm to obtain a solution path that stops when the TVD property is achieved.

4.2.2 Setting $\mathcal{L} = \mathcal{L}^3$

Higher order PA operators yield faster convergence to the approximation of a jump function in smooth regions. To study these effects on numerical hyperbolic conservation laws, we consider using $\mathcal{L}^3$ for the l_1 regularization. As in Sect. 4.2.1, here we use various choices for λs and observe the accuracy of the corresponding numerical solution. Figures 5 and 6 respectively show the numerical solution when the l_1 minimization is applied at every time step and at the final time step (as a post-processor). We then again look for the threshold λ_{TV} that yields the TVD property at the final time step-first by comparing the TV norms of the numerical solution U^M and U^0 (yielding $\lambda_{\text{TV}} = .538$), and then by using the TV norm of the analytic solution at $T = .1$ (yielding $\lambda_{\text{TV}} = .571$). The corresponding solutions are plotted in Fig. 7.

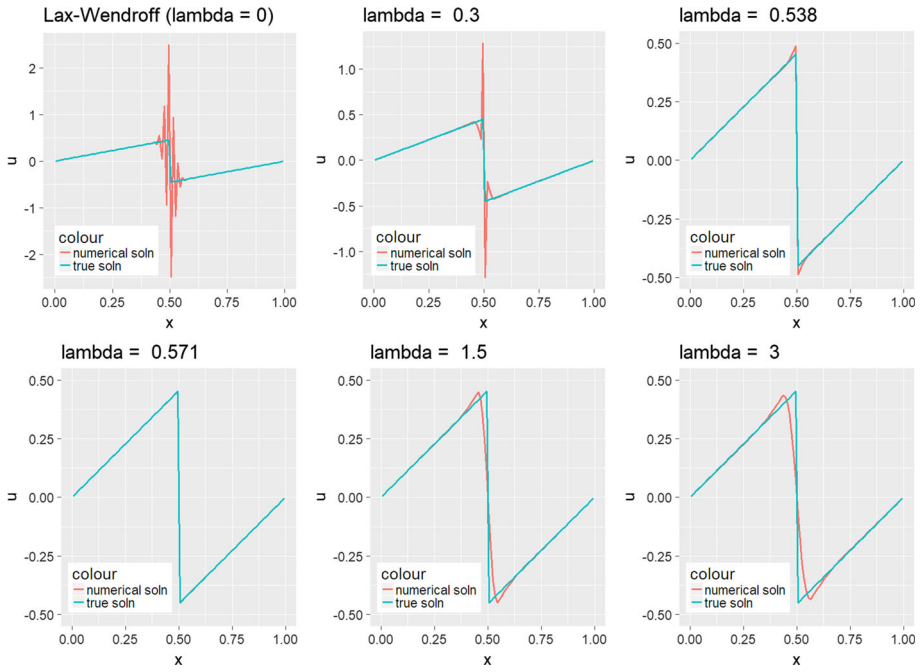

Fig. 6 Numerical solution using different choices for λ obtained by applying l_1 regularization as a single post processing step on the LxW solution for (22) at time $T = .1$ with $\mathcal{L}^3$, $\Delta x = .01$, and $\Delta t = \frac{\Delta x}{10}$

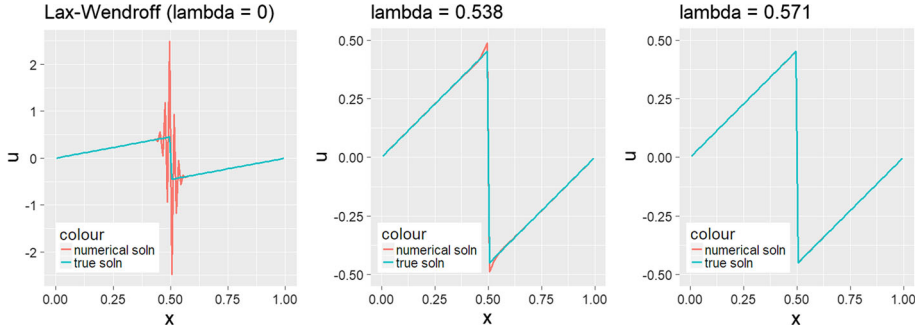

Fig. 7 Solution to (22) using regularization at the final time step with different λ. (left) $\lambda = 0$, or LxW solution. (middle) Solution using the smallest λ that guarantees the TVD property as obtained by the LARS algorithm at the final time step. (right) Solution using the smallest λ that guarantees the TVD property as obtained from the analytical solution at $T = .1$. Here we use $\mathcal{L}^3$

As is readily observed, the results using $\mathcal{L}^3$ are too oscillatory for this problem. More study is needed to determine if using $\mathcal{L}^3$ would yield relatively better results than using $\mathcal{L}^1$ in under-resolved environments, as is demonstrated to be the case for function reconstruction in [1], or if the conservation law solution has more variation in smooth regions. It may also be advantageous to use a weighted l_1 norm in this case, see e.g. [4]. This will be the topic of future investigations.

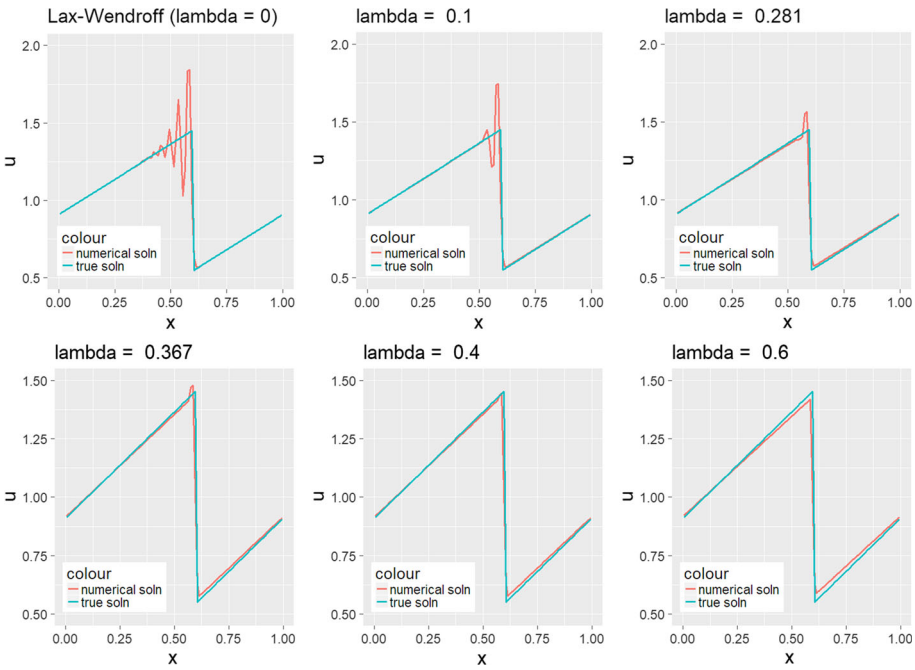

Fig. 8 Numerical solution using different choices for λ obtained by applying l_1 regularization as a single post processing step on the LxW solution for (23) at time $T = .1$ with $\mathcal{L}^1$, $\Delta x = .01$, and $\Delta t = \frac{\Delta x}{10}$

4.3 Burgers' Equation with a Moving Shock

As our third example, we consider Burgers' equation with a moving shock, given by

$$\partial_t u + \partial_x \left(\frac{u^2}{2}\right) = 0\,, \quad (t,x) \in \mathbb{R}^+ \times [0,1] \quad \text{with} \quad u(t=0,x) = \begin{cases} x+1 & x < 0.5 \\ x & x > 0.5 \end{cases}. \tag{23}$$

The solution has a shock which moves right with shock speed 1. The analytical solution is computed as

$$u = \begin{cases} \dfrac{x+1}{t+1} & x < 0.5 + t \\ \dfrac{x}{t+1} & x > 0.5 + t \end{cases}$$

for $t < 0.5$. We compute the equation up to $T = 0.1$ using LxW method with grid size $\Delta x = 0.01$ and time step $\Delta t = \frac{\Delta x}{10}$ ($M = 100$). Figure 8 shows the numerical solutions with l_1 minimization applied only at the final time step for different values of λ. When $\lambda = 0$ we recover the standard LxW solution, and, as expected, increasing λ yields more mitigated oscillations. As in the previous example, we determine the λ that ensures the TVD property at the final time. Comparing the TV norms of the numerical solutions at U^M and U^0 yields $\lambda_{\text{TV}} = 0.281$, while using the exact solution for $u(.1, x)$ to obtain the required TV threshold yields $\lambda_{\text{TV}} = 0.367$. The solutions for these preset λ values are plotted in Fig. 9.

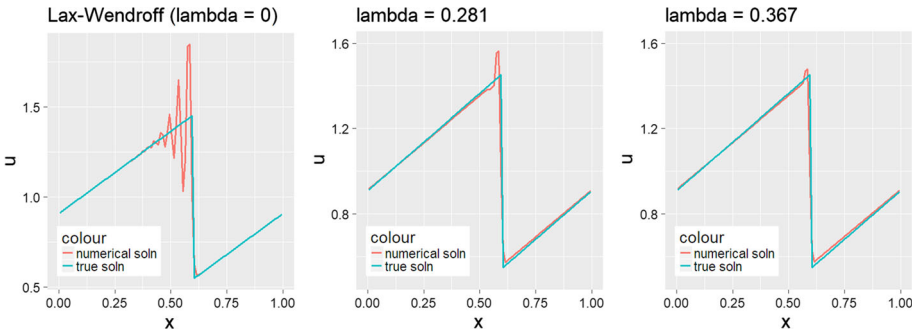

Fig. 9 Solution to (23) using regularization at the final time step with different λ. (left) $\lambda = 0$, or LxW solution. (middle) Solution using the smallest λ that guarantees the TVD property as obtained by the LARS algorithm at the final time step. (right) Solution using the smallest λ that guarantees the TVD property as obtained from the analytical solution at $T = .1$. Here we use $\mathcal{L}^1$

4.4 Euler's Equations

As a final example, we consider the Euler's equations, given by

$$\frac{\partial}{\partial t}\begin{pmatrix}\rho\\ \rho u\\ E\end{pmatrix} + \frac{\partial}{\partial x}\begin{pmatrix}\rho u\\ \rho u^2 + p\\ u(E+p)\end{pmatrix} = 0\,,$$

where ρ, u, and E denote density, velocity, and energy, respectively. Pressure is defined as $p = (\gamma - 1)(E - \frac{\rho u^2}{2})$. For this experiment we set $\gamma = 1.4$ and use the shock tube problem initial conditions

$$\begin{cases}\rho_0 = 1,\ u_0 = 0,\ E_0 = 2.5,\ p_0 = 1 & x < 0\\ \rho_0 = 0.125,\ u_0 = 0,\ E_0 = 0.25,\ p_0 = 0.1 & x > 0\end{cases}.$$

Note the jump discontinuity at $x = 0$ in the initial data. We compute the solution on $[-0.5, 0.5]$ with grid size $\Delta x = \frac{1}{400}$ and time step $\Delta t = \frac{\Delta x}{10}$ up to $T = 0.2$. Based on the results in the previous examples, we apply l_1 minimization only at the final time step as a post-processor for various choices of λ. We note that in this case we also impose conservation of momentum and energy in (3) of Algorithm 1. Figure 10 illustrates that the oscillations that appear at the shocks in the standard LxW solution can be effectively reduced by including an l_1 penalty term.

5 Concluding Remarks

The purpose of this paper was to design and analyze the properties of numerical schemes for conservation laws that combines the Lax Wendroff method with l_1 regularization. A critical component in our scheme (and what differentiates it from the scheme introduced in [17]) is the *conservation constraint*. We demonstrated that our method is equivalent to a lasso problem, and therefore guarantees the existence and uniqueness of the numerical solution. In Theorem 5 we proved consistency, convergence, and conservation of our scheme, while in Propositions 4 and 5 we showed that the method is TVD and satisfies the weak entropy condition for conservation laws. Our results rely on the use of the LARS algorithm for

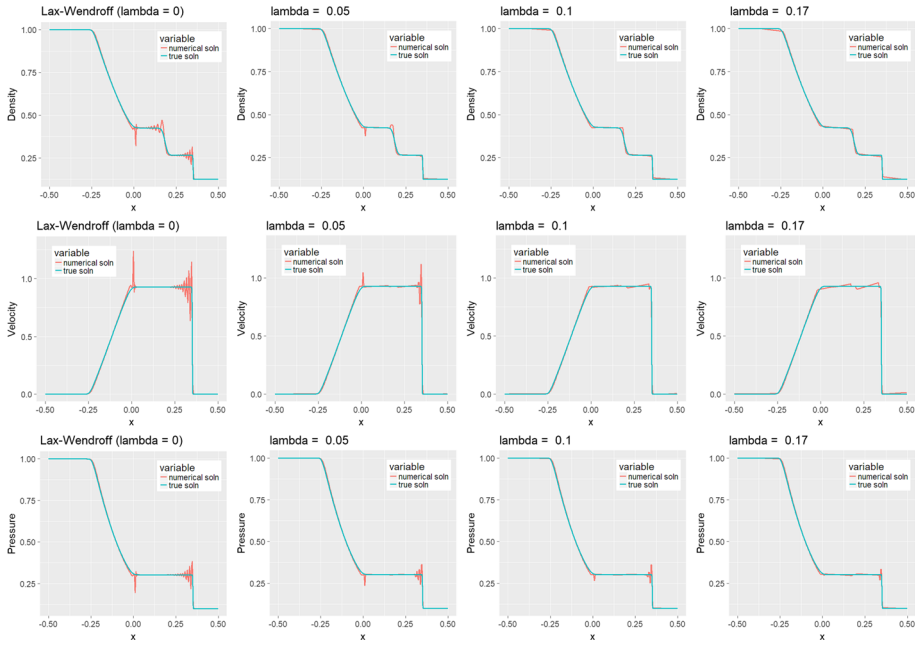

Fig. 10 Solutions of Euler equation with shock tube problem initial conditions. The l_1 minimization is applied at the final time step for different choices of λ

solving the corresponding lasso problem. In practice using the LARS algorithm would be cost prohibitive, so a surrogate technique for implementing Algorithm 1 should be used. For example, the the alternating direction method of multipliers (ADMM) is used in [17] for similar examples. It is not clear, however, that the convergence and conservation properties are maintained for surrogate implementation procedures, so more investigation is needed. As observed in our numerical results, it is not necessary to apply l_1 regularization at every time step. Specifically, it sufficed to apply the regularization only as a post-processing algorithm (after 100 time steps in the Burgers' equation examples). This suggests that one might be able to apply l_1 regularization after a fixed number of time steps to reduce unwanted oscillations, maintain stability and be TVD for those problems for which we seek long term solutions. We observed that the optimal choice of λ for the post-processed solution corresponded to the number of time steps. In particular, for M time steps we saw $\lambda_M \sim M\lambda$, where λ is the l_1 regularization parameter corresponding to the minimization performed at every time step, while λ_M corresponds to the regularization parameter when the minimization is performed only after M time steps. More rigorous analysis is needed, however, to see if λ can always be chosen accordingly. Finally, numerical PDE solvers other than Lax Wendroff may be better suited to use in Stage 1 of Algorithm 1. These issues all remain for future investigations.

References

1. Archibald, R., Gelb, A., Platte, R.B.: Image reconstruction from undersampled Fourier data using the polynomial annihilation transform. J. Sci. Comput. **67**(2), 432–452 (2016)
2. Archibald, R., Gelb, A., Yoon, J.: Polynomial fitting for edge detection in irregularly sampled signals and images. SIAM J. Numer. Anal. **43**(1), 259–279 (2005)

3. Arvanitis, C., Makridakis, C., Sfakianakis, N.I.: Entropy conservative schemes and adaptive mesh selection for hyperbolic conservation laws. J. Hyperb. Differ. Equ. **07**(03), 383–404 (2010)
4. Candes, E.J., Wakin, M.B., Boyd, S.P.: Enhancing sparsity by reweighted ℓ_1 minimization. J. Fourier Anal. Appl. **14**(5), 877–905 (2008)
5. Don, W.-S., Gao, Z., Li, P., Wen, X.: Hybrid compact-WENO finite difference scheme with conjugate Fourier shock detection algorithm for hyperbolic conservation laws. SIAM J. Sci. Comput. **38**(2), A691–A711 (2016)
6. Friedman, J., Hastie, T., Tibshirani, R.: Regularization paths for generalized linear models via coordinate descent. J. Stat. Softw. **33**(1), 1–22 (2010)
7. Gottlieb, D., Shu, C.-W.: On the Gibbs phenomenon and its resolution. SIAM Rev. **39**(4), 644–668 (1997)
8. Guermond, J.-L., Marpeau, F., Popov, B.: A fast algorithm for solving first-order PDEs by l1-minimization. Commun. Math. Sci. **6**(1), 199–216 (2008)
9. Gustafsson, B., Kreiss, H.-O., Oliger, J.: Time-Dependent Problems and Difference. Wiley, London (2013)
10. Hou, T.Y., Li, Q., Schaeffer, H.: Sparse + low-energy decomposition for viscous conservation laws. J. Comput. Phys. **288**(C), 150–166 (2015)
11. Lavery, J.E.: Solution of steady-state one-dimensional conservation laws by mathematical programming. SIAM J. Numer. Anal. **26**(5), 1081–1089 (1989)
12. Lavery, J.E.: Solution of steady-state, two-dimensional conservation laws by mathematical programming. SIAM J. Numer. Anal. **28**(1), 141–155 (1991)
13. LeVeque, R.J.: Numerical Methods for Conservation Laws. Springer, Berlin (1992)
14. LeVeque, R.J.: Finite Volume Methods for Hyperbolic Problems. Cambridge Texts in Applied Mathematics. Cambridge University Press, Cambridge (2002)
15. LeVeque, R.J.: Finite Difference Methods for Ordinary and Partial Differential Equations: Steady-State and Time-Dependent Problems. Society for Industrial and Applied Mathematics, Philadelphia (2007)
16. Liska, R., Wendroff, B.: Composite schemes for conservation laws. SIAM J. Numer. Anal. **35**(6), 2250–2271 (1998)
17. Scarnati, T., Gelb, A., Platte, R.B.: Using *l*1 regularization to improve numerical partial differential equation solvers. J. Sci. Comput. **75**, 225–252 (2018)
18. Schaeffer, H., Caflisch, R., Hauck, C.D., Osher, S.: Sparse dynamics for partial differential equations. Proc. Nat. Acad. Sci. **110**(17), 6634–6639 (2013)
19. Shu, C.-W., Wong, P.S.: A note on the accuracy of spectral method applied to nonlinear conservation laws. J. Sci. Comput. **10**(3), 357–369 (1995)
20. Tadmor, E.: Convergence of spectral methods for nonlinear conservation laws. SIAM J. Numer. Anal. **26**(1), 30–44 (1989)
21. Tadmor, E.: Shock capturing by the spectral viscosity method. Comput. Methods Appl. Mech. Eng. **80**(1), 197–208 (1990)
22. Tadmor, E., Waagan, K.: Adaptive spectral viscosity for hyperbolic conservation laws. SIAM J. Sci. Comput. **34**(2), A993–A1009 (2012)
23. Tibshirani, R.J.: The lasso problem and uniqueness. Electron. J. Stat. **7**, 1456–1490 (2013)
24. van den Berg, E., Friedlander, M. P.: SPGL1: a solver for large-scale sparse reconstruction (2007, June). http://www.cs.ubc.ca/labs/scl/spgl1
25. van den Berg, E., Friedlander, M.P.: Probing the pareto frontier for basis pursuit solutions. SIAM J. Sci. Comput. **31**(2), 890–912 (2008)