Journal of Scientific Computing manuscript No.
(will be inserted by the editor)

Using ℓ_1 Regularization to Improve Numerical Partial Differential Equation Solvers

Theresa Scarnati · Anne Gelb · Rodrigo Platte

Received: date / Accepted: date

Abstract Sparse regularization plays a central role in many recent developments in imaging and other related fields. However, it is still of limited use in numerical solvers for partial differential equations (PDEs). In this paper we investigate the use of ℓ_1 regularization to promote sparsity in the shock locations of hyperbolic PDEs. We develop an algorithm that uses a *high order* sparsifying transform which enables us to effectively resolve shocks while still maintaining stability. Our method does not require a shock tracking procedure nor any prior information about the number of shock locations. It is efficiently implemented using the alternating direction method of multipliers. We present our results on one and two dimensional examples using both finite difference and spectral methods as underlying PDE solvers.

Keywords numerical conservation laws · ℓ_1 regularization · alternating direction method of multipliers · image denoising;

Mathematics Subject Classification (2000) 65M06 · 65M12 · 65M70 · 65F10

1 Introduction

Hyperbolic systems of partial differential equations (PDEs) model a variety of phenomena in fields such as gas dynamics, acoustics, elastodynamics, optics and

This work is supported in part by the grants NSF-DMS 1502640 and AFOSR FA9550-15-1-0152.

Theresa Scarnati
School of Mathematical and Statistical Sciences, Arizona State University, Tempe, AZ 85287.
Tel.: 814-440-2741
E-mail: tscarnat@asu.edu

Anne Gelb
Department of Mathematics, Dartmouth College, 27 N. Main St., Hanover, NH 03755. E-mail: annegelb@math.dartmouth.edu

Rodrigo Platte
School of Mathematical and Statistical Sciences, Arizona State University, Tempe, AZ 85287.
E-mail: rplatte@asu.edu

geophysics, [20]. Solutions to hyperbolic PDEs often contain discontinuities such as shock waves and fronts, which can develop in finite time even when the initial conditions are smooth. Although numerical algorithms for solving hyperbolic PDEs have been broadly investigated, [9, 17, 19–21, 35–38], the presence of shock discontinuities still causes complications in the solutions. For example, high-order methods must include viscosity or slope limiters of some kind in order to avoid oscillations that lead to instabilities. On the flip side, too much viscosity yields a loss of resolution near shock discontinuities. Clever upwinding algorithms can reduce the amount of dissipation and still retain sharp features in the solution, but can be more difficult and computationally intensive to implement. Finally, the stability condition for non-linear PDEs is often more stringent, making long term solutions more computationally expensive, [19–21].

The use of ℓ_1 regularization methods to promote sparsity is frequently encountered in imaging and signal processing applications, but they are still of limited use in solving PDEs. Sparse dynamics for hyperbolic PDEs with solutions exhibiting behaviors on multiple spatial scales was investigated in [31], where it was proposed to include the constraint that the approximate solution resides on a sparse subspace of a basis. However, solutions with singularities in the physical domain were not considered. In particular, the solutions were such that they exhibited high frequencies on a small spatial scale so that they had sparse representation when projected onto a Fourier basis. 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 considerable computational complexity. In [17], a method was proposed to approximate solutions to viscous conservation laws. The method utilizes sparse and low-rank decompositions for which sharp-contrast features are separated from smooth, low-energy behaviors. While higher accuracy is achieved, the technique relies on knowledge of the sharp-contrast feature locations, which is not always practical.

A hybrid, spatially-adaptive, weighted, essentially non-oscillatory (WENO) scheme was developed in [9]. With this technique, the spatial scheme is updated based on the given spatial location and the dynamics of the system at a given time. At each iteration of the algorithm, the shock is detected using various shock detection algorithms, then the domain is divided into non-smooth and smooth sub-domains, where appropriate schemes are then used to approximate the solution in each sub-domain The method is non-linear and also relies on accurate shock detection. In [38] 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, [15, 32].

One of the main difficulties with the approaches discussed above is their reliance on detecting discontinuities. Mis-identification leads either to instability, when a shock goes undetected, or to unnecessary dissipation, when a shock is determined to exist in smooth regions. Typically, the algorithms are also computationally intensive, and often require small time steps to satisfy the CFL stability conditions.

In this investigation we offer an alternative approach that incorporates an ℓ_1 regularization term directly into a PDE solver. Our method yields some distinct advantages. First, because we account for the sparsity of the singularities in the physical domain as part of the PDE solver, we do not need to explicitly locate any shock discontinuities or subdivide the domain. Second, because we use the *polynomial annihilation* (PA) operator as our ℓ_1 regularization term, our method is high-order [3, 4, 40]. We note that using TV would recover piecewise-constant solutions; that is, it yields first order approximations. Finally, our numerical results demonstrate that our method maintains stability even when the time step is larger than normally dictated by the CFL condition. Therefore, it is efficient when fast optimization algorithms are employed.

The rest of this paper is organized as follows. In §2 we briefly discuss ℓ_1 regularization, which is often used to numerically solve ill-posed inverse problems. We also review the PA transform which we will use as the ℓ_1 norm operator. In §3 we describe how a given PDE solver can be enhanced using the PA operator in the ℓ_1 regularization term. While the PDE solver can take on a variety of forms, we use standard finite-difference schemes and pseudo-spectral methods to demonstrate our results. To ensure efficiency, we then propose an alternating direction method of multipliers algorithm for solving the resulting convex optimization problem. We present our numerical results in §4. Examples of hyperbolic PDEs include Burgers and Euler's equations. We also test our method on a two dimensional multiplicative noise model, which is often used to reduce speckle in images. In all cases we are better able to resolve functions and images with discontinuities without explicit knowledge of the jump locations. Concluding remarks are provided in §5.

2 Preliminaries

2.1 ℓ_1 regularization

Let $f \in \mathbb{R}^N$ be an unknown image or signal. We assume that some measurable features of f have sparse representation in a particular domain or basis. Define $\hat{f} \in \mathbb{R}^M$ to consist of data samples corresponding to f, and let $\mathcal{F} : \mathbb{R}^N \to \mathbb{R}^M$ be the forward model that projects f to $\hat{f}$. Typically, $\mathcal{F}$ is defined as a linear operator or invertible matrix. Define $H : \mathbb{R}^N \to \mathbb{R}$ to be the regularization operator. The objective is to solve the following optimization problem:

$$\arg\min_f H(f) \quad \text{s.t.} \quad \left|\left|\mathcal{F}f - \hat{f}\right|\right|_2^2 = 0. \tag{1}$$

Note that if the data $\hat{f}$ are under-sampled, then $\mathcal{F}$ may only contain a subset of rows of the forward model matrix. The equality constraint, known as the *data fidelity term*, measures how well the reconstructed function fits the given data for the particular forward model. The *regularization term* H enforces the known sparsity present in the underlying image by penalizing highly varying solutions and restricting the solution space to a desired class of functions. We will consider H to be the ℓ_1 norm of f (or some transformation of f) and note that any ℓ_p norm with $p \leq 1$ will enforce sparse solutions.

Typically, for measured data which is inherently noisy, the related total variation (TV) denoising problem is solved, which relaxes the equality constraint on the data fidelity term. It is formulated as

$$\arg\min_f H(f) \quad \text{s.t.} \quad \left|\left|\mathcal{F}f - \hat{f}\right|\right|_2^2 < \sigma, \tag{2}$$

where $\sigma \in \mathbb{R}$ is some regularization parameter [27, 39]. Equivalently, the TV denoising problem can be formulated as the following unconstrained problem by the introduction of a non-negative regularization parameter $\lambda \in \mathbb{R}$ that represents the trade-off between smoothness and fidelity to the original data as ([11, 24–27])

$$\arg\min_f \left(H(f) + \frac{\lambda}{2} \left|\left| \mathcal{F}f - \hat{f} \right|\right|_2^2 \right). \tag{3}$$

The problem is often solved with $H(f) = TV(f)$, although other operators, such as wavelets, are commonly used in conjunction with the ℓ_1 norm ([11]) as

$$H(f) := ||\mathcal{L}f||_1.$$

2.2 Polynomial annihilation

Consider a function $f : [a, b] \to \mathbb{R}$. For all $y \in (a, b)$, let $f(y^-)$ and $f(y^+)$ respectively denote the left and right hand limits of f. The jump function of f at y is defined at each y as

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

Let $[a, b]$ be partitioned into N points, and let us assume that there is at most one jump within a cell $I_j = [x_j, x_{j+1})$ for $j = 0, ..., N-1$. We can write the jump function (4) on $[a, b]$ as

$$[f](y) = \sum_{j=0}^{N-1} [f](x_j)\chi_j(y), \tag{5}$$

where $[f](x_j)$ is the value of the jump occurring within cell I_j, and χ_j is the indicator function defined as

$$\chi_j(y) = \begin{cases} 1 & y \in I_j \\ 0 & \text{else.} \end{cases}$$

Because $[f](x_j) = 0$ everywhere except in cells containing jumps, we say that $[f](y)$ has sparse representation.

The polynomial annihilation (PA) edge detection method, originally proposed in [4], is defined as

$$\mathcal{L}^m f(y) = \frac{1}{q^m(y)} \sum_{x_j \in S} c_j(y) f(x_j), \tag{6}$$

where m is the order of approximation to (4), S is a local set of $m+1$ grid points about y from the set of given grid points, and the annihilation coefficients c_j are obtained by solving

$$\sum_{x_j} c_j(y) p_l(x_j) = p_l^{(m)}(y), \quad j = 1, ..., m+1. \tag{7}$$

Here p_l, $l = 0, ..., m$, is a basis for the space of polynomials of degree less than or equal to m. An explicit formula based on Newton divided differences was provided in [4] as

$$c_j(y) = \frac{m!}{\omega_j(S)}, \qquad \omega_j(S) := \Pi_{i=1, i\neq j}^m (x_j - x_i), \tag{8}$$

for $j = 1, \cdots, m$. When the grid points are equally spaced, that is $x_j = j\Delta x$ with $\Delta x = (b-a)/N$ and $j = 0, \cdots, N$, (8) leads to

$$c_j = \frac{m!}{\prod\limits_{\substack{k=1\\k\neq j}}^{m+1} (j-k)\,\Delta x}. \tag{9}$$

The normalization factor q^m ensures convergence to the jump value at each discontinuity and is given by

$$q^m(y) = \sum_{x_j \in S^+} c_j(y), \tag{10}$$

where S^+ is the set of points $x_j \in S$ such that $x_j \geq y$. It was shown in [4] that (6) yields *mth* order convergence in smooth regions (outside the stencil containing the jump discontinuity) as long as f has m continuous derivatives in those regions. The method yields a first-order approximation to the jump value in regions where singularities are present. As m increases, oscillations develop in each jump location region. Post-processing methods can reduce the impact of these oscillations. However, because we are using (6) to enforce sparsity, and *not* to explicitly detect edges, the oscillations occurring in jump regions do not directly impact our results. That is, we are only interested in those locations which produce a high response when the function is transformed by the PA operator. Specifically, the PA method was used to design a *high order* ℓ_1 regularization operator in [3] as

$$H(f) := ||\mathcal{L}^m f||_1.$$

Consequently, the sparsity of edges in f are enforced when solving (3). A main advantage in using PA with $m > 1$ is that it enables distinction of jump discontinuities from steep gradients, which is critical in solving non-linear conservation laws and when approximating functions that are not simply piecewise constant. This will help to eliminate the staircase effect that occurs when using the TV operator (equivalently, $m = 1$) as the regularization operator.

In the current investigation, (6) is included inside the optimization problem (3), and therefore only solved on the discrete set of grid points $y = x_j$, $j = 0, \cdots, N$. Hence (10) can also be determined explicitly and we are able to write the PA transform matrix for any integer m. For example, when considering a uniform grid spread over the domain $[a, b]$ and a periodic solution f, we have

$$\mathcal{L}^1 = \begin{bmatrix} 1 & -1 & & & \\ & 1 & -1 & & \\ & & \ddots & \ddots & \\ & & & 1 & -1 \\ -1 & & & & 1 \end{bmatrix}, \quad \mathcal{L}^3 = \frac{1}{2}\begin{bmatrix} 3 & -3 & 1 & & & -1 \\ -1 & 3 & -3 & 1 & & \\ & \ddots & \ddots & \ddots & \ddots & \\ & & 1 & 3 & -3 & 1 \\ 1 & & & 1 & 3 & -3 \\ -3 & 1 & & & -1 & 3 \end{bmatrix}. \tag{11}$$

As noted previously, when $m = 1$, the PA transform is equivalent to using TV regularization. In this regard, the PA transform can be seen as a high-order total variation (HOTV) operator. Also note that (11) assumes periodicity. For non-periodic problems, there is an analogous transform that becomes one-sided as the boundaries are approached, [4]. Finally, even orders may also be used, since they still achieve sparsity in the edge domain.[1] Indeed even order transforms may be advantageous for some PDE solvers, and also in the case where there is some information known about the underlying solution. As will be demonstrated later, using $m = 2$ for Sod's shock tube problem yields the best results because the solution between the shock discontinuities is essentially piecewise linear. More information on the general properties of the PA transform used for recovering piecewise smooth solutions can be found in [3, 4, 34, 40].

3 Proposed algorithm

3.1 Using ℓ_1 regularization in PDE solvers

Consider the one-dimensional non-linear conservation law of the form

$$u_t(x,t) + f(u(x,t))_x = 0 \tag{12}$$

on a bounded domain, Ω. Here, $f : \Omega \to \mathbb{R}$ is the flux function and $u : \Omega \to \mathbb{R}$ is a conserved quantity with appropriate initial and boundary conditions. We will use (12) as a prototype to develop our new ℓ_1 regularization PDE solver. We seek an approximation $U_j^n \in \mathbb{R}^N$ to the solution $u_j^n := u(x_j, t_n)$ of (12) given by

$$U_j^{n+1} = U_j^n - \frac{\Delta t}{\Delta x}\left(F_{j+1/2}^n - F_{j-1/2}^n\right), \quad j = 0, ..., N-1, \tag{13}$$

where $F_{j+1/2}^n$ is an approximation of the average flux along $x = x_{j+1/2}$. That is, in reference to (12),

$$F_{j+1/2}^n \approx \frac{1}{\Delta t}\int_{t_n}^{t_{n+1}} f\left(u\left(x_{j+1/2}, t\right)\right) dt. \tag{14}$$

and Δx and Δt are the appropriate grid size and time step size, respectively. We note that our technique does not require a uniform grid, as the PA operator can be formulated for any set of data points. For our purposes, we will write (13) as

$$U^{n+1} = \mathcal{D}\left(F^n, U^n\right) =: b^n, \tag{15}$$

where U^{n+1} represents the vector of approximations U_j^{n+1} at time t_{n+1}, F^n represents the vector of approximations $F_{j+1/2}^n$ at time t_n for $j = 0, ..., N-1$, and $\mathcal{D}$ is the operator representing the chosen numerical method. Observe that b^n is simply a vector of values explicitly calculated at time t_n which will be incorporated into the data fidelity term. We consider only explicit schemes, since an implicit scheme may lead to non-convexities in the objective function. This will be explored more in future research.

As noted previously, one of the main challenges in solving (12) is balancing the amount of viscosity introduced for stability purposes with maintaining desirable high resolution properties, especially near shocks. This is particularly difficult when

[1] Even orders were not used in [4] because the post-processing techniques used for pinpointing the edges assumed that maximum (minimum) values occurred at the edge, which is true only for odd orders.

the shock locations are unknown. Applying the techniques described in §2.2, we can now adapt (13) to include an ℓ_1 regularization term, reflecting that the jump discontinuities in the solution of (12) are sparse. Specifically, our new algorithm is given by

$$U^{n+1} = \arg\min_{V} \left(||\mathcal{L}^m V||_1 + \frac{\lambda}{2}||V - b^n||_2^2 \right), \tag{16}$$

which is analogous to the convex optimization problem given in (3). Observe that minimizing $||U^{n+1} - b^n||_2^2$ obtains the best solution in the least squares sense, but does not adequately capture the shocks that may appear in (12). However, by augmenting (15) with the PA transform ℓ_1 regularization term, we encourage the solution U^{n+1} to have sparse representation jump function domain. Consequently, we obtain a stable solution without introducing too much artificial viscosity, and thus are able to maintain high resolution near shock locations. Moreover, by using $m > 1$ in (6), we are able to see greater variation in smooth regions. The fidelity term in (16) contains the numerical method chosen to solve (12), and the ℓ_1 regularization term includes the PA transform of order m. The parameter $\lambda \in \mathbb{R}$ determines the influence of the sparsity constraint upon the fidelity term, and its tuning is typically application-dependent. Our numerical results demonstrate that (16) is robust for a range of λ, but more study is needed to verify its impact on stability.

3.1.1 ℓ_1 enhancement for finite difference methods

We first demonstrate our technique using the second order Lax Wendroff (LW) scheme for (12) in the spatial domain $x \in [a, b]$. Assume we are given grid points

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

and define

$$\begin{aligned} U_{j+\frac{1}{2}}^{n+\frac{1}{2}} &= \frac{1}{2}\left(U_j^n + U_{j+1}^n\right) - \frac{\Delta t}{2\Delta x}\left(f(U_{j+1}^n) - f(U_j^n)\right) \\ U_{j-\frac{1}{2}}^{n+\frac{1}{2}} &= \frac{1}{2}\left(U_{j-1}^n + U_j^n\right) - \frac{\Delta t}{2\Delta x}\left(f(U_j^n) - f(U_{j-1}^n)\right). \end{aligned}$$

The LW scheme is then given by

$$U_j^{n+1} = U_j^n - \frac{\Delta t}{\Delta x} \underbrace{\left[f\left(U_{j+\frac{1}{2}}^{n+\frac{1}{2}}\right) - f\left(U_{j-\frac{1}{2}}^{n+\frac{1}{2}}\right)\right]}_{\mathcal{D}(F^{n+1/2})}. \tag{17}$$

It is well known that using (17) to solve (12) results in unwanted oscillations behind shock locations and may also lead to instability, [19,20]. To apply ℓ_1 enhancement, we define

$$b^n = U^n - \frac{\Delta t}{\Delta x}\mathcal{D}(F^{n+1/2}), \tag{18}$$

and insert it directly into (16). We will refer to this particular combination as the ℓ_1 *enhanced LW* method, and note that while we only employ LW in our numerical examples, the ℓ_1 enhancement can be applied to any finite difference method that can be written in the form (16).

In general, (17) is stable for

$$\Delta t \leq \frac{\Delta x}{|\zeta|_{\max}},$$

where $|\zeta|_{\max}$ is the maximum characteristic speed in magnitude. However, the LW method yields significant dissipation, leading to a loss of shock information at each time step. Thus, as will be seen in our results, it is beneficial to consider high-order methods which are better at resolving shocks. Of course, in this case stability becomes a concern as we discuss in the next section.

3.1.2 ℓ_1 enhancement for spectral methods

Spectral methods provide highly accurate approximations for sufficiently smooth functions $u : [a, b] \to \mathbb{R}$. When u contains discontinuities, the resulting Gibbs phenomenon leads to $\mathcal{O}(1)$ oscillations in neighborhoods of discontinuities and first order accuracy in smooth regions. The method becomes unstable as a result of non-linear interactions within the partial differential equation. Filtering or additional viscosity is often introduced to mitigate this problem. However, too much dissipation is undesirable as the shocks are "smeared over" and information is lost. Methods such as (super) spectral viscosity (SV), [36,37], were introduced to apply minimal amounts of diffusion near shocks, and in [38] some attempt was made to localize the effects of dissipation even further by determining the discontinuous regions as time evolved. The SV methods are computationally expensive, as they amount to adding high-order viscosity in the underlying conservation law. Nevertheless, they offer an alternative to standard filtering. In §4 we demonstrate that using ℓ_1 regularization improves the accuracy of standard filtering and vanishing viscosity methods.[2]

We first consider the Fourier pseudo-spectral (PS) approximation of $u : [-\pi, \pi] \to \mathbb{R}$ given by [7,16]

$$P_N u(x,t) = \sum_{k=-N/2}^{N/2-1} \tilde{u}_k(t) e^{ikx}, \qquad \tilde{u}_k(t) = \sum_{j=0}^{N-1} u(x_j,t) e^{-ikx_j}, \tag{19}$$

with $x_j = -\pi + j\Delta x$, and $\Delta x = \frac{2\pi}{N}$. The pseudo-spectral (PS) approximation of (12) takes the form

$$(u_N)_t + (P_N f(u_N))_x = 0, \tag{20}$$

with u_N denoting the numerical solution. As noted previously, even when given a smooth initial condition, the solution to (12) may develop singularities. Due to the non-linear interaction, the resulting Gibbs phenomenon will yield instabilities within finite time, [35].

Let us now define

$$b^n = \{G(U^n) - \Delta t \sum_{k=-N/2}^{N/2-1} ik\tilde{f}_k(t_n) e^{ikx_j}, \qquad j = 0, \cdots, N-1\}, \tag{21}$$

[2] We also applied ℓ_1 enhancement to the spectral viscosity method for the Fourier and Chebyshev cases. Both resulted in improved accuracy that essentially mirrored the approximations displayed in Figures 4 and 7. Hence they are not reported here.

where

$$\tilde{f}_k(t) = \sum_{j=0}^{N-1} f\left(u_N(x_j,t)\right) e^{-ikx_j} \tag{22}$$

and $G(U^n)$ represents a linear time-stepping scheme, e.g. Runge Kutta. All of our numerical examples (except Lax-Wendroff and denoising) use fourth order Runge Kutta. The ℓ_1 enhanced pseudo-spectral method is now constructed by substituting b^n into the fidelity term of (16) and using (6) to enforce sparsity in the jump discontinuity domain.

A small amount of viscosity is often introduced in the numerical solver to reduce oscillations and overshoots resulting from the Gibbs phenomenon, [7,16]. The approximation of (12) takes the form

$$(u_N)_t + (P_N f(u_N))_x = \epsilon (u_N)_{xx}, \tag{23}$$

where $\epsilon > 0$ and u_N denotes the numerical solution. Following the terminology in [7,16], we refer to (23) as the *vanishing viscosity* (VV) method. To incorporate ℓ_1 regularization into (23), we define

$$b^n = \{G(U^n) - \Delta t \sum_{k=-N/2}^{N/2-1} ik\tilde{f}_k(t_n)e^{ikx_j} + \epsilon\Delta t \sum_{k=-N/2}^{N/2-1} k^2\tilde{u}_k(t)e^{ikx}, \quad j=0,...,N-1\} \tag{24}$$

with $\tilde{f}_k(t)$ defined in (22) and $\tilde{u}_k(t)$ defined in (19). The ℓ_1 *enhanced VV method* is constructed by inserting (24) into the (16). We note that (20) and (23) could have been written in collocation form, and subsequently the corresponding ℓ_1 enhanced collocation method. However, applying the FFT is sufficiently efficient in the given form.

Filtering also helps to reduce oscillations and improve stability, [16]. In our experiments, we applied an exponential filter (EF) to the solution after each time step. Enhancement with ℓ_1 regularization is straightforward.

Since most PDEs do not admit periodic solutions, we also consider the Chebyshev collocation method. In this case we modify the interval to $[-1,1]$ and define the Chebyshev grid points as

$$y_j = -\cos\left(\frac{\pi j}{N}\right), \quad j=0,...,N. \tag{25}$$

The Chebyshev approximation of $u(x,t)$ is [7,16]

$$P_N u(x,t) = \sum_{k=0}^{N} \tilde{u}_k(t)T_k(x), \quad \tilde{u}_k(t) = \frac{1}{\tilde{\gamma}_k}\sum_{j=0}^{N} u(x_j)T_k(x_j)w_j, \tag{26}$$

where the Chebyshev polynomials are

$$T_k(x) = \cos(k\arccos(x)), \tag{27}$$

and the weights and normalizing factors are given respectively as

$$w_j = \begin{cases} \frac{\pi}{2N} & j=0,N \\ \frac{\pi}{N} & j=1,...,N-1 \end{cases}, \quad \tilde{\gamma}_k = \begin{cases} \pi & k=0,N \\ \frac{\pi}{2} & k=1,...,N-1 \end{cases}.$$

To improve computational efficiency, we map the Chebyshev points according to [18][3] given by

$$x_j = \frac{2}{\alpha\pi} \arcsin\left(\beta y_j\right), \quad j = 0, ..., N, \tag{28}$$

where we have chosen parameters

$$\alpha = 1 + \frac{2}{N\pi} \log(10^{-5}), \quad \beta = \sin\left(\frac{\alpha\pi}{2}\right). \tag{29}$$

We note that no attempt was made to optimize the mapping parameters. The Chebyshev approximation of (12) takes the form

$$(u_N)_t + \mathcal{D}(P_N f(u_N)) = 0, \tag{30}$$

with u_N denoting the numerical solution and $\mathcal{D}$ denoting the Chebyshev differentiation matrix. Each entry in $\mathcal{D}$ is

$$\mathcal{D}(i,j) = \frac{1}{Nc_j} \sum_{k=0}^{N} \frac{T_k(x_j)T_k'(x_i)}{c_k}, \tag{31}$$

where $c_j = 1$ for $1 \leq j \leq N-1$ and $c_0 = c_N = 2$.[4] In our experiments, we used the software described in [10] to construct the Chebyshev differentiation matrix (31). To implement the sparsity enforcing PDE solver we first define

$$b^n = \{G(U^n) - \Delta t \mathcal{D}(P_N f^n)\}, \tag{32}$$

where

$$P_N f^n = \{f(u_N(x_j, t_n)), \qquad j = 0, \cdots, N\}$$

so that

$$\mathcal{D}\left(P_N f(u_N)\right) \approx \mathcal{D}(P_N f^n).$$

Here again $G(U^n)$ represents an appropriate time stepping method. Inserting (32) into (16) with the PA transform operator (6) yields the ℓ_1 *enhanced Chebyshev method.* Boundary conditions are implemented at the end of each time step. When the enhancement is not implemented, we then apply an exponential filter of order $p = 16$ to maintain stability, [16].

[3] With a small decrease in accuracy, the mapped Chebyshev method allows the time step to increase to $\mathcal{O}(\frac{1}{N})$, [18].

[4] The explicit matrix entries for (31) for (25) can be found in [7, 16]. In our examples, we use the mapped Chebyshev points, [18], so the derivative matrix depends on the chosen grid points x_j.

3.2 The alternating direction method of multipliers (ADMM)

Many techniques have been developed for solving (2). Here we use an alternating direction method of multipliers (ADMM) algorithm, [22, 23, 28, 29], which we describe below. In this regard, it is important to eliminate the non-linear term within the ℓ_1 norm in (16). This is accomplished by formulating an equivalent problem

$$\arg \min_{U^{n+1},g} \left(||g||_1 + \frac{\mu}{2}||U^{n+1} - b^n||_2^2 \quad \text{s.t.} \quad \mathcal{L}^m U^{n+1} = g \right), \tag{33}$$

where we have introduced the slack variable $g \in \mathbb{R}^N$. It is now evident that non-linearities are not present within the ℓ_1 term. To approximate (33), we introduce the Lagrangian multipliers $\sigma, \delta \in \mathbb{R}^N$ and solve the unconstrained minimization problem given by

$$\arg \min_{U^{n+1},g} J_{\sigma,\delta}(U^{n+1}, g) \tag{34}$$

where

$$\begin{aligned} J_{\sigma,\delta}(U^{n+1}, g) := ||g||_1 + \frac{\mu}{2}||U^{n+1} - b^n||_2^2 + \frac{\beta}{2}||\mathcal{L}^m U^{n+1} - g||_2^2 \\ - \langle \mathcal{L}^m U^{n+1} - g, \sigma \rangle - \langle U^{n+1} - b^n, \delta \rangle. \end{aligned} \tag{35}$$

Here $\mu, \beta \in \mathbb{R}$ are non-negative regularization parameters. In particular, if the Lagrangian multipliers are updated a sufficient number of times, then the solution will converge to the solution of the constrained problem. The objective function is minimized using ADMM, [12, 22, 23]. Specifically, the solution is approximated by alternating between minimizations of U^{n+1} and g.

Given the current values of U^{n+1} and the multipliers, the optimal g can be exactly determined using the shrinkage-like formula, [13, 22]:

$$g_{k+1} = \max \left(\left| \mathcal{L}^m U_k^{n+1} - \frac{\sigma_k}{\beta} \right| - \frac{1}{\beta}, 0 \right) \text{sign} \left(\mathcal{L}^m U_k^{n+1} - \frac{\sigma_k}{\beta} \right). \tag{36}$$

The gradient descent method is used to find the minimum over U^{n+1} as

$$U_{k+1}^{n+1} = U_k^{n+1} - \alpha \nabla_{U^{n+1}} J_{\sigma,\delta}(U^{n+1}, g_{k+1}), \tag{37}$$

where the gradient of $J_{\sigma,\delta}(U^{n+1}, g_{k+1})$ with respect to U^{n+1} is given by

$$\begin{aligned} \nabla_{U^{n+1}} J_{\sigma,\delta}(U^{n+1}, g) =& \mu(U^{n+1} - b^n) + \beta \left(\mathcal{L}^m\right)^* \left(\mathcal{L}^m U^{n+1} - g_{k+1}\right) \\ &- \left(\mathcal{L}^m\right)^* \sigma_k - \delta_k, \end{aligned} \tag{38}$$

and the step size α is chosen to give sufficient descent in the gradient direction. The Lagrange multipliers δ and σ are updated at each iteration as

$$\begin{aligned} \sigma_{k+1} &= \sigma_k - \beta(\mathcal{L}^m U_{k+1}^{n+1} - g_{k+1}) \\ \delta_{k+1} &= \delta_k - \mu(U_{k+1}^{n+1} - b^n). \end{aligned} \tag{39}$$

Algorithm 1 describes the general ADMM procedure. More details for Algorithm 1 in the context of synthetic aperture radar imaging can be found in [29].[5]

[5] MATLAB code is available at [28].

Algorithm 1

```
Determine parameters $\mu$, $\beta$ and $tol$.
Initialize $U_0^{n+1}$, $g_0$, $\sigma_0$ and $\delta_0$.
for $k = 0$ to $K$ do
    while $||U_{k+1}^{n+1} - U_k^{n+1}|| > tol$ do
        Minimize $J$ for $g$ according to (36).
        Minimize $J$ for $U^{n+1}$ according to (37) and (38).
    end while
    Update Lagrangian multipliers according to (39).
end for
```

4 Numerical results

We are now ready to demonstrate our ℓ_1 enhanced numerical solver, (16), by employing Algorithm 1, for Burgers' and Euler's equations. Our results show that we are able to resolve solutions *without* explicit knowledge of shock locations. We are also able to relax the usual CFL conditions so that our method is cost efficient.

In each case we will compare our new method to standard techniques based on the log of the point-wise error

$$E_{log}(u_N(x,t)) = \log_{10}|u_N(x,t) - u(x,t)| \tag{40}$$

where $u_N(x,t)$ is the numerical approximation to the true solution $u(x,t)$ at the final time step. To demonstrate the stability gained when using (16), we define

$$\Delta t = \alpha \min|x_{j+1} - x_j|, \quad j = 0, ..., N-1, \tag{41}$$

so that the step size is increased with α. We then measure the error of the solution at a grid point neighboring a discontinuity. Hence we are able to determine the value of α for which each method becomes unstable. Our results show that stability is maintained for larger values of α (and correspondingly Δt) when using the ℓ_1 enhanced PDE solver as compared to traditional solvers.

As a final example, we solve an image denoising problem, demonstrating that the ℓ_1 enhanced PDE solver may be used in multi-dimensions, again without explicit knowledge of the jump discontinuities.

4.1 Burgers' equation

Let $u : (-\pi, \pi) \to \mathbb{R}$ be the solution to

$$\begin{cases} u_t + \left(\frac{1}{2}u^2\right)_x = 0 & x \in (-\pi,\pi), \quad t > 0 \\ u_0(x) = 1 + \frac{1}{2}\sin(x) & x \in (-\pi,\pi), \quad t = 0 \\ u(-\pi,t) = u(\pi,t) & t > 0 \end{cases} \tag{42}$$

A shock develops in the solution to (42) when the wave breaks at time

$$t_b = \min_{x\in[-\pi,\pi]} \left[-\frac{1}{u_0'(x)}\right] = \min_{x\in[-\pi,\pi]} \left[-\frac{2}{\cos(x)}\right] = 2,$$

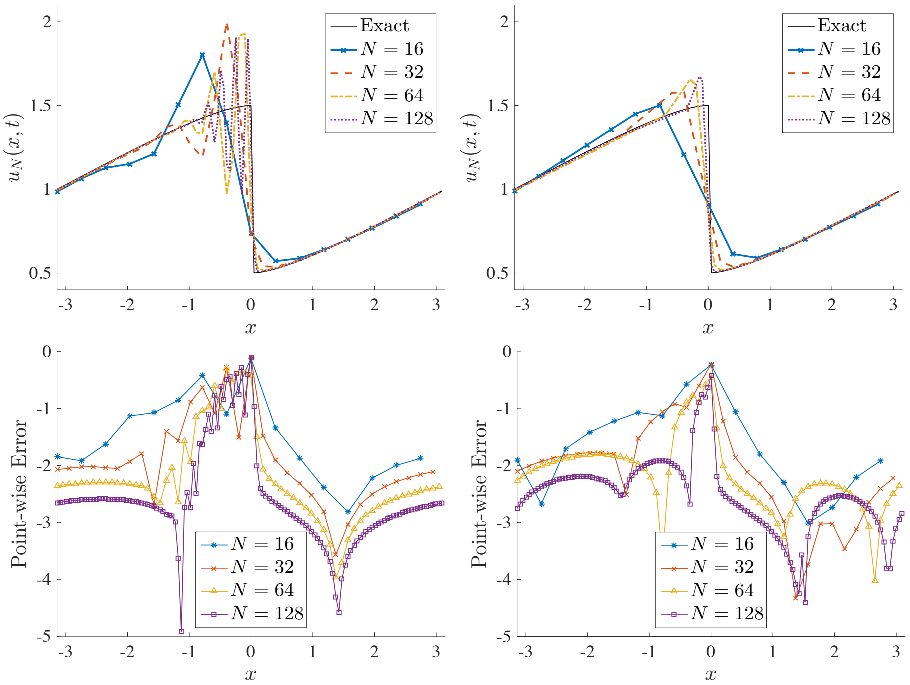

Fig. 1: Approximation of the solution to (42) for $N = 16, 32, 64$ and 128 with $\Delta t = \frac{\Delta x}{2}$. (top-left) LW (top-right) ℓ_1 enhanced LW method. Here we used the PA transform with $m = 2$ and $\lambda = .35$. Point-wise errors given in logarithmic scale for (bottom-left) the LW and (bottom-right) the ℓ_1 enhanced LW methods.

after which it will continue to re-form at each time step and propagate throughout the domain. In our simulations we advance our solutions to time $t = \pi$ so that the shock will be fully developed. The exact solution for this case is calculated using characteristic tracing, as is done in [38].

Figure 1 (a) and (b) compare the LW scheme solution to (42) with and without ℓ_1 regularization. Here we choose $m = 2$ for the PA transform and $\lambda = .35$ in (16). The point-wise errors are shown in Figure 1 (c) and (d). In both cases the time step is chosen as $\Delta t = \frac{\Delta x}{2}$ to ensure stability. As is apparent in Figure 1, augmenting the LW method with ℓ_1 regularization improves the accuracy near the shock locations. Away from the shock locations the accuracy is dictated by the LW approximation.

Figure 2 demonstrates that by enhancing the Fourier pseudo-spectral (PS) method with ℓ_1 regularization, we are able to achieve accurate and stable results. As mentioned previously, the fourth order Runge-Kutta time stepping method was used for both the Fourier and Chebyshev cases.

Figure 3 compares the approximation to (42) using VV (23) with and without ℓ_1 regularization, for which we chose PA transform order $m = 2$ and regularization parameter $\lambda = .12$. Due to the numerical dissipation present in (23), the ℓ_1 enhancement does not appear to significantly improve the results.

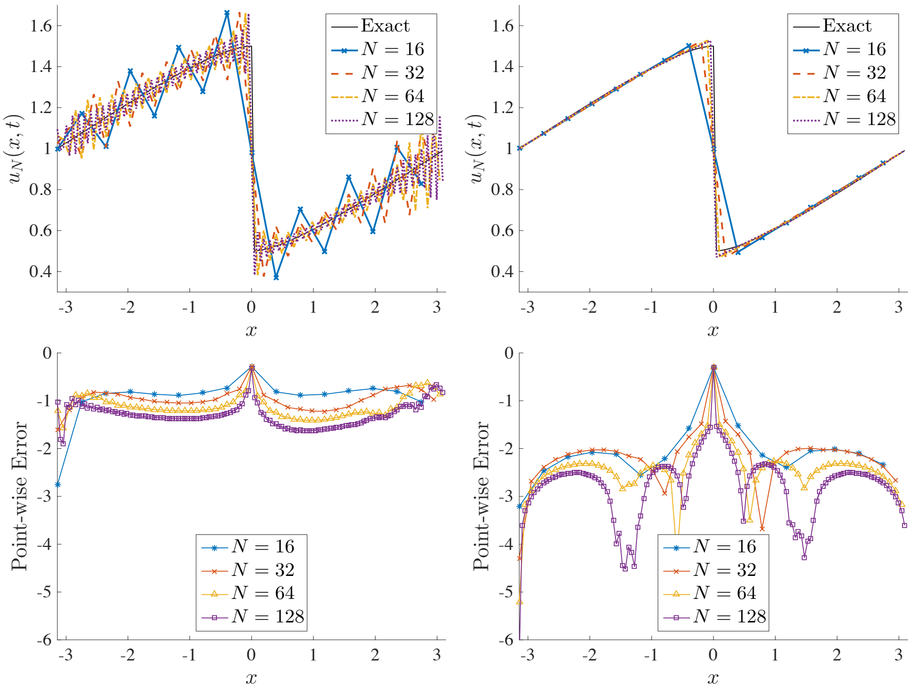

Fig. 2: Approximation of the solution to (42) for $N = 16, 32, 64$ and 128 with $\Delta t = \frac{\Delta x}{2}$. (top-left) PS (top-right) ℓ_1 enhanced PS method. Here we used the PA transform with $m = 2$ and $\lambda = .22$. Point-wise errors given in logarithmic scale for (bottom-left) the PS and (bottom-right) the ℓ_1 enhanced PS methods.

The solutions to (42) using the pseudo-spectral Fourier method with a tenth order exponential filter (EF), [16], with and without ℓ_1 enhancement are shown in Figure 4. Here we used the PA transform with $m = 2$ and $\lambda = .22$ in (16). As expected, filtering improves the accuracy away from the jump discontinuities, but the ℓ_1 enhancement dramatically reduces the oscillations and overshoot.

Note that no post-processing was applied to any of the ℓ_1 enhancement approximations. It has been shown that spectral reprojection, [15], improves the accuracy in smooth regions of the spectral viscosity method, [32]. However, this requires a priori knowledge of jump discontinuity locations. While the task is manageable in one dimension, it becomes increasingly difficult in multi-dimensions. In future investigations we will use the methods proposed in [3, 34] to post-process the ℓ_1 enhanced solutions. Indeed, Figures 1-4 demonstrate that in all cases the ℓ_1 enhancement improves resolution and reduces oscillations near shocks. Although the enhancement does not have significant impact on the vanishing viscosity results, it is interesting to note that using ℓ_1 enhancement *directly on the pseudo-spectral method without filtering* yields the best approximation. This is also the most cost efficient choice, and it does not require additional derivative approximations or additional parameter inputs, such as amount of viscosity.

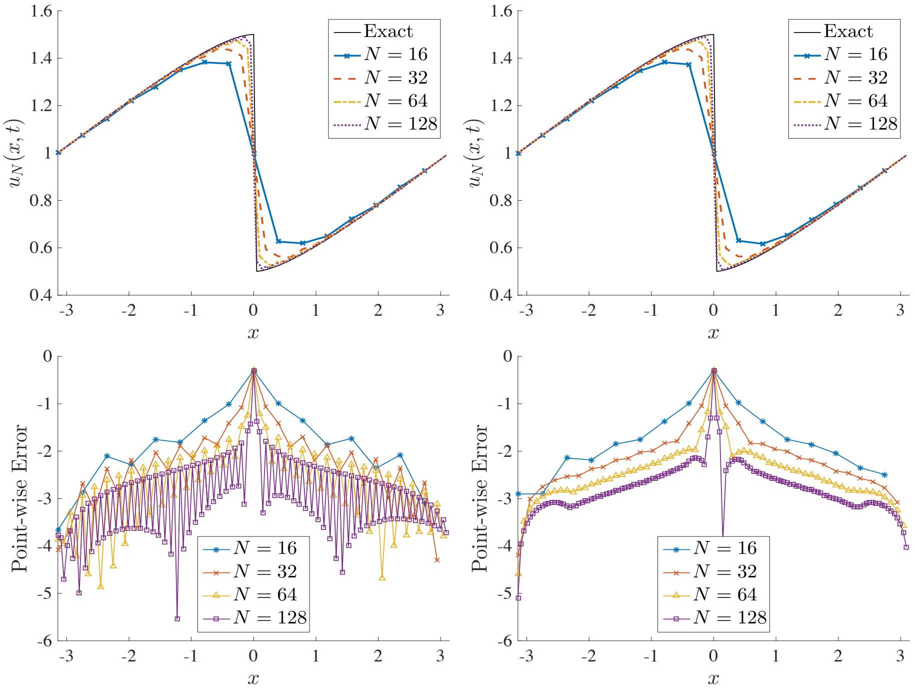

Fig. 3: Approximation of the solution to (42) for $N = 16, 32, 64$ and 128 with $\Delta t = \frac{\Delta x}{2}$. (top-left) VV (top-right) VV with ℓ_1 regularization. Here we used the PA transform with $m = 2$ and $\lambda = .12$. Point-wise errors given in logarithmic scale for (bottom-left) VV and (bottom-right) VV with ℓ_1 regularization.

Figure 5 shows the results of our stability analysis experiment. We consider four different numerical solvers with and without the ℓ_1 enhancement for a grid size of $N = 128$. For each solution, for various choices of α in (41), we calculate the error at location $x = -0.2454$, which is four grid points behind the shock at time $t = \pi$. Observe that the ℓ_1 enhancement yields greater accuracy for larger values of Δt than the standard solvers. Figure 5 also displays the final ℓ_1 enhanced solutions for $\alpha = 2$, so to show that accuracy is maintained along with stability when using a high order method in the fidelity term of (16).

4.2 Euler's equations

The Riemann problem for the one dimensional Euler equations is given as

$$\frac{\partial}{\partial t}\begin{pmatrix}\rho\\ \rho q\\ E\end{pmatrix} + \frac{\partial}{\partial x}\begin{pmatrix}\rho q\\ \rho q^2 + P\\ (E+P)q\end{pmatrix} = 0, \tag{43}$$

where $\rho : (a,b) \to \mathbb{R}$ is density, $q : (a,b) \to \mathbb{R}$ is velocity and $E : (a,b) \to \mathbb{R}$ is the total energy. The pressure $P : (a,b) \to \mathbb{R}$ is related to the conserved quantities

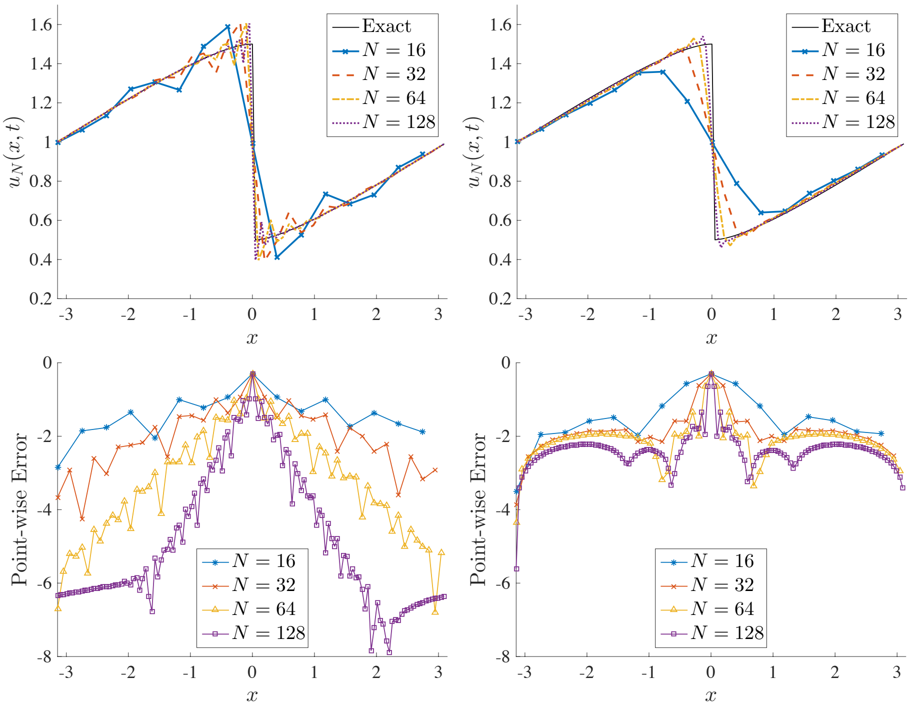

Fig. 4: Approximation of the solution to (42) for $N = 16, 32, 64$ and 128 with $\Delta t = \frac{\Delta x}{2}$. (top-left) EF (top-right) EF with ℓ_1 regularization. Here we used the PA transform with $m = 2$ and $\lambda = .22$. Point-wise errors given in logarithmic scale for (bottom-left) EF and (bottom-right) EF with ℓ_1 regularization.

through the equation of state:

$$P = (\gamma - 1)\left(E + \frac{\rho q^2}{2}\right),$$

with $\gamma = 1.4$ defined as the ratio of specific heat constants. The set of initial conditions we consider describe Sod's shock tube problem, [8], and are given by

$$\begin{aligned}&\rho_0(x) = 1, \quad q_0(x) = 0, \quad P_0(x) = 1, \text{ when } x \in [-1, 0)\\ &\rho_0(x) = 0.125, \quad q_0(x) = 0, \quad P_0(x) = 0.1, \text{ when } x \in [0, 1].\end{aligned} \tag{44}$$

We demonstrate the result of approximating the solution to (43) with (44) using the two-step Lax-Wendroff method (17) with and without regularization. The stability condition for this numerical scheme is

$$\max_i\{|\zeta_i|\}\frac{\Delta t}{\Delta x} \leq 1,$$

where ζ_i, $i = 1, 2, 3$ are the eigenvalues of the Jacobian matrix $\partial f/\partial u$, regarded as the propagation speeds of the corresponding characteristic waves. It can be

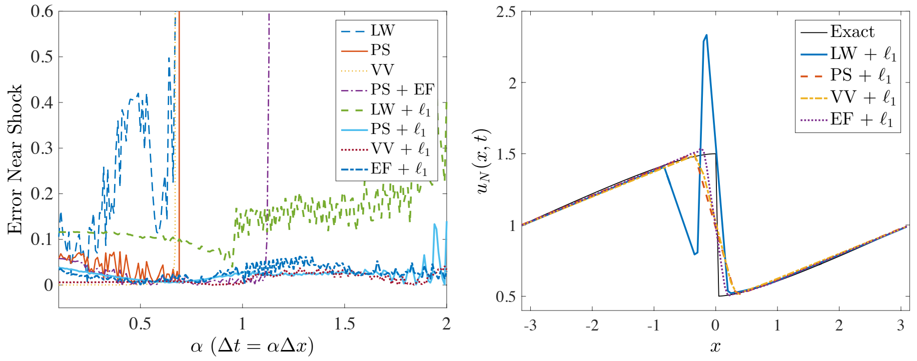

Fig. 5: (left) Stability analysis results. (right) The ℓ_1 enhanced solutions for a value of $\alpha = 2$ at final time $t = \pi$ for $N = 128$.

shown [19] that for the Euler system of equations, the stability condition leads to an adaptive time step given by

$$\Delta t = \text{cfl}\frac{\Delta x}{|q| + a}, \quad \text{cfl} < 1 \tag{45}$$

where $a = \sqrt{\gamma P/\rho}$ is the local speed of sound. We chose cfl $= .8$ for our numerical experiments.

Figure 6 compares solutions using the LW and ℓ_1 enhanced LW methods for density ρ, velocity q and pressure P when $N = 256$. While it is evident that the ℓ_1 enhancement reduces the size of the overshoots and oscillations that occur near discontinuities, the LW method appears to be too dissipative for the ℓ_1 enhancement to be very effective.

Figure 7 displays the approximation results using the filtered (mapped) Chebyshev method (30). The approximation results using the ℓ_1 enhanced Chebyshev method (32) are also displayed. In the non-regularized version, a 16th order exponential filter is implemented to the solution after each time step to ensure stability. No additional filtering is needed in the ℓ_1 enhancement case. Time stepping was implemented using fourth order Runge Kutta, with the time step Δt chosen according to the stability requirements in [18]. We approximate the solution at resolutions $N = 64, 128, 256$ and 512. In each case, we use PA transform order $m = 2$ and regularization parameter $\mu = 100$. The parameter β varies with N, with $\beta = .75$ for $N = 64, 256$, $\beta = 1.25$ for $N = 128$, and $\beta = .5$ for $N = 512$. Although these values for β represent the "best" results, in general the choice of β did not greatly affect the quality of the results. Future investigations will consider parameter optimization To obtain the results in Figure 7, Algorithm 1 requires 5 outer and 5 inner iterations. Hence there is a maximum of 25 iterations per time step. The efficiency of Algorithm 1 ensures that no additional significant computational time is required. Figure 8 displays the point-wise errors associated with estimating the final density for various resolutions with and without ℓ_1 regularization. As noted previously, due to the low order of accuracy, ℓ_1 enhancement does not significantly improve the results in the LW case. However, the ℓ_1 enhancement does dramatically improve the resolution properties near the discontinuities in the

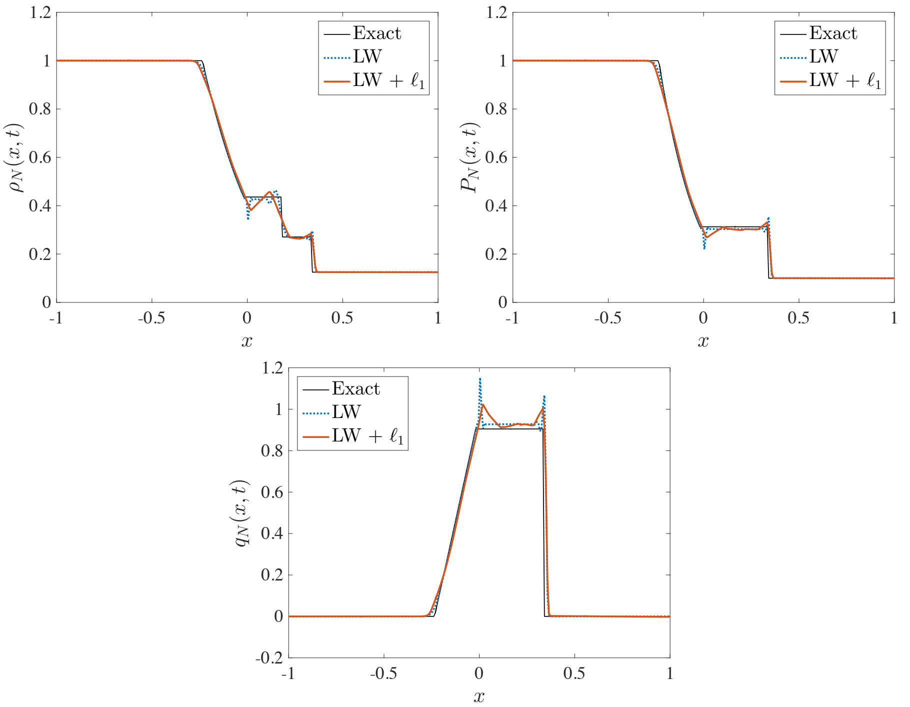

Fig. 6: (top-left) Density (top-right) Pressure (bottom) Velocity. Here, for the ℓ_1 enhancement, we used the PA transform with $m = 2$ and $\lambda = .35$.

Chebyshev case. Future investigations will include studies on post-processing these solutions.

4.3 Image denoising – A two-dimensional example

Let $\Omega \subset \mathbb{R}^2$ denote an open bounded set with a Lipschitz (sufficiently regular) boundary. Consider an image $u : \Omega \to \mathbb{R}^2$ corrupted with multiplicative noise $\nu : \Omega \to \mathbb{R}^2$ such that

$$g = u\nu \tag{46}$$

where $g \in L^2(\Omega)$. We assume that ν follows a Gamma distribution. The model in (46) often is used to represent speckle noise, which occurs in any form of coherent imaging where the objects being illuminated have surface features that are rough on the microscopic scale of the illuminating wavelength, and it is the result of the constructive and destructive summation of many random phasors being reflected from a single resolution cell [14]. Many filtering techniques have been explored to reduce the effects speckle. TV regularization has also been used with some success, [5].

Variational techniques can also be used to reduce speckle. For example, in [6], maximum a-posteriori probability (MAP) estimation theory is used in generating

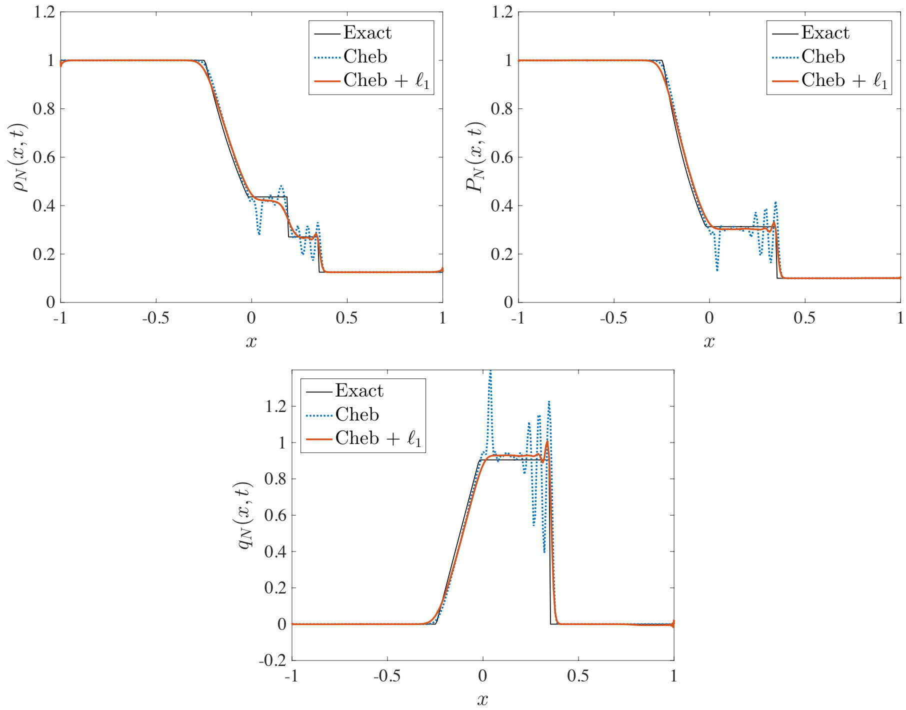

Fig. 7: (top-left) Density (top-right) Pressure (bottom) Velocity. For the ℓ_1 enhancement we used the PA transform with $m = 2$.

the model

$$\min_{u\in\Omega}\int_{\Omega}\left\{|\nabla u|+\eta\left(\log u+\frac{g}{u}\right)\right\}. \tag{47}$$

Here $\eta > 0$ is the inherent model parameter. We will refer to (47) the *AA model*. As is classically done in image denoising, the solution to (47) is computed by embedding the integral equation into a dynamical system using the Euler-Lagrange equation, which is driven to a steady state, [27]. The Euler-Lagrange equation is a second-order PDE with solutions that are functions for which a given functional is stationary. Because a differentiable function is stationary at local minima and maxima, this is a useful tool for optimization. The time dependent Euler-Lagrange equation associated with solving the AA model is

$$\begin{cases}\frac{\partial u}{\partial t}=\frac{1}{\eta}\mathrm{div}\left(\frac{\nabla u}{\sqrt{\epsilon^2+|\nabla u|^2}}\right)+\frac{g-u}{u^2} & \text{in} \quad \Omega\\ \frac{\partial u}{\partial \nu}=0 & \text{on} \quad \partial\Omega\end{cases}, \tag{48}$$

with given initial conditions. To remove the possible singularity when $|\nabla u| = 0$, ($u > 0$ so $u^2 \neq 0$) the model has been relaxed with $\epsilon > 0$, where we define

$$\mathrm{div}\left(\frac{\nabla u}{\sqrt{\epsilon^2+|\nabla u|^2}}\right):=\frac{\partial}{\partial x}\left(\frac{u_x}{\sqrt{\epsilon^2+u_x^2+u_y^2}}\right)+\frac{\partial}{\partial y}\left(\frac{u_y}{\sqrt{\epsilon^2+u_x^2+u_y^2}}\right).$$

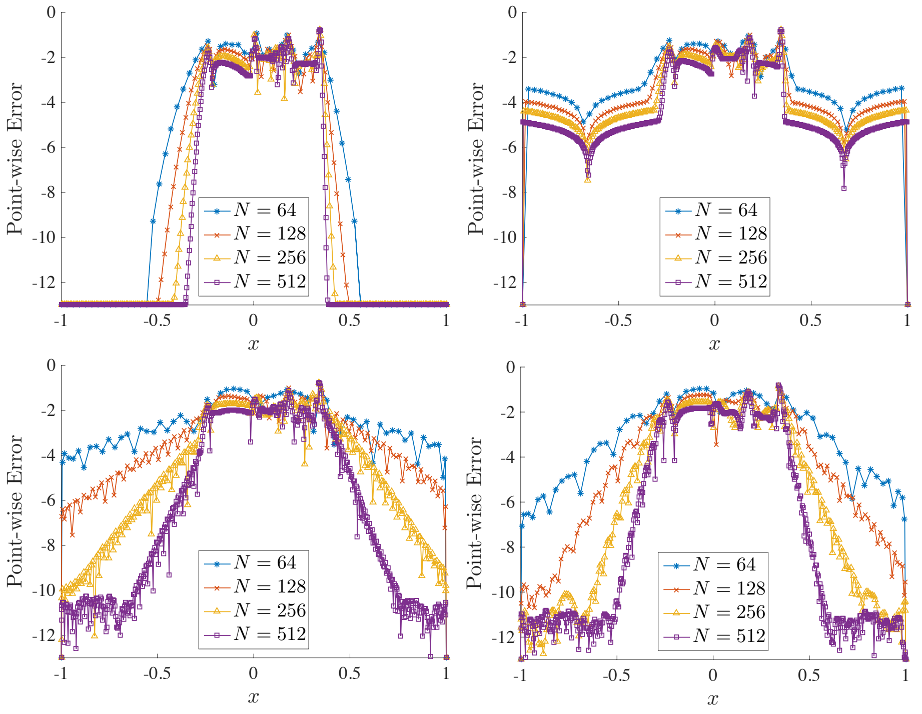

Fig. 8: Pointwise error in approximating the solution to Sod's shock tube problem using the (top) LW method (bottom) Chebyshev method, (left) without regularization and (right) with the ℓ_1 enhancement.

As time increases, the energy in (47) will decrease leading to a denoised image at steady state.

To discretize the solution to (48), for $0 \leq i, j \leq M$, let

$$x_i = i\Delta x, \quad y_j = j\Delta x,$$

where Δx is the pixel size. A forward Euler time stepping scheme was proposed to solve (48) and is given by [6, 27]

$$\begin{aligned} \frac{u_{i,j}^{n+1} - u_{i,j}^n}{\Delta t} = \frac{u_{i,j}^n - g_{i,j}^n}{\left(u_{i,j}^n\right)^2} &+ \frac{c_1}{2\eta\Delta x^2}\left(u_{i+1,j}^n - u_{i,j}^{n+1}\right) - \frac{c_2}{2\eta\Delta x^2}\left(u_{i,j}^{n+1} - u_{i-1,j}^n\right) \\ &+ \frac{c_3}{2\eta\Delta x^2}\left(u_{i,j+1}^n - u_{i,j}^{n+1}\right) - \frac{c_4}{2\eta\Delta x^2}\left(u_{i,j}^{n+1} - u_{i,j-1}^n\right), \end{aligned} \tag{49}$$

where

$$
\begin{aligned}
c_1 &= \frac{1}{\sqrt{\epsilon^2 + \left(\frac{u^n_{i+1,j} - u^n_{i,j}}{\Delta x}\right)^2 + \left(\frac{u^n_{i,j+1} - u^n_{i,j}}{\Delta x}\right)^2}},\\
c_2 &= \frac{1}{\sqrt{\epsilon^2 + \left(\frac{u^n_{i,j} - u^n_{i-1,j}}{\Delta x}\right)^2 + \left(\frac{u^n_{i-1,j+1} - u^n_{i-1,j}}{\Delta x}\right)^2}},\\
c_3 &= \frac{1}{\sqrt{\epsilon^2 + \left(\frac{u^n_{i+1,j} - u^n_{i,j}}{\Delta x}\right)^2 + \left(\frac{u^n_{i,j+1} - u^n_{i,j}}{\Delta x}\right)^2}},\\
c_4 &= \frac{1}{\sqrt{\epsilon^2 + \left(\frac{u^n_{i+1,j-1} - u^n_{i,j-1}}{\Delta x}\right)^2 + \left(\frac{u^n_{i,j} - u^n_{i,j-1}}{\Delta x}\right)^2}}.
\end{aligned}
$$

for interior grid points $1 \leq i, j, \leq M-1$ and time step Δt. The boundary conditions imposed are

$$
\begin{aligned}
&u_{0,j} = u_{1,j}, \quad u_{M,j} = u_{M-1,j}, \quad u_{i,0} = u_{i,1}, \quad u_{i,M} = u_{i,M-1},\\
&u_{0,0} = u_{1,1}, \quad u_{0,M} = u_{1,M-1}, \quad u_{M,0} = u_{M-1,1} \text{ and } u_{M,M} = u_{M-1,M-1}.
\end{aligned}
$$

The PDE is advanced in time for $n = 0, ..., N_t$. We note that using higher order temporal and spatial derivatives might yield higher accuracy, especially when using the ℓ_1 regularization enhancement. We will explore this idea in future investigations.

To reduce speckle and preserve the edges present in images, we augment the Euler-Lagrange PDE associated with the AA model with ℓ_1 regularization using the PA transform. To accomplish this, we first solve (49) for $u^{n+1}_{i,j}$ and define $B^n_{i,j}$ for $1 \leq i, j \leq M-1$ to be the right hand side of of the resulting expression

$$
\begin{aligned}
B^n_{i,j} := \frac{1}{A}\Big[u^n_{i,j} + \Delta t \frac{u^n_{i,j} - f_{i,j}}{\left(u^n_{i,j}\right)^2} &+ \frac{c_1 \Delta t}{2\eta \Delta x^2} u^n_{i,j-1} + \frac{c_2 \Delta t}{2\eta \Delta x^2} u^n_{i-1,j}\\
&+ \frac{c_3 \Delta t}{2\eta \Delta x^2} u^n_{i,j+1} + \frac{c_4 \Delta t}{2\eta \Delta x^2} u^n_{i,j-1}\Big],
\end{aligned}
\tag{50}
$$

where

$$
A = 1 + \frac{\Delta t}{2\eta \Delta x^2}\left(c_1 + c_2 + c_3 + c_4\right).
$$

Analogous to (15), we write

$$
U^n := \begin{bmatrix} u^n_{1,1} & u^n_{1,2} & \cdots & u^n_{1,M-1} \\ u^n_{2,1} & u^n_{2,2} & \cdots & u^n_{2,M-1} \\ \vdots & \vdots & \ddots & \vdots \\ u^n_{M-1,1} & u^n_{M-1,2} & \cdots & u^n_{M-1,M-1} \end{bmatrix}, \quad b^n := \begin{bmatrix} B^n_{1,1} & B^n_{1,2} & \cdots & B^n_{1,M-1} \\ B^n_{2,1} & B^n_{2,2} & \cdots & B^n_{2,M-1} \\ \vdots & \vdots & \ddots & \vdots \\ B^n_{M-1,1} & B^n_{M-1,2} & \cdots & B^n_{M-1,M-1} \end{bmatrix}.
$$

These matrices are embedded into (16) to determine the solution at time t_{n+1} at the interior grid points.

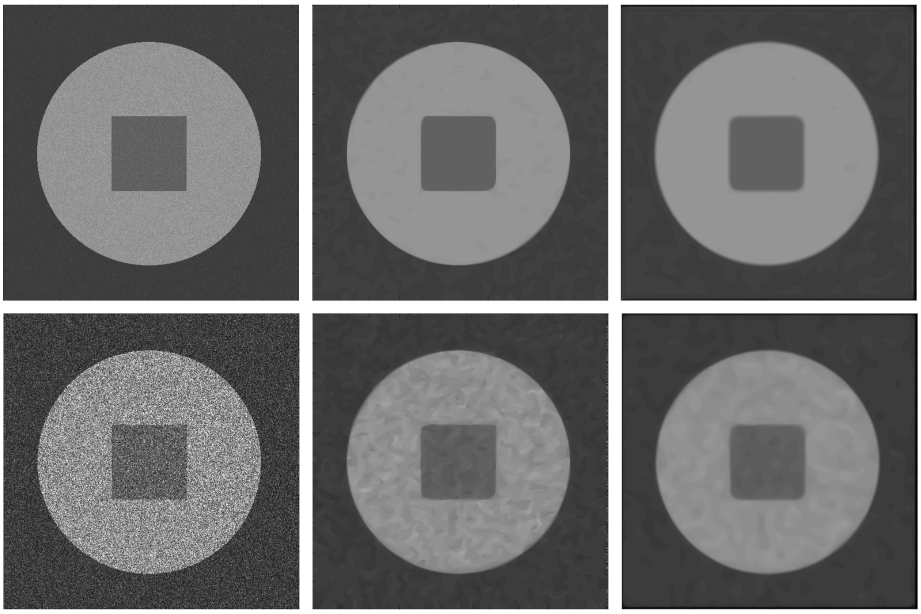

Fig. 9: (left) Noisy 512×512 pixel images corrupted with multiplicative speckle with parameter (top) $L = 200$ and (bottom) $L = 5$. (middle) Images that have been despeckled using the AA model as in Equation (49) with $\eta = .2$, $\Delta t = .1$ and $\Delta x \approx .2$. (right) Images denoised using the ℓ_1 enhanced AA method with parameters $m = 4$, $\mu = 10$ and $\beta = 40$.

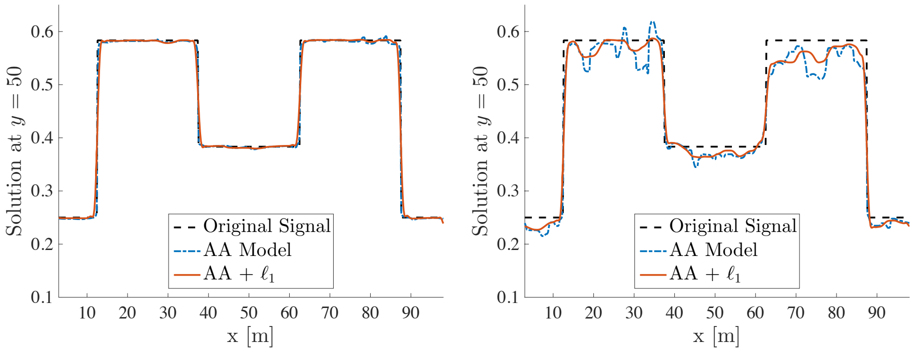

Fig. 10: Cross section of results in Figure 9 at location $y = 50$ for noise modeled with (left) $L = 200$ and (right) $L = 5$.

Consider the image $u : [-50, 50] \times [-50, 50] \to \mathbb{R}$ defined as

$$u(x, y) = \begin{cases} \frac{3}{5} & (x - 50)^2 + (y - 50)^2 \leq r^2 \\ \frac{2}{5} & r \leq x, y \leq r + 25 \\ \frac{1}{4} & \text{otherwise,} \end{cases} \tag{51}$$

where $r = 37.5$. The scene is discretized using $N = 512$ equally spaced grid points in the x and y directions as

$$x_i = -50 + i\Delta x, \quad y_j = -50 + j\Delta x, \quad i, j = 0, ..., N-1, \quad \Delta x = \frac{100}{N}.$$

The multiplicative noise in (46) is modeled as gamma distributed and is given by

$$\nu \sim \Gamma(L, 1/L), \tag{52}$$

where $L \geq 1$ represents the number of independent noisy signals (or looks) available. Figure 9 displays the noisy image for $L = 200$ ($SNR \approx 5.05$dB) (top-left) and $L = 5$ ($SNR \approx .13$dB) (bottom-left). As is evident in Figure 9(top-middle), when the SNR is large enough, the AA method is sufficient in despeckling the piecewise constant image. However, as the noise level is increased, the low order TV is insufficient as it attempts to generate piecewise constants from the noisy data (Figure 9 (bottom-middle)). On the other hand, although the higher order PA transform is not necessary when the SNR is high (Figure 9 (top-right)), as seen in Figure 9 (bottom-right), it is better able to recognize the smooth regions between the internal boundaries of the image when SNR is low, resulting in a more accurate recovery. We applied the ℓ_1 enhancement using the PA transform with $m = 4$, $\mu = 10$ and $\beta = 40$. Cross sectional reconstructions at $y = 50$ are provided in Figure 10.

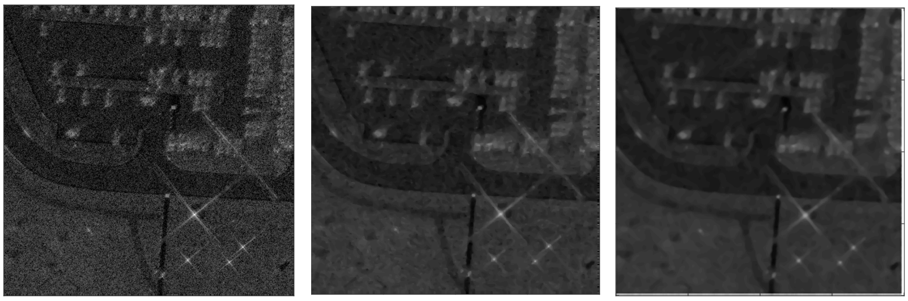

Fig. 11: Noisy Gotcha parking lot image (left). Data processed using the AA method (center) and corresponding ℓ_1 enhancement with $m = 4$, $\mu = 100$, and $\beta = 5$ (right).

Table 1: Gotcha Speckle Reduction Metrics

	ENL	σ_{speck}	Bias
Noisy Image	-	5.6291	-
AA Model	1752.7	1.3544	0.0072
AA + ℓ_1	3517.5	0.9615	0.0159

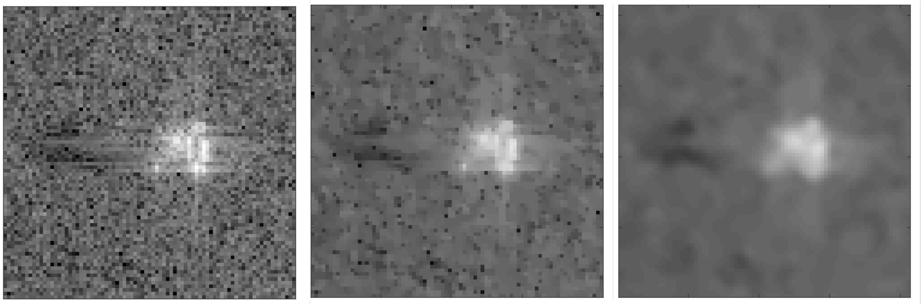

Fig. 12: Noisy MSTAR image (left). Data processed using the AA method (center) and corresponding ℓ_1 enhancement with $m = 4$, $\mu = 100$, and $\beta = 5$ (right).

Finally, Figures 11 and 12 depict two scenes provided in [1] and [2] respectively.[6] In both cases, we compare the results using the AA method and its ℓ_1 enhanced version with PA transform parameters $m = 4$, $\mu = 100$ and $\beta = 5$. It is evident that the speckle is reduced in each scene.

To further validate our results, we calculate three noise suppression metrics. The *equivalent number of looks* (ENL) is a metric used for evaluating the level of smoothing in homogeneous areas. In particular, the scene variation should be negligible with respect to speckle noise fluctuations. The ENL calculates the number of multiple looks necessary to create an image with the same level of reduced speckle. A good despeckling technique would yield a high ENL number. In reference to (46), if u is the despeckled image, then in the homogeneous areas h we have

$$\text{ENL} := \frac{E_h[u]^2}{\text{var}_h[u]}. \tag{53}$$

The *speckle standard deviation* given by

$$\sigma_{\text{speck}} := \sqrt{\text{var}_h[u]} \tag{54}$$

measures the average variation in speckle amplitude throughout homogeneous regions h. A small speckle standard deviation is indicative of desirable smoothness and speckle reduction. To calculate (53) and (54), we must select a region in each image where only noise is expected to be present (no signal), [5]. For the MSTAR image, we chose the bottom twenty rows of the image matrix to correspond to a homogeneous region, and in the Gotcha image, the image data corresponds to a patch within the parking lot.

Finally, the *reconstruction bias*, B, measures the level of bias in the estimated image. One way to calculate the bias is to measure the relative expected value of the reconstructed image to the cluttered image g, and is given by

$$B := E\left[\frac{g-u}{u}\right]. \tag{55}$$

A bias measurement of zero would represent a completely unbiased estimate, while a value of $B > .2$ would indicate a high level of bias, [33].

[6] More information about these images can be found at [1], [2], and [30].

Table 2: MSTAR Speckle Reduction Metrics

	ENL	σ_{speck}	Bias
Noisy Image	-	5.6843	-
AA Model	203.3	2.7396	0.0026
AA + ℓ_1	933.67	1.2820	0.0777

Tables 2 and 1 display the results of calculating these metrics for the MSTAR and Gotcha parking lot image respectively. It is evident that our proposed ℓ_1 enhanced AA method reduces the noise beyond the AA method while maintaining acceptable image bias in the reconstructions.

5 Concluding Remarks

In this paper we introduced a method for solving non-linear partial differential equations using ℓ_1 regularization, and specifically, using the polynomial annihilation (PA) transform operator in the ℓ_1 term. Our results demonstrate that it is possible to efficiently implement a method that is accurate and resolves shock discontinuities. The method does not require advance knowledge of the shock locations.

Our new method is particularly useful for numerically solving hyperbolic partial differential equations that develop shocks or discontinuities. The ℓ_1 regularization enhanced method exploits the knowledge that there are a sparse number of singularities in the solution. Our method is made numerically efficient by employing the alternating direction of multipliers algorithm. We used our technique to approximate the solution to Burgers' equation with a smooth initial condition, Euler's equations with initial conditions describing Sod's shock tube, and a two-dimensional PDE often employed for denoising imagery corrupted with multiplicative gamma noise. In all cases, our new method showed improved accuracy near the shock locations. These results are obtained without post-processing or the need for shock tracking. However, post-processing may further improve our results. In addition, our method remains stable for larger time steps than those typically used by conventional solvers. In future investigations we will study how to optimize the parameters of our method as well as develop a more rigorous understanding of stability conditions.

Other areas to explore include using different PDE solvers, such as the discontinuous Galerkin method, in the data fidelity term. We anticipate that this will help localize the effect of smoothing to only those cells that contain discontinuities. Also beneficial could be the inclusion of time dependent regularization to enforce the knowledge that the solutions do not vary significantly from one time step to the next. Both topics will be explored in future studies.

References

1. GOTCHA volumetric SAR data set overview. https://www.sdms.afrl.af.mil/index.php?collection=gotcha. Accessed: 2016-19-08

2. MSTAR overview. `https://www.sdms.afrl.af.mil/index.php?collection=mstar`. Accessed: 2016-19-08
3. Archibald, R., Gelb, A., Platte, R.B.: Image reconstruction from undersampled Fourier data using the polynomial annihilation transform. Journal of Scientific Computing **67**(2), 432–452 (2016)
4. Archibald, R., Gelb, A., Yoon, J.: Polynomial fitting for edge detection in irregularly sampled signals and images. SIAM Journal on Numerical Analysis **43**(1), 259–279 (2005)
5. Argenti, F., Lapini, A., Bianchi, T., Alparone, L.: A tutorial on speckle reduction in synthetic aperture radar images. IEEE Geoscience and remote sensing magazine **1**(3), 6–35 (2013)
6. Aubert, G., Aujol, J.F.: A variational approach to removing multiplicative noise. SIAM Journal on Applied Mathematics **68**(4), 925–946 (2008)
7. Canuto, C., Hussaini, M.Y., Quarteroni, A.M., Thomas Jr, A., et al.: Spectral methods in fluid dynamics. Springer Science & Business Media (2012)
8. Danaila, I., Joly, P., Kaber, S.M., Postel, M.: An introduction to scientific computing: Twelve computational projects solved with MATLAB. Springer Science & Business Media (2007)
9. 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 Journal on Scientific Computing **38**(2), A691–A711 (2016)
10. Driscoll, T.A., Hale, N., Trefethen, L.N.: Chebfun guide (2014)
11. Durand, S., Froment, J.: Reconstruction of wavelet coefficients using total variation minimization. SIAM Journal on Scientific computing **24**(5), 1754–1767 (2003)
12. Glowinski, R., Le Tallec, P.: Augmented Lagrangian and operator-splitting methods in nonlinear mechanics. SIAM (1989)
13. Goldstein, T., Osher, S.: The split bregman method for l1-regularized problems. SIAM Journal on Imaging Sciences **2**(2), 323–343 (2009)
14. Goodman, J.W.: Speckle phenomena in optics: theory and applications. Roberts and Company Publishers (2007)
15. Gottlieb, D., Shu, C.W.: On the Gibbs phenomenon and its resolution. SIAM Review **39**(4), 644–668 (1997)
16. Hesthaven, J.S., Gottlieb, S., Gottlieb, D.: Spectral methods for time-dependent problems, vol. 21. Cambridge University Press (2007)
17. Hou, T.Y., Li, Q., Schaeffer, H.: Sparse + low-energy decomposition for viscous conservation laws. Journal of Computational Physics **288**, 150–166 (2015)
18. Kosloff, D., Tal-Ezer, H.: Modified chebyshev pseudospectral method with $O(N^{-1})$ time step restriction. Journal of Computational Physics **104**(2), 457–469 (1993)
19. LeVeque, R.J.: Numerical methods for conservation laws. Springer Science & Business Media (1992)
20. LeVeque, R.J.: Finite volume methods for hyperbolic problems, vol. 31. Cambridge University Press (2002)
21. LeVeque, R.J.: Finite difference methods for ordinary and partial differential equations: steady-state and time-dependent problems. SIAM (2007)
22. Li, C.: An efficient algorithm for total variation regularization with applications to the single pixel camera and compressive sensing. Ph.D. thesis, Rice University (2009)
23. Li, C., Yin, W., Jiang Hongand Zhang, Y.: An efficient augmented Lagrangian method with applications to total variation minimization. Comput. Optim. Appl. **56**(3), 507–530 (2013)
24. Lustig, M., Donoho, D., Pauly, J.M.: Sparse mri: The application of compressed sensing for rapid mr imaging. Magnetic Resonance in Medicine **58**(6), 1182–1195 (2007)
25. Moulin, P.: A wavelet regularization method for diffuse radar-target imaging and speckle-noise reduction. In: Wavelet Theory and Application, pp. 123–134. Springer (1993)
26. Osher, S., Burger, M., Goldfarb, D., Xu, J., Yin, W.: An iterative regularization method for total variation-based image restoration. Multiscale Modeling and Simulation **4**(2), 460–489 (2005)
27. Rudin, L.I., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. Physica D: Nonlinear Phenomena **60**(1-4), 259–268 (1992)
28. Sanders, T.: Matlab imaging algorithms: Image reconstruction, restoration, and alignment, with a focus in tomography. `http://www.toby-sanders.com/software`, `https://doi.org/10.13140/RG.2.2.33492.60801`. Accessed: 2016-19-08

29. Sanders, T., Gelb, A., Platte, R.B.: Composite SAR imaging using sequential joint sparsity. Journal of Computational Physics **338**, 357–370 (2017)
30. Scarnati, T., Zelnio, E., Paulson, C.: Exploiting the sparsity of edge information in synthetic aperture radar imagery for speckle reduction. pp. 102,010C0–102,010C–13 (2017). DOI 10.117/12.2267790. URL http://dx.doi.org/10.1117/12.2267790
31. Schaeffer, H., Caflisch, R., Hauck, C.D., Osher, S.: Sparse dynamics for partial differential equations. Proceedings of the National Academy of Sciences **110**(17), 6634–6639 (2013)
32. Shu, C.W., Wong, P.S.: A note on the accuracy of spectral method applied to nonlinear conservation laws. Journal of scientific computing **10**(3), 357–369 (1995)
33. Solbo, S., Eltoft, T.: A stationary wavelet-domain wiener filter for correlated speckle. IEEE Transactions on Geoscience and Remote Sensing **46**(4), 1219–1230 (2008)
34. Stefan, W., Renaut, R.A., Gelb, A.: Improved total variation-type regularization using higher order edge detectors. SIAM Journal on Imaging Sciences **3**(2), 232–251 (2010)
35. Tadmor, E.: Convergence of spectral methods for nonlinear conservation laws. SIAM Journal on Numerical Analysis **26**(1), 30–44 (1989)
36. Tadmor, E.: Shock capturing by the spectral viscosity method. Computer Methods in Applied Mechanics and Engineering **80**(1-3), 197–208 (1990)
37. Tadmor, E.: Super viscosity and spectral approximations of nonlinear conservation laws. In: M. Baines, K. Morton (eds.) Proceedings of the 1992 Conference on Numerical Methods for Fluid Dynamics, pp. 69–82 (1993)
38. Tadmor, E., Waagan, K.: Adaptive spectral viscosity for hyperbolic conservation laws. SIAM Journal on Scientific Computing **34**(2), A993–A1009 (2012)
39. Vogel, C.R., Oman, M.E.: Iterative methods for total variation denoising. SIAM Journal on Scientific Computing **17**(1), 227–238 (1996)
40. Wasserman, G., Archibald, R., Gelb, A.: Image reconstruction from fourier data using sparsity of edges. Journal of Scientific Computing **65**(2), 533–552 (2015)