



Sparse + low-energy decomposition for viscous conservation laws



Thomas Y. Hou, Qin Li*, Hayden Schaeffer

ARTICLE INFO

Article history:

Received 7 October 2014

Accepted 13 February 2015

Available online 20 February 2015

Keywords:

Compressive sensing

Sparsity

Viscous conservation laws

Decomposition

ABSTRACT

For viscous conservation laws, solutions contain smooth but high-contrast features, which require the use of fine grids to properly resolve. On coarse grids, these high-contrast jumps resemble shocks rather than their true viscous profiles, which could lead to issues in the numerical approximation of their underlying dynamics. In many cases, the equations of motion emit traveling wave solutions which can be used to represent the viscous profiles analytically. The traveling wave solutions can be thought of as a lower dimensional representation of the motion, since they contain information from the evolution equation, but are constant along certain time–space curves. Using a parameterized basis involving the traveling waves, along with the sparse + low-energy decompositions found in imaging sciences, we propose an approximation to viscous conservation laws which separates the coarse smooth component from the sharp fine one. Our method provides an appropriate approximation to the solution on a coarse grid, thereby accurately under-resolving the viscous profile. This is similar to the philosophy of shock capturing methods, in the sense that we want to capture the viscous front without needing to resolve the profile. Theoretical results on the consistency of our method are shown in general. We provide several computational examples for convex and non-convex fluxes.

© 2015 Elsevier Inc. All rights reserved.

1. Introduction

Solutions to viscous conservation laws typically contain both smooth components, related to linear diffusion terms, and sharp-contrast features, which are generated by the nonlinearities. These equations commonly arise in physical models of evolution equations with viscosity, for example, in viscous conservation laws, fluid dynamics, combustion theory, and other reaction–diffusion systems. The difficulty in many of these equations is related to the largely different gradient scales which appear in the solutions, making it computationally inefficient to resolve the viscous profile and the smooth components simultaneously. Since the solutions are smooth, the application of shock-capturing schemes would lead to the wrong physical behavior of the solutions in time (see Fig. 1.1), therefore, one may wish to consider pseudo-spectral methods.

In general, the (pseudo)-spectral method is a popular choice for solving quasi-linear evolution equations [29,30,19,11,14]. One reason for their success has been related to their computational simplicity and analytic properties. However in practice, the standard spectral method is known to have issues when applied to hyperbolic systems [21,12,1], even in the linear case. These issues are related to the strong restrictions placed on the regularity of the solution, which cannot be guaranteed for hyperbolic systems. It is well known that a spectral method gives spectral accuracy if the solution is sufficiently smooth. More precisely, for a C^{r-1} function whose r th derivative is bounded in L^1 , the error in the spectral approximation will decay

* Corresponding author.

E-mail addresses: hou@cms.caltech.edu (T.Y. Hou), qinli@caltech.edu (Q. Li), schaeffer@caltech.edu (H. Schaeffer).

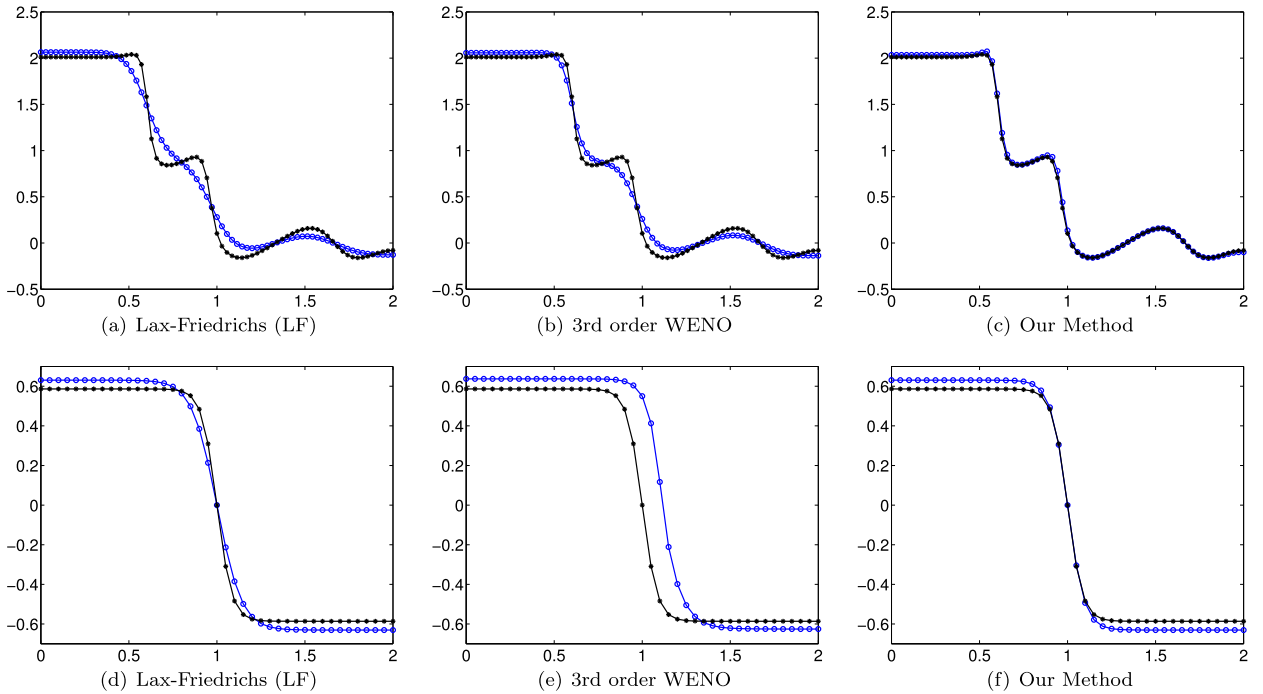


Fig. 1.1. Burgers' equation simulated with a high-resolution central differencing scheme (black) and various methods on a coarse grid (blue). Plots (a)–(c) are simulated with a 70 point grid (blue) and are taken at $t = 0.25$ with physical viscosity ($\gamma = 1 \times 10^{-2}$). The artificial viscosity present in the standard shock-capturing schemes causes over smoothing of the viscous profiles (as seen in plot (a)). For this problem, the 3rd order weight essentially non-oscillatory methods tend to have difficulties distinguishing shocks, viscous profiles, and smoother features. This leads to a flattening of the solution near the first viscous profile and some over-smoothing in other regions. For plots (d)–(f), the simulation is on a 40 point grid with $t = 25$ and $\gamma = 2.5 \times 10^{-2}$. Our method produces a stable long-time stationary profile without over-smoothing the profile. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

with rate $O(\frac{1}{N^r})$, where N is the number of modes used in the basis. However, the solution to hyperbolic conservation laws contains profiles with sharp transitions. In the extreme cases, for example the inviscid Burgers' equation with Riemann initial data (decreasing to the right), the solution will contain a shock that propagates with a constant speed. If one utilizes the standard spectral method, the error in the L^∞ norm would be $O(1)$ as $N \rightarrow \infty$ (this is referred to as the so-called Gibbs phenomena). Several methods have appeared in the literature to overcome this, which use variable viscosity to smooth the solution or post-processing to filter out the oscillations [20,28,30,31]. On the other hand, for these nonlinear hyperbolic equations, methods that employ local polynomial interpolation such as the essentially non-oscillatory (ENO) method [27] and the weighted ENO (WENO) method [17] are preferred.

In this work, we focus on the case where the solution is continuous, *i.e.* contains smooth shock profiles, and we develop the corresponding appropriate basis functions to optimally represent the features found in solutions. We assume that the solution lives in the function space $\mathcal{H} = \text{span}\{\psi_i\}$ spanned by the orthonormal basis ψ_i . The truncated space of order N is defined as:

$$\mathcal{H}_N := \text{span}\{\psi_i, i = 0, \dots, N\}.$$

For each function in \mathcal{H} , we look for its best approximation in \mathcal{H}_N through the subspace projection:

$$\mathcal{P}_N u = \sum_{i=0}^N \langle u, \psi_i \rangle_A \psi_i,$$

which is well-defined due to the orthogonality of the basis. To analyze the error in this approximation, one measures the distance between the solution u and its truncated version $\mathcal{P}_N u$, which is typically bounded by some norm of u :

$$\text{Error} = \|u - \mathcal{P}_N u\|_A \leq C(N) \|u\|_B.$$

The norm on A is usually induced through the metric $\langle \cdot, \cdot \rangle_A$, while the norm in B is problem dependent. For example, the classical spectral method (either using Fourier or orthogonal polynomials) bounds the L^2 error by an H^α norm of u :

$$\|u - \mathcal{P}_N u\|_{L^2} \lesssim \frac{1}{N^\alpha} \|u\|_{H^\alpha},$$

for some integer $\alpha \geq 0$. We can see here that the error is only as good as the value of u in an H^α norm. Therefore, even if the regularity of u is high outside of a small localized region in space (which is the case for viscous profiles), this type

of approximation will still be inaccurate. In such cases, the number of modes must increase so that $C(N)$ can balance the large value of $\|u\|_{H^\alpha}$. Thus in order to approximate u using fewer modes, the B -norm should have little dependence on derivatives of the solution. For bounded solutions u , the optimal error bound takes the form:

$$\text{Error} = \|u - \mathcal{P}_N u\|_{L^2} \leq C(N) \|u\|_\infty \lesssim C(N),$$

removing the need to bound higher-order derivatives.

Inspired by the recent work in imaging sciences for approximating complex data using sparse + low-rank decompositions [6,7,13,16,25], we propose an approximation that uses the sparse and low-energy (*i.e.* low-frequency or small gradient) behavior of the solution to optimally decompose it in a redundant basis, and thus achieve a better error bound. Although the use of sparse and/or data-driven basis in imaging and the related sciences are prevalent in the literature, their applications to dynamical systems and PDE are still limited. In [26,18] the evolutions of quasi-linear oscillatory PDE is shown to be well-captured in a sparse Fourier expansion via an L^1 minimization. In [23,24], problems in quantum chemistry are solved by constructing localized basis functions using an L^1 regularized energy. L^1 regularized methods are also used to efficiently represent the compactly supported behavior found in free boundary and obstacle problems, see [4,32]. The data-driven techniques found in [8–10], use the Karhunen–Loeve expansion (related to the SVD) to construct a basis which can compactly represent the solutions of stochastic PDE. Low-rank basis and libraries are also the foundation for the POD and DMD methods. By learning a low-rank basis, the POD/DMD method represents the evolution of a given dynamical system using a small number of modes [3,2,34,33].

In all of these methods, the dynamic data-driven basis representation provides advantages over the standard spectral methods. In particular, the decay requirement of the spectral method is equivalent to the number of terms needed to approximate a solution in the Fourier basis. Within the data science framework, these basis representations are optimal when a function is low-rank (or low-energy) in the prescribed basis. However, for various data sets, solutions are rarely low-rank against the Fourier basis, and thus the methods above choose a low-rank expansion with respect to a basis that better reflects the behavior of the solutions. Following this idea, we propose a sparse + low-energy decomposition in a basis that reflects the dynamics of the problem.

In this work, we construct a numerical approximation to scalar viscous conservation laws that can capture the viscous profiles on under-resolved grids. This is similar to the philosophy of shock-capturing schemes; however, we consider the case where the viscosity that appears in the PDE is physically meaningful and cannot be neglected. Our approach uses the idea of sparse + low-rank decomposition in order to decompose the solution into the sharp component (*i.e.* the viscous profile or traveling wave front) and a low-energy remainder. The low-energy component is smooth and can be well-approximated by methods such as central finite differences or spectral methods. This differs from the data-driven or information science-based methods that we discussed previously, since our redundant basis is given analytically. Also since the remainder is smooth, we can easily incorporate other numerical solvers. Therefore, our method complements many of the known algorithms for conservation laws.

The outline of this work is as follows. In Section 2, we detail our model for decomposing the solution and approximating the system of PDE. Analysis of the system of PDE is provided in Section 3. In Section 4, numerical experiments for viscous conservation laws are shown for both convex and non-convex fluxes. Concluding remarks are given in Section 5.

2. Model

In this section, we present our numerical method for solving the viscous conservation law:

$$\partial_t u + \partial_x f(u) = \gamma \partial_x^2 u, \quad (2.1)$$

where f is the flux term and $\gamma > 0$ is the viscosity. For a concrete example, we will focus on Burgers' equation, which has a quadratic flux term:

$$\partial_t u + \partial_x \left(\frac{u^2}{2} \right) = \gamma \partial_x^2 u. \quad (2.2)$$

Our method can be applied to other fluxes as well, which is explained in Remark 2.1. In the following sections, we derive the viscous profile for Burgers' equation, detail the proposed splitting method for the evolution equation, and explain the variational decomposition method.

2.1. Viscous profile

For hyperbolic conservation laws with C^∞ initial data, the solution will develop shocks in the limit of vanishing viscosity, and the H^α norm will grow in time. When the viscosity is small, the diffusion term $\partial_x^2 u$ must balance the flux term, creating a sharp but smooth profile. The viscous profile can be derived via a change of variables, η , which moves with speed $x_1'(t)$:

$$\eta = x - x_1(t).$$

This transforms Eq. (2.2) to:

$$\partial_t u + \left(u - \frac{d\eta}{dt}\right) \partial_\eta u = \gamma \partial_\eta^2 u.$$

If the profile is stationary with respect to the change of variables, then $\partial_t u(\eta, t) = 0$, yielding the following equation:

$$\left(u - \frac{d\eta}{dt}\right) \partial_\eta u = \gamma \partial_\eta^2 u.$$

Assuming that the profile approaches constant values away from the sharp transition, i.e. $u \rightarrow c_1$ as $\eta \rightarrow \infty$ and $u \rightarrow c_2$ as $\eta \rightarrow -\infty$, and moves with constant speed, the equation above has the following analytic solution:

$$u := \phi(\eta; c_1, c_2, x_0) = c_1 + \frac{c_2 - c_1}{1 + e^{\frac{(c_2 - c_1)\eta}{2\gamma}}}, \quad \text{with} \quad \eta = x - x_0 - \frac{1}{2}(c_1 + c_2)t. \tag{2.3}$$

For convenience, ϕ will be used to denote the profile as a function of the parameters. This traveling profile (also referred to as a traveling wave solution) is centered at x_0 and moves with constant speed computed by averaging the two endpoint values. Note that the “jump” along the smooth transition is positive, since $c_2 - c_1 > 0$.

2.2. Splitting into a system

Given arbitrary initial data, the evolution equation will generate sharp transitions in the solution, with the leading order behavior converging to the profile given by Eq. (2.3). Away from these sharp transitions, the solution is smooth and has smaller derivative. Using this known behavior, we treat each component separately, using the analytic solution to approximate sharp transitions and direct numerical simulations on the smooth (low-energy) component.

Let u be a solution to Eq. (2.2), then we decompose u into two components $u = v + w$, where v is the smooth (low-energy) component and w approximates the sharp (high-energy) component. Using this decomposition, Eq. (2.2) can be split into the following equivalent system:

$$\begin{cases} \partial_t v + (v + w)\partial_x v = \gamma \partial_x^2 v \\ \partial_t w + (v + w)\partial_x w = \gamma \partial_x^2 w. \end{cases} \tag{2.4}$$

Our assumption is that w contains the high-energy component, which we will model using the following space:

$$\tilde{B} = \text{span}\{\phi(x; c_1, c_2, x_0)\}.$$

In practice, we consider $w \in B$ that can be written as a finite sum:

$$w = \sum_{k=1}^K \phi(x; c_{1,k}, c_{2,k}, x_{0,k}), \tag{2.5}$$

and based on the condition that the centers of the sharp transitions are well-separated, i.e. $x_{0,k} - x_{0,k-1} \gg \gamma$, w can be further approximated by:

$$w \approx \sum_{k=1}^K \chi_k \phi(x; e_{1,k}, e_{2,k}, x_{0,k}), \tag{2.6}$$

where χ_k is characteristic function on $[\frac{x_{0,k-1} + x_{0,k}}{2}, \frac{x_{0,k} + x_{0,k+1}}{2}]$, with $x_{0,k} < x_{0,k+1}$, $x_{0,K+1} := \infty$ and $x_{0,0} := -\infty$. Eq. (2.6) is taken as our ansatz for w . The new parameters, e , are computed from c by the following relations:

$$e_{1,k} = \sum_{j>k} (c_{2,j} - c_{1,j}) + c_{1,k}, \quad e_{2,k} = e_{1,k} + c_{2,k} - c_{1,k}. \tag{2.7}$$

Parameter e is the cumulative jump height and preserves the values of the jumps from parameter c . The function w resembles a stair-case, since each profile is summed together and shifts the heights of the other profiles. Altogether, we call the collection of these functions B , which can be shown to be close to \tilde{B} by $\mathcal{O}(\exp(-\frac{1}{\gamma}))$.

Overall, each of the approximations relies on several conditions. These conditions break down when sharp transitions form in v or when the profiles in w interact. Therefore, we require that the decomposition be reinitialized when:

1. v is no longer smooth enough, i.e. $|\partial_x v| > C_T$;
2. the viscous profiles are no longer well separated, i.e. $x_{0,k} - x_{0,k-1} < \text{tol}$, for some tolerance, tol .

The algorithm requires two solvers: one to decompose the solution and a second to evolve the equations under our assumptions (see Sections 2.3 and 2.4, respectively, for more detail). An outline of the algorithm is shown in Algorithm 1.

Algorithm 1: Outline of our algorithm.

```

Data: initial data:  $u(x, 0)$ 
Result:  $u$  at stopping time  $T$ :  $u(x, T)$ 
initialization: decompose  $u(x, 0) = v + w$  such that  $v \in A$  and  $w \in B$ ;
while  $t < T$  do
  Evolution with time step  $\Delta t$ :
   $t \rightarrow t + \Delta t$ ,  $v^n \rightarrow v^{n+1}$ ,  $w^n \rightarrow w^{n+1}$ ;
  if  $|\partial_x v^{n+1}| > C_T$  or  $x_{0,k} - x_{0,k-1} \leq \text{tol}$  then
    Re-initialization: (Decompose)  $u^{n+1} = \tilde{v} + \tilde{w}$ , and set  $v^{n+1} \rightarrow \tilde{v}$ ,  $w^{n+1} \rightarrow \tilde{w}$ 
  else
  end
end

```

2.3. Variational decomposition model

In order to evolve the approximate system of equations, we must first decompose the solution u^n into the two components: $u^n = \tilde{v} + \tilde{w}$, where $\tilde{v} \in A$ and $\tilde{w} \in B$. For simplicity, we drop the tilde superscript on \tilde{v} and \tilde{w} in this section. The decomposition is generally non-unique; however, we will determine it using an optimization procedure to encourage a proper expansion. Based on our ansatz (Eq. (2.6)), w is represented by a linear combination of the viscous profiles $\tilde{\phi}_k$:

$$w = \sum_{k=1}^K \alpha_k \tilde{\phi}_k(x; p_k, x_k),$$

where p is a general set of parameters defining the profile and x_k are the centers of the profile. For simplicity, we define the new profiles $\tilde{\phi}_k$, which are the normalized (in L^∞) form of $\chi_k \phi_k$. Since the basis $\{\tilde{\phi}_k(x; p_k, x_k)\}_k$ is redundant (i.e. the number of terms exceeds the size of the grid), the expansion above is overdetermined. Also, depending on the number of parameters, this representation could be high dimensional. Therefore, we must simplify the problem further in order to more easily compute this expansion.

The basis is global in the sense that each term ϕ_k is non-zero throughout the domain of interest. Therefore, at any point $x \in \Omega$ the decomposition mixes information from both terms $v(x, t)$ and $w(x, t)$. However, $\partial_x \phi_k$ is concentrated around x_k , thus the decomposition in the gradient space: $\partial_x u = \partial_x v + \partial_x w$ better separates the functions v and w , since near x_k we expect $\partial_x u \approx \partial_x w$ and away from x_k we expect $\partial_x u \approx \partial_x v$. Thus, the terms are less coupled in the gradient domain.

One of our assumptions is that the number of sharp transitions is very small relative to the dimension and size of the problem. The natural way to measure this would be to count the number of non-zero α_k (commonly referred to as the l^0 "norm") to ensure that only a few basis functions are needed to represent w . Altogether, the following optimization problem could be used:

$$\begin{aligned}
 & \min_{\partial_x v, \partial_x w, p, x^c} \|\alpha\|_{l^0} \\
 & \text{s.t. } \partial_x u = \partial_x v + \partial_x w \text{ and} \\
 & w = \sum_k \alpha_k \tilde{\phi}_k(x; p_k, x_k).
 \end{aligned}$$

This is unsatisfactory for several reasons: 1. the appearance of the l^0 "norm" makes it difficult due to its non-convexity, 2. the dependence on the parameters is non-convex and increases the dimensionality of the problem, and 3. the optimization should be run in "real-time" since it is used to decompose the solutions on-the-fly. To handle these possible issues, we propose a two-step optimization scheme in which the first step is convex and efficient but may not be accurate and the second step is non-convex, accurate, and made feasible by restricting onto a smaller dimensional problem.

First, we fix the parameters (p_k, x_k) by assigning $p_k = \bar{p}$, where \bar{p} is an approximate value based on the data and by setting x_k to be fixed multiples of the grid points. We replace the l^0 regularizer by the l^1 norm:

$$\begin{aligned}
 & \min_{\partial_x v, \alpha} \|\alpha\|_{l^1} \\
 & \text{s.t. } \partial_x u = \partial_x v + \sum_k \alpha_k \partial_x \tilde{\phi}_k
 \end{aligned} \tag{2.8}$$

where we intentionally removed the dependence of the basis terms on their parameters. It was shown that the l^1 norm provides a convex relaxation to the l^0 "norm" for certain inverse problems [5]. To solve Eq. (2.8), first consider the case when $\partial_x v = 0$:

$$\begin{aligned}
 & \min_{\alpha} \|\alpha\|_{l^1} \\
 & \text{s.t. } \partial_x u = \Phi \alpha.
 \end{aligned} \tag{2.9}$$

We replace the sum by a linear operator: $\Phi\alpha := \sum_k \alpha_k \partial_x \tilde{\phi}_k$. Then, following the work of [35], we relax the problem further using an l^2 norm perturbation:

$$\begin{aligned} \min_{\alpha} \quad & \|\alpha\|_{l^1} + \frac{\epsilon}{2} \|\alpha\|_2^2 \\ \text{s.t.} \quad & \partial_x u = \Phi\alpha. \end{aligned} \tag{2.10}$$

It is shown that for $0 < \epsilon \leq 10\|\alpha^*\|_{\infty}$, where α^* is the exact solution of Eq. (2.9), the minimization above is an exact regularization in the sense that the solutions will agree with solutions of the original minimization [15]. The dual problem of the exact regularization becomes:

$$\min_{\beta} - \langle \partial_x u, \beta \rangle_{L^2} + \frac{1}{2\epsilon} \|\Phi^T \beta - P(\Phi^T \beta)\|_2^2$$

where P is the point-wise projection onto the l^{∞} unit ball. The minimization can be solved via the gradient descent method:

$$\begin{cases} \beta^{j+1} = \beta^j + \sigma(\partial_x u - \frac{1}{\epsilon} \Phi S(\Phi^T \beta^j)) \\ \alpha^{j+1} = \frac{1}{\epsilon} S(\Phi^T \beta^{j+1}) \end{cases}$$

where j is the iterative index, $\sigma > 0$ is the pseudo-timestep, and $Sz := z - Pz$. To accelerate the convergence of the scheme, Nesterov’s algorithm is used [22]:

$$\begin{cases} \Theta^j = \frac{1 + \sqrt{1 + 4(\Theta^{j-1})^2}}{2} \\ \theta^j = \frac{1 - \Theta^j}{\Theta^{j+1}} \\ \tilde{\beta}^j = \beta^j + \theta^j(\beta^j - \beta^{j-1}) \\ \beta^{j+1} = \tilde{\beta}^j + \sigma(\partial_x u - \frac{1}{\epsilon} \Phi S(\Phi^T \tilde{\beta}^j)) \\ \alpha^{j+1} = \frac{1}{\epsilon} S(\Phi^T \beta^{j+1}) \end{cases}$$

where $\Theta^0 = 0$. Since the optimization problem is convex in α , the scheme uniquely determines α . This method converges to the optimal solution of Eq. (2.10) at a rate of $\mathcal{O}(\frac{1}{j^2})$. To account for the case when $\partial_x v \neq 0$, the iterations are stopped when

$$\|\partial_x v\|_2 = \|\partial_x u - \Phi\alpha\| \leq \text{tol},$$

which is consistent with the regularity assumption on v . The assumptions made in these approximations place the problem within a convex framework, making it easier to solve. However, the accuracy of the approximation relies on the guess of the basis parameters as well as the stopping criteria. Therefore, once α is determined, its support set and values are used as an initial guess to solve a local fitting of the true parameters. This is done as follows: given a non-zero α_k , fix the index k and solve the least squares problem:

$$\min_{p_k, x_c, k} \int_{B(x_c^0, r)} (u^n - \tilde{\phi}_k(x; p_k, x_c))^2 dx,$$

where the integral is over the ball of radius r centered at the guess x_c^0 . Using the Levenberg–Marquardt algorithm (in MATLAB), this optimization is done locally for each of the non-trivial values of α_k . Although this local refinement is non-convex, since we restrict the solutions to a small region near the initial guess, we expect the minimization to converge quickly to a local minimizer. Also, this allows us to correct any inaccuracies obtained in the convex minimization. Once all of the basis terms are determined locally, the function v is defined as the remainder $v = u - w$.

2.4. System of equations

Assuming that at time t^n , we have decomposed the solution u^n into $u^n = v^n + w^n$, we can evolve Eq. (2.4) in order to approximate Eq. (2.2). Based on the ansatz for w , we want to guarantee that w remains in B during the evolution. To compute the system of equations (see Eq. (2.4)), we use a first order splitting on the evolution equation for w . This amounts to taking one step forward on the following three PDE:

$$\partial_t w + w \partial_x w = \gamma \partial_x^2 w, \tag{2.11a}$$

$$\partial_t w + v \partial_x w = 0, \tag{2.11b}$$

$$\partial_t v + (v + w) \partial_x v = \gamma \partial_x^2 v. \tag{2.11c}$$

In this form, each equation is easier to solve. It is possible to use higher-order splitting methods (such as Strang splitting); however, we will show that to preserve the approximation of Eq. (2.11b) our method will degenerate to first order. So we do not consider higher-order splitting methods here and will consider it in subsequent works.

Solving Eq. (2.11a) forward by one time step can be done completely analytically:

$$w^{n+\frac{1}{2}} = \sum_k \chi_k^{n+\frac{1}{2}} \phi(x; e_{1,k}^{n+\frac{1}{2}}, e_{2,k}^{n+\frac{1}{2}}, \chi_k^{n+\frac{1}{2}});$$

with $e_{1,k}^{n+\frac{1}{2}} = e_{1,k}^n$, $e_{2,k}^{n+\frac{1}{2}} = e_{2,k}^n$ and $\chi_k^{n+\frac{1}{2}} = \chi_k^n + \frac{e_{1,k}^n + e_{2,k}^n}{2} \Delta t$. The characteristic functions, χ_k^n , are adjusted so that the boundaries are between two adjacent centers and the function ϕ is defined in Eq. (2.3).

Computing Eq. (2.11b) directly will cause w^{n+1} to leave the space B . Instead, we solve the problem analytically:

$$w^{n+1} = \sum_k \chi_k^{n+1} \phi(x; e_{1,k}^{n+1}, e_{2,k}^{n+1}, \chi_k^{n+1}),$$

with $e_{1,k}^{n+1} = e_{1,k}^n$, $e_{2,k}^{n+1} = e_{2,k}^n$ and

$$\chi_k^{n+1} = \chi_k^{n+\frac{1}{2}} - \int_{t^n}^{t^{n+1}} s_k(\tau) d\tau,$$

where $s_k(t) := v(x_k(t), t)$. Numerically, we approximate the values of v at points $(x_k(t), t)$ via interpolation. In practice, since the overall method is first order, a first order update is sufficient, i.e. $x_k^{n+1} = x_k^{n+\frac{1}{2}} - s_k^n \Delta t$, but it can be shown numerically that direct integration of the interpolating polynomial will produce a slightly more accurate approximation.

Lastly, to compute (2.11c) and update v , we use the standard central differencing method:

$$\frac{v_m^{n+1} - v_m^n}{\Delta t} + (v_m^n + w_m^{n+1}) \frac{v_{m+1}^n - v_{m-1}^n}{2h} = \gamma \frac{v_{m+1}^n - 2v_{m-1}^n + v_{m-1}^n}{h^2}, \tag{2.12}$$

where m is the spatial index.

Remark 2.1. For general flux terms, the first splitting is written as:

$$\begin{cases} \partial_t v + f'(v + w) \partial_x v = \gamma \partial_x^2 v \\ \partial_t w + f'(v + w) \partial_x w = \gamma \partial_x^2 w, \end{cases}$$

and is further separated to:

$$\begin{cases} \partial_t w + f'(w) \partial_x w = \gamma \partial_x^2 w \\ \partial_t w + (f'(v + w) - f'(w)) \partial_x w = 0 \\ \partial_t v + f'(v + w) \partial_x v = \gamma \partial_x^2 v \end{cases} \tag{2.13}$$

The method proposed in this section can be applied directly to Eq. (2.13) – we approximate the wave speed using:

$$s_k^n = (f'(v + w) - f'(w))|_{x=x_k^{n+1/2}}.$$

3. Numerical analysis

In this section, we provide several theoretical results on the method proposed here. In Section 3.1, we show that u can be approximated using the decomposition, with $v \in A$ and $w \in B$, which amounts to showing L^2 is contained in $A + B$. We show consistency of the PDE solver in Section 3.2. The theoretical results here are shown for Burgers' equation, but the analysis could be extended to other viscous conservation law.

3.1. Decomposition error analysis

In this subsection we consider the approximation theory. We would like to show that the approximate decomposition space is dense in the solution space. The main result appears in Theorem 3.4.

To do that, we firstly set up the mesh to discretize A :

$$G^A = \{x_m^A, m = 0, \dots, M + 1\},$$

with mesh sizes $h_A = \frac{1}{M}$. The low-energy space is defined by $A_{r,M} = P_{r,M}$, where P_r is the collection of all piecewise r th order polynomials on G^A . The superposition set, therefore defined as:

$$S_{r,M} := A_{r,M} + B = \{h = f + g, \quad \text{with } f \in A_{r,M}, g \in B\},$$

contains all functions which can be decomposed into the sets $A_{r,M}$ and B . Once again, the decomposition is not unique and depends on the algorithm used; however, we can show that under simple assumptions, this type of decomposition converges to the correct solution.

We define the distance of a given function to this superposition function space:

$$d_{r,M} := \max_{u \in H^1} \min_{f \in A_{r,M}, g \in B} \|u - f - g\|_2.$$

We also denote the distance measured in a traditional fashion:

$$\tilde{d}_{r,M} := \max_{u \in H^1} \min_{f \in A_{r,M}} \|u - f\|_2.$$

We want to show that $d_{r,M} \rightarrow 0$ as $M \rightarrow \infty$ for all r . Note that the error has the following ordering: $d_{r,M} \leq d_{1,M} \leq \tilde{d}_{1,M}$, thus we first consider the case of linear interpolation.

Theorem 3.1. *Let $v \in H^s$ and π be the r th order interpolant, i.e. $\pi v \in P_r$. Then there is an absolute constant C such that:*

$$\|v - \pi v\|_2 \leq Ch_A^\alpha |v|_{H^\alpha} \tag{3.1}$$

for any $\alpha \leq r + 1$, where $|v|_{H^\alpha} := \|\partial_x^\alpha v\|_2$ is the H^α semi-norm.

This theorem is a classical result in approximation theory, so we omit the proof here. Using this interpolation bound, we have the following result.

Corollary 3.2. *Under the assumption of Theorem 3.1, one has:*

$$\min_{f \in A_{r,M}, g \in B} \|u - f - g\|_2 \leq Ch_A^{r+1} |u|_{H^{r+1}}.$$

Proof. The proof is straightforward as one observes $S_{r,M} \supset A_{r,M}$, and the minimization can be bounded above by:

$$\begin{aligned} \min_{f \in A_{r,M}, g \in B} \|u - f - g\|_2 &\leq \min_{f \in A_{r,M}} \|u - f\|_2 \\ &\leq \|u - \pi u\|_2. \end{aligned}$$

The final term is bounded above using Eq. (3.1). \square

This indicates that in the worst case, the method is still better than classical interpolation. To obtain a sharper estimate, especially for the scale separation case, we denote the set $\Omega_C := \{x : |u'| < C\}$ and Ω_C^c its complement. It can be easily seen that Ω_C^c is composed of several separated intervals, each of which has width no more than $d_C = \frac{2\|u\|_\infty}{C}$. Also, w is monotone in each of these separated intervals.

If we choose the threshold to satisfy $C < \frac{1}{\gamma}$, then the set $\Omega_{\frac{1}{\gamma}}$ has a non-trivial intersection with Ω_C^c and can be measured by:

$$\varepsilon_C := |\Omega_{\frac{1}{\gamma}} \cap \Omega_C^c| = |\{x : C < |u'(x)| < \gamma^{-1}\}|.$$

Assumption 3.3. *We say that the system has scale separation if ε_C is small when $C = \mathcal{O}(1) \ll \mathcal{O}(\gamma^{-1})$.*

In the following arguments, we let C vary (as a free parameter) and decompose the function as follows:

$$v = u|_{\Omega_C} \quad \text{and} \quad w = u|_{\Omega_C^c},$$

where v and w are zero outside the domains above. This is a rough decomposition and we use it to give an upper bound for $d_{r,M}$.

Theorem 3.4. *Let u be a smooth function and assume that v and w are constructed as above. Then the following holds:*

- For arbitrary C :

$$d_{r,M} \lesssim Ch_A + \mathcal{O}\left(\frac{1}{\sqrt{C}}\right),$$

where $h_A = \frac{1}{M}$.

- If the solution has scale separation, then:

$$d_{r,M} \lesssim Ch_A + \mathcal{O}(\sqrt{\gamma}) + \mathcal{O}(\sqrt{\varepsilon}).$$

Proof. First note that:

$$\begin{aligned} \min_{f \in A_{r,M}, g \in B} \|u - f - g\|_2 &= \min_{f \in A_{r,M}, g \in B} \|v + w - f - g\|_2 \\ &\leq \min_{f \in A_{r,M}, f|_{\Omega_C^c} = 0} \|f - v\|_2 + \min_{g \in B, g|_{\Omega_C} = 0} \|g - w\|_2. \end{aligned}$$

Using Corollary 3.2, we can bound the error between f and v by:

$$\|f - v\|_{2, \Omega_C}^2 \lesssim h_A^2 \|v'\|_{2, \Omega_C}^2 \Rightarrow \|f - v\|_2 \lesssim Ch_A. \tag{3.2}$$

To analyze the error between g and w , note that both functions are monotonically decreasing and are nonzero only in isolated intervals with the width no more than $d_C = \frac{2\|u\|_\infty}{c}$. Therefore, the L^2 error has the bound:

$$\|g - w\|_{2, \Omega_C} \leq 2\sqrt{d_C} \|u\|_\infty. \tag{3.3}$$

Combining these two bounds concludes the first part of the theorem.

When the scale separation assumption holds, one easily sees that $d_C \leq d_{1/\gamma} + \varepsilon_C$. Also, we have that $d_{1/\gamma} < \frac{2\|u\|_\infty}{1/\gamma} = \mathcal{O}(\gamma)$. Plugging it into Eq. (3.3) one obtains:

$$d_{r,M} \lesssim Ch_A + \mathcal{O}(\sqrt{\gamma}) + \mathcal{O}(\sqrt{\varepsilon}). \quad \square \tag{3.4}$$

Remark 3.5. We briefly compare our decomposition with the classical methods.

- As argued in Corollary 3.2, without the scale separation assumption our method is, at worst, better than the classical method;
- With scale separation, the classical method gives:

$$\|u - \tilde{u}\|_2^2 = \|v - \tilde{u}|_{\Omega_C}\|_2^2 + \|w - \tilde{u}|_{\Omega_C^c}\|_2^2 \lesssim h^2 |v|_{H^1}^2 + h^2 |w|_{H^1}^2.$$

Since there is no control on the $|w|_{H^1}$ term, the error bound is bigger than what we obtained in Theorem 3.4. We verify this numerically in Section 4.6.

Remark 3.6. Theorem 3.4 is achieved under ideal circumstances. We expect other computational errors to be incurred, which we list below.

1. Since we need to compute $u'(x)$ for the decomposition, we incur additional computational error. This can be accounted for by modifying the constant C in Eq. (3.2) to $\max(C, 2h^2 \|w\|_\infty)$, where $h = \frac{1}{N}$.
2. The variational decomposition model (see Section 2.3) also produces errors in its approximation of the parameters, which is not considered here.

3.2. Consistency

In this section we will show that the error incurred during one evolution step using Eqs. (2.11a)–(2.11c) is $\mathcal{O}(\sqrt{\gamma}\Delta t + \Delta x^2)$.

First, the error in splitting the evolution of w into two steps is $\mathcal{O}(\Delta t^2)$. There is no error in solving Eq. (2.11a) since this is done analytically. Also, Eq. (2.11c) is solved using central differencing on v , and the error estimation is standard. The only non-trivial error that must be shown is for the evolution of Eq. (2.11b), under our assumption that the variable coefficient hyperbolic equation is replaced by constant coefficients.

For convenience, we assume that w is made up of one profile, but the arguments below can be generalized to multi-profile solutions. To approximate the solution, W , to Eq. (2.11b), we use:

$$\partial_t w + s(t)\partial_x w = 0, \quad w(x, 0) = W(x, 0), \tag{3.5}$$

where the initial condition $w(x, 0) = \phi(x; c_1, c_2, x_0) \in B$ and $s(t) = v(x_1(t), t)$ with $x_1(t) = x_0 - \int_{t^n}^t v(x_1(\tau), \tau) d\tau$ being the center of the viscous profile.

Theorem 3.7. Let w be computed by one step of Eq. (3.5) and W be the solution computed by one step of Eq. (2.11b), then

$$\|W - w\|_2 \lesssim \mathcal{O}(\sqrt{\gamma}\Delta t)$$

for sufficiently small $\gamma > 0$.

Proof. By subtracting Eq. (2.11b) from Eq. (3.5), we obtain that the error term $E := w - W$ satisfies:

$$\partial_t E + v\partial_x E = v\partial_x w - s(t)\partial_x w.$$

Using the typical energy estimate (multiplying the PDE above by E and integrating with respect to x) yields:

$$\partial_t E^2 + v \partial_x E^2 \lesssim 2E (v \partial_x w - s(t) \partial_x w).$$

Perform integration by parts and use the fact that $|\partial_x v|_\infty$ is bounded; one obtains

$$\frac{d\|E\|_2}{dt} \lesssim \|E\|_2 + D,$$

where

$$D := \|(v(x) - s(t)) \partial_x w\|_2.$$

Note that the constant in front of $\|E\|_2$ is $|\partial_x v|_\infty$ and it can be as big as $\mathcal{O}(\frac{1}{\gamma})$. Under our scale separation assumption, it is $\mathcal{O}(1)$. Therefore, by Gronwall's inequality we have:

$$\|E\|_2 \lesssim e^t D. \tag{3.6}$$

Since w satisfies Eq. (3.5), with its initial condition in B , the solution is given analytically by:

$$w(x, t) = \phi(x; c_1, c_2, x_1).$$

Taking its derivative with respect to x , we have:

$$\partial_x w = -\frac{(c_2 - c_1)^2}{2\gamma} \cdot \frac{e^{\zeta(x)}}{(1 + e^{\zeta(x)})^2}, \quad \text{where } \zeta(x) := \frac{c_2 - c_1}{2\gamma}(x - x_1).$$

This gives:

$$\begin{aligned} D^2(t) &= \int |(v(x, t) - s(t)) \partial_x w|^2 dx = \int |(v(x, t) - v(x_1, t)) \partial_x w|^2 dx \\ &\leq \|v'\|_\infty^2 \int |(x - x_1) \partial_x w|^2 dx. \end{aligned} \tag{3.7}$$

Note that $\partial_x w$ is centered around x_1 , so by plugging $\partial_x w$ into Eq. (3.7), we have:

$$D^2(t) = (c_2 - c_1)\gamma \int |\zeta \frac{e^\zeta}{(1 + e^\zeta)^2}|^2 d\zeta \lesssim \gamma \left(\int_0^\infty (\zeta e^{-\zeta})^2 d\zeta + \int_{-\infty}^0 (\zeta e^\zeta)^2 d\zeta \right) = \mathcal{O}(\gamma),$$

which implies $D = \mathcal{O}(\sqrt{\gamma})$. And with Eq. (3.6), we obtain:

$$E(\Delta t) = e^{\Delta t} (\sqrt{\gamma} \Delta t) = \mathcal{O}(\sqrt{\gamma} \Delta t). \quad \square$$

4. Numerical results

We validate our approximation on various one dimensional viscous conservation laws. The numerical results focus on the cases where the grid resolution is on the same scale as the viscosity, meaning the solution is under-resolved by conventional methods and the solutions have scale separation.

4.1. Burgers

First we consider Burgers' equation:

$$\partial_t u + \partial_x \left(\frac{u^2}{2} \right) = \gamma \partial_x^2 u$$

with Neumann boundary conditions and with the initial condition:

$$u(x, 0) = \frac{1.5}{1 + \exp(\frac{1.5(x-0.2)}{2\gamma})} + \frac{1}{1 + \exp(\frac{x-0.8}{2\gamma})} + 0.4 \sin(3\pi x)$$

containing two viscous profiles and a smooth oscillatory component. In Fig. 4.1, we simulate the evolution until $t = 0.5$ with $\gamma = 2.5 \times 10^{-2}$ using a high-resolution central differencing scheme (shown in black), the central differencing scheme on a grid with 64 points (shown in red), and our splitting method on a grid with 64 points (shown in blue). Our method generates solutions which do not contain the classical Gibbs phenomena (overshooting near sharp gradients). In Fig. 4.2(a), the solution over (x, t) verifies that our system of equations produces a smooth evolution over time with no intermediate instability. In Fig. 4.2(b), the space-time contours show the evolution of the sharp-contrast wave front.

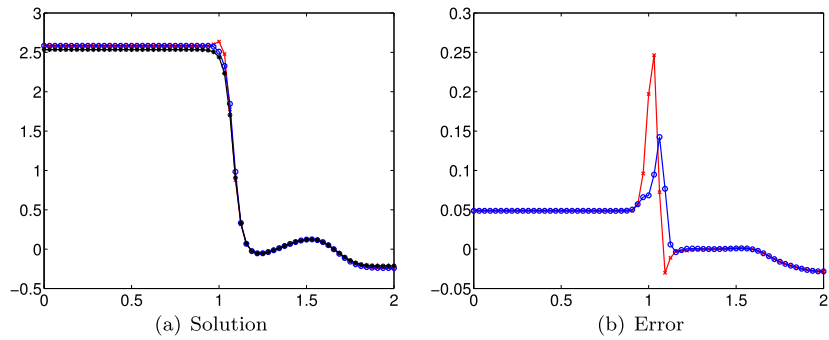


Fig. 4.1. Burgers' equation simulated with a high-resolution central differencing scheme (black), the central differencing scheme on a grid with 64 points (red), and our splitting method on a grid with 64 points (blue). The plots are taken at $t = 0.5$ with $\gamma = 2.5 \times 10^{-2}$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

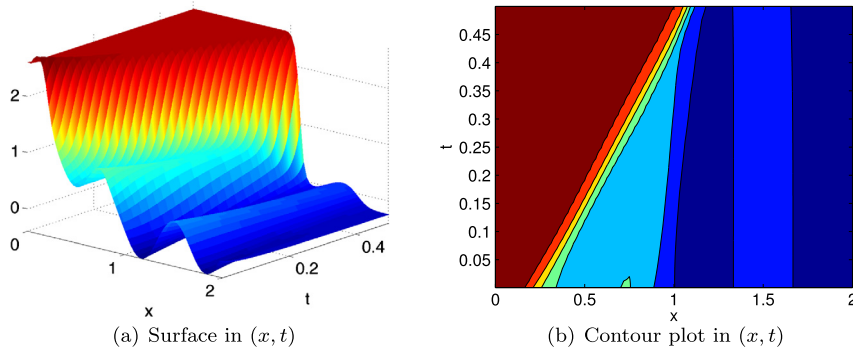


Fig. 4.2. Space-time plots of Burgers' equation simulated with our splitting method on a grid with $\gamma = 2.5 \times 10^{-2}$.

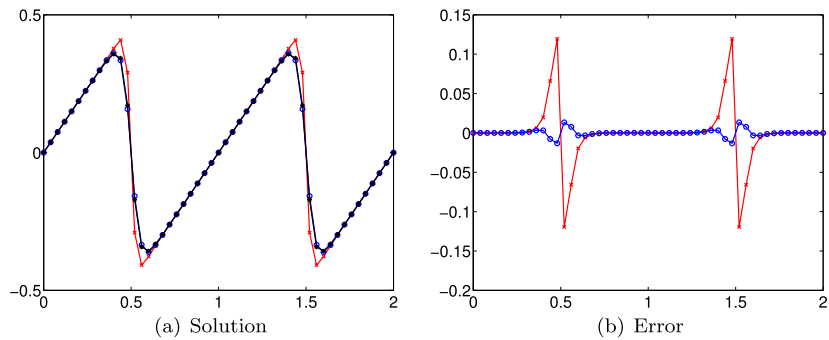


Fig. 4.3. Burgers' equation simulated with a high-resolution central differencing scheme (black), the central differencing scheme on a grid with 50 points (red), and our splitting method on a grid with 128 points (blue). The plots are taken at $t = 0.5$ with $\gamma = 1 \times 10^{-2}$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Next, we consider Burgers' equation ($\gamma = 1 \times 10^{-2}$) with Dirichlet boundary conditions and with the initial condition:

$$u(x, 0) = 0.5 \sin(2\pi x) + \sin(4\pi x)$$

which does not contain viscous profiles but will produce them later in time. In Fig. 4.3, we simulate the evolution until $t = 0.5$ with $\gamma = 1 \times 10^{-2}$ using a high-resolution central differencing scheme (shown in black), the central differencing scheme on a grid with 50 points (shown in red), and our splitting method on a grid with 50 points (shown in blue). Our algorithm produces a smoother (and more accurate) solution since it evolves the high-contrast part analytically. Fig. 4.4(a) contains a plot of the evolution of the approximate centers which shows the merging of the approximate wave fronts. Also, in Fig. 4.4(b) the solution over (x, t) shows that our method produces a smooth evolution over time even though it interprets the sine waves as viscous profiles (since on the unresolved grids they seem to have higher contrast). In Fig. 4.4(c), the space-time contours justify the motion of the characteristics found in Fig. 4.4(a).

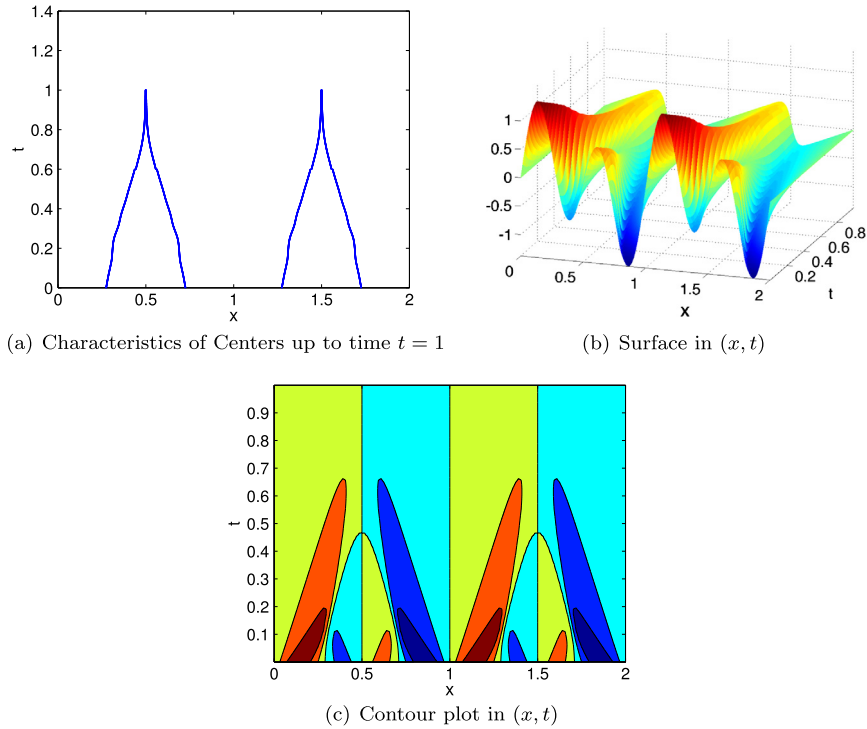


Fig. 4.4. Space-time plots of Burgers' equation simulated with our splitting method on a grid with $\gamma = 1 \times 10^{-2}$.

4.2. Traffic flow equation

Similar to Burgers' equation, we test the traffic flow equation:

$$\partial_t u + \partial_x ((1 - u)u) = \gamma \partial_x^2 u$$

with Dirichlet boundary conditions and with $\gamma = 0.5 \times 10^{-3}$. The initial condition:

$$u(x, 0) = 0.9 - \frac{0.25}{1 + \exp(\frac{0.25(x-1.25)}{\gamma})} - \frac{0.25}{1 + \exp(\frac{0.25(x-0.5)}{\gamma})} + 0.025 \sin(4\pi x)$$

contains two viscous profiles and a smooth oscillatory component. In Fig. 4.5(a)–(b), we simulate the evolution until $t = 2$ using a high-resolution central differencing scheme (shown in black) and our splitting method on a grid with 64 points (shown in blue). The central differencing scheme on a grid with 64 points does not produce a bounded solution, while our method will produce bounded results even on the under-resolved grids. Furthermore, our method approximates the viscous solution and not the discontinuous solution that a shock-capture method would. On a more resolved grid, we can see that our method again does not produce the overshooting behavior that smooth stencils would (see Fig. 4.5(c)–(d)). In Fig. 4.6, we see that our method captures the correct movement of the characteristics and the correct behavior of the smooth regions.

4.3. Cubic law

For a non-convex example, we test our method on the cubic conservation law:

$$\partial_t u + \partial_x \left(\frac{u^3}{3} \right) = \gamma \partial_x^2 u + F(x, t)$$

with Neumann boundary conditions and with $\gamma = 1 \times 10^{-2}$. The initial condition:

$$u(x, 0) = \frac{\sqrt{3} \times 0.3}{1 + \exp(\frac{2 \times 0.2(x+0.75)}{\gamma})}$$

contains one viscous profile. The background force:

$$F(x, t) = \max(\sin(10\pi t), 0) \sin(\pi(x - t + 2))$$

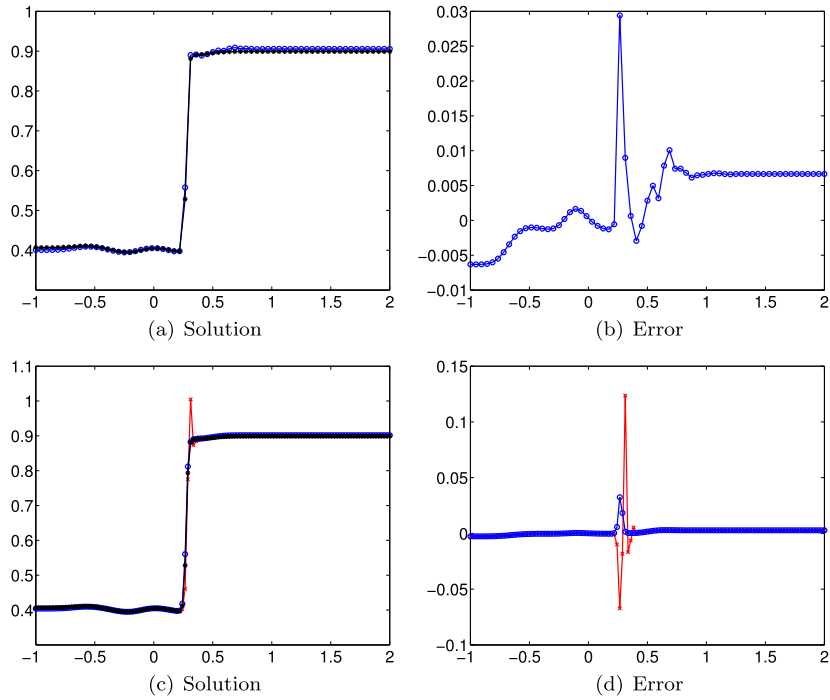


Fig. 4.5. Traffic flow equation simulated with a high-resolution central differencing scheme (black), the central differencing scheme on a grid with 128 (bottom) points (red), and our splitting method on a grid with 64 (top) or 128 (bottom) points (blue). The plots are taken at time $t = 2$ with $\gamma = 0.5 \times 10^{-3}$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

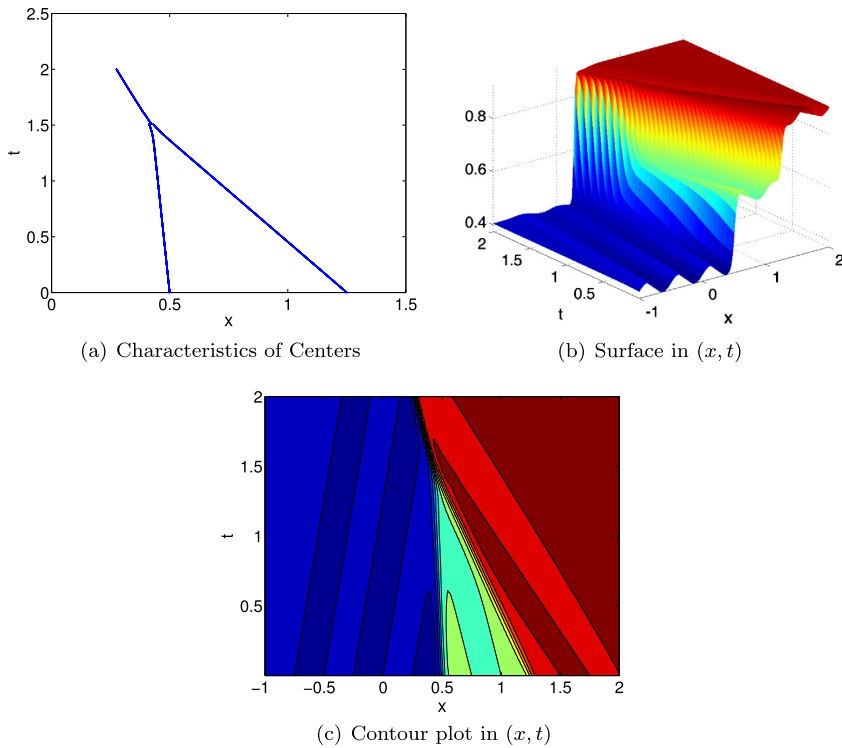


Fig. 4.6. Space-time plots of traffic flow equation simulated with our splitting method with $\gamma = 0.5 \times 10^{-3}$.

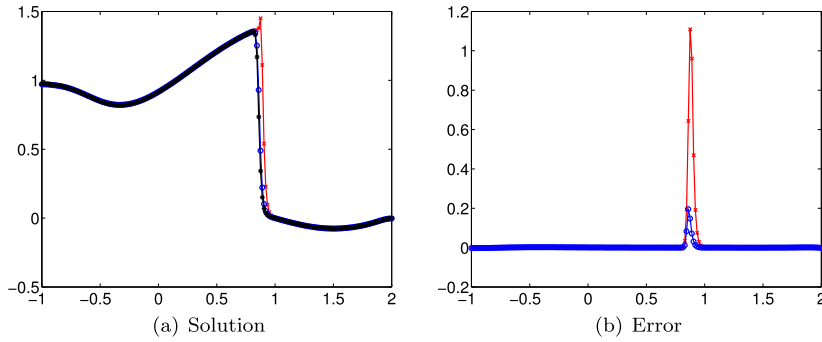


Fig. 4.7. Cubic law equation simulated with a high-resolution central differencing scheme (black), the central differencing scheme on a grid with 200 points (red), and our splitting method on a grid with 200 points (blue). The plots are taken at time $t = 4.25$ with $\gamma = 1 \times 10^{-2}$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

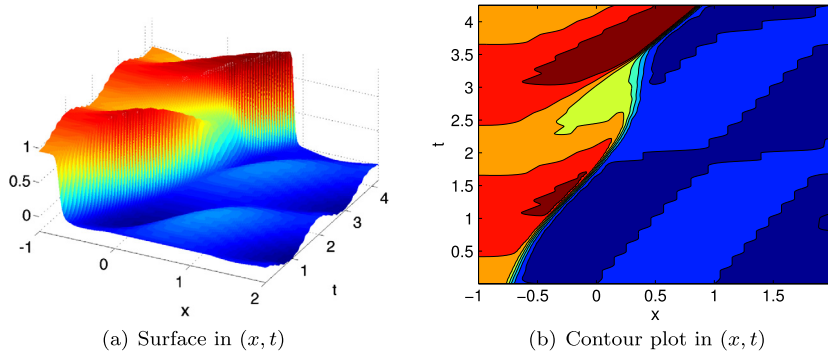


Fig. 4.8. Space–time plots of traffic flow equation simulated with our splitting method with $\gamma = 1 \times 10^{-2}$.

is oscillatory in space and in time. In Fig. 4.7, we simulate the evolution until $t = 4.25$ with $\gamma = 1 \times 10^{-2}$ using a high-resolution central differencing scheme (shown in black), the central differencing scheme on a grid with 50 points (shown in red), and our splitting method on a grid with 50 points (shown in blue). Our method locates the correct wave front over time, which is smoothly altered by the background force. The space–time dynamics appear in Fig. 4.8, where one can see the appearance of a new wave generated by the forcing term, that interacts with the larger viscous profile causing it to change its course.

4.4. A fractional power law

Our last example uses a less smooth flux:

$$\partial_t u + \partial_x (|u|^{3/2}) = \gamma \partial_x^2 u,$$

with time-dependent Dirichlet boundary conditions and with $\gamma = 2 \times 10^{-2}$. The initial condition:

$$u(x, 0) = \frac{1}{(1 + \exp(\frac{x+0.5}{2\gamma}))^2},$$

contains one viscous profile. The boundary condition on the left:

$$u(-1, t) = 0.25 \sin(3\pi t) + 1$$

is oscillatory in time. In Fig. 4.9, we simulate the evolution until $t = 2$ with $\gamma = 2 \times 10^{-2}$ using a high-resolution central differencing scheme (shown in black), the central differencing scheme on a grid with 64 (top) or 128 (bottom) points (shown in red), and our splitting method on a grid with 64 (top) or 128 (bottom) points (shown in blue). Our method locates the correct wave front over time, and as the grid is resolved, captures the correct smooth flow. The high contrast, yet smooth, space–time dynamics appear in Fig. 4.10, and are well-approximated using our method. This shows that our method can simulate viscous flows without the need to resolve the grid.

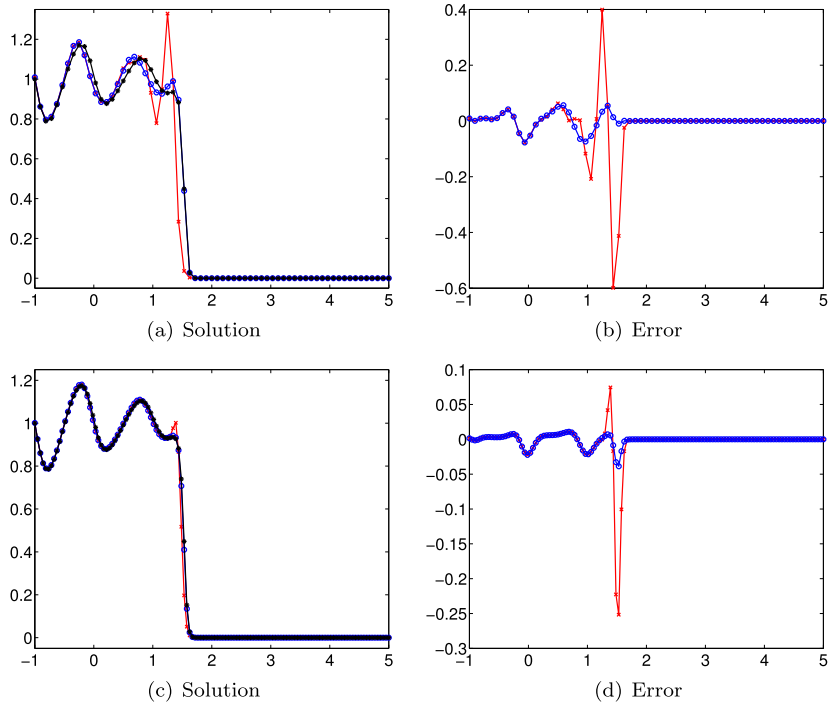


Fig. 4.9. The fractional law equation simulated with a high-resolution central differencing scheme (black), the central differencing scheme on a grid with 64 (top) or 128 (bottom) points (red), and our splitting method on a grid with 64 (top) or 128 (bottom) points (blue). The plots are taken at time $t = 2$ with $\gamma = 2 \times 10^{-2}$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

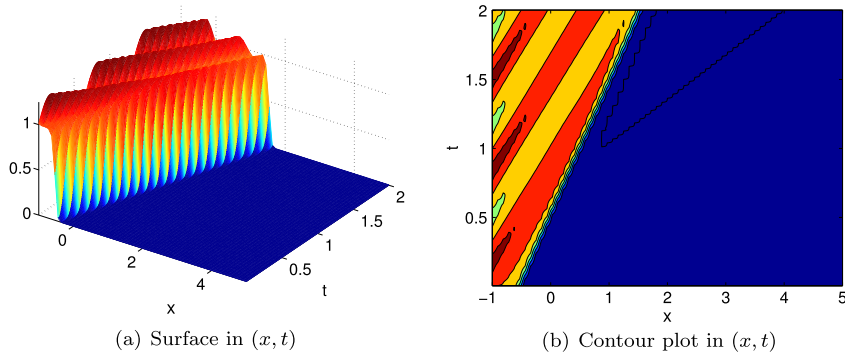


Fig. 4.10. Space-time plots of the fractional law equation simulated with our splitting method with $\gamma = 2 \times 10^{-2}$.

4.5. Convergence

To check the numerical convergence of our method, we simulate the dynamics with various grid resolutions. We take the maximum resolution to be 256, because if the grid is fully-resolved our approximation should degenerate to the standard central differencing scheme. Also our assumptions (for example, that the gradients of the sharp transition are very localized and that the solution is scale separated) will no longer be valid. However, since our method is developed to deal with viscous profiles on under-resolved grids, we feel that this is an appropriate grid size. The results are shown in Table 4.1, and suggest first order convergence.

4.6. Complexity

To analyze the relationship between the grid size N and the viscosity γ , we simulate Burgers' equation with shrinking (but non-zero) viscosity. The initial data is taken to be:

$$u(x, 0) = \frac{1}{1 + \exp(\frac{x-0.2}{2\gamma})} + 0.2 \sin(\pi x),$$

Table 4.1Errors (in L^2) calculated for the examples above.

Grid size	Error for Burgers	Errors for fractional law	Errors for cubic law
32	0.1302	0.1539	0.2690
50	0.0927	0.1039	0.1113
64	0.0623	0.0611	0.0423
100	0.0353	0.0248	0.0645
128	0.0265	0.0185	0.0450
256	0.0186	0.0128	0.0286
Order	1.0042	1.3166	0.9551

Table 4.2The relative L^2 and H^1 errors (as a percentage) for the solution generated by our method for various viscosities are calculated. The grid size is fixed in all cases to 50 points. The relative errors are stable as γ decreases.

γ	0.5×10^{-1}	0.25×10^{-1}	0.125×10^{-1}	0.625×10^{-2}	0.3125×10^{-2}	0.15625×10^{-2}
Error L^2	0.91%	0.79%	0.96%	1.00%	1.05%	1.03%
Error H^1	4.08%	3.23%	3.03%	2.94%	3.22%	2.28%

Table 4.3The minimal grid size necessary to achieve a relative L^2 error (as a percentage) of 1.05% for the solutions generated by our method are calculated. As a reference we also include the relative H^1 norm.

γ	0.5×10^{-1}	0.25×10^{-1}	0.125×10^{-1}	0.625×10^{-2}	0.3125×10^{-2}	0.15625×10^{-2}
Min. grid	46	39	46	50	50	49
Error L^2	1.03%	1.04%	1.05%	1.00%	1.05%	1.03%
Error H^1	4.33%	3.94%	3.20%	2.94%	3.22%	2.30%

and the initial separation is nearly exact. The equation evolves up to $t = 0.05$, which ensures that the system of equations are well separated for all time. The time-step $dt = \frac{dx^2}{2}$ is fixed during this experiment (dx^2 constraint could be further relaxed if v is computed implicitly for the diffusion term, but we do not pursue it in this paper). In Table 4.2, the grid size is fixed to $N = 50$ and the viscosity varies from 0.5×10^{-1} down to 0.15625×10^{-2} . The relative L^2 error (as a percentage) stays close to 1% in all cases. The relative H^1 error is shown as a reference. Next taking the largest relative L^2 error from Table 4.2 (which is 1.05%), we show that the minimal amount of grid points needed to achieve this error tolerance is stable (see Table 4.3). These two tests show that our splitting algorithm decouples γ and the grid size.

5. Conclusion

In order to compute solutions of viscous conservation laws, we borrow the idea of sparse + low-rank decompositions, to separate sharp-contrast features from smooth low-energy ones. Our method provides an appropriate approximation to the solution on a coarse grid, without the need to refine the grid resolution in order to resolve the continuity of the sharp transition. Theoretical results support a simplified version of the decomposition along with consistency of the numerical system of PDE that approximate the original dynamics. Several numerical results show the applicability of the proposed method to various equations.

Acknowledgements

T.Y. Hou is supported by AFOSR MURI Grant FA9550-09-1-0613, DOE grant DE-FG02-06ER25727, and NSF Grants DMS 1318377 and DMS 1159138. H. Schaeffer is supported by NSF 1303892.

References

- [1] Claude Bardos, Eitan Tadmor, Stability and spectral convergence of Fourier method for nonlinear problems. On the shortcomings of the 2/3 de-aliasing method, *Numer. Math.* (2013) 1–34.
- [2] Ido Bright, Guang Lin, J. Nathan Kutz, Compressive sensing based machine learning strategy for characterizing the flow around a cylinder with limited pressure measurements, *Phys. Fluids* 25 (12) (2013) 127102.
- [3] Steven L. Brunton, Jonathan H. Tu, Ido Bright, J. Nathan Kutz, Compressive sensing and low-rank libraries for classification of bifurcation regimes in nonlinear dynamical systems, arXiv:1312.4221, 2013.
- [4] Russel E. Caflisch, Stanley J. Osher, Hayden Schaeffer, Giang Tran, PDEs with compressed solutions, arXiv:1311.5850, 2013.
- [5] Emmanuel J. Candès, Justin Romberg, Terence Tao, Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information, *IEEE Trans. Inf. Theory* 52 (2) (2006) 489–509.
- [6] Emmanuel J. Candès, Xiaodong Li, Yi Ma, John Wright, Robust principal component analysis? *J. ACM* 58 (3) (2011).
- [7] Venkat Chandrasekaran, Sujay Sanghavi, Pablo A. Parrilo, Alan S. Willsky, Rank-sparsity incoherence for matrix decomposition, *SIAM J. Optim.* 21 (2) (2011) 572–596.

- [8] Mulin Cheng, Thomas Y. Hou, Mike Yan, Zhiwen Zhang, A data-driven stochastic method for elliptic PDEs with random coefficients, *SIAM/ASA J. Uncertain. Quantificat.* 1 (1) (2013) 452–493.
- [9] Mulin Cheng, Thomas Y. Hou, Zhiwen Zhang, A dynamically bi-orthogonal method for time-dependent stochastic partial differential equations I: derivation and algorithms, *J. Comput. Phys.* 242 (2013) 843–868.
- [10] Mulin Cheng, Thomas Y. Hou, Zhiwen Zhang, A dynamically bi-orthogonal method for time-dependent stochastic partial differential equations II: adaptivity and generalizations, *J. Comput. Phys.* 242 (2013) 753–776.
- [11] W.M. Feng, Peng Yu, S.Y. Hu, Zi-Kui Liu, Qiang Du, Long-Qing Chen, Spectral implementation of an adaptive moving mesh method for phase-field equations, *J. Comput. Phys.* 220 (1) (2006) 498–510.
- [12] Jonathan Goodman, Thomas Hou, Eitan Tadmor, On the stability of the unsmoothed Fourier method for hyperbolic equations, *Numer. Math.* 67 (1) (1994) 93–129.
- [13] Hui Ji, Sibin Huang, Zuwei Shen, Yuhong Xu, Robust video restoration by joint sparse and low rank matrix approximation, *SIAM J. Imaging Sci.* 4 (4) (2011) 1122–1142.
- [14] Dmitry Kolomenskiy, Kai Schneider, A Fourier spectral method for the Navier–Stokes equations with volume penalization for moving solid obstacles, *J. Comput. Phys.* 228 (16) (2009) 5687–5709.
- [15] Ming-Jun Lai, Wotao Yin, Augmented ℓ_1 and nuclear-norm models with a globally linearly convergent algorithm, *SIAM J. Imaging Sci.* 6 (2) (2013) 1059–1091.
- [16] Sajan Goud Lingala, Yue Hu, Edward DiBella, Mathews Jacob, Accelerated dynamic MRI exploiting sparsity and low-rank structure: k-t SLR, *IEEE Trans. Med. Imaging* 30 (5) (2011) 1042–1054.
- [17] Xu-Dong Liu, Stanley Osher, Tony Chan, Weighted essentially non-oscillatory schemes, *J. Comput. Phys.* 115 (1) (1994) 200–212.
- [18] Alan Mackey, Hayden Schaeffer, Stanley Osher, On the compressive spectral method, *UCLA CAM Report* 14-33, 2014.
- [19] Yvon Maday, Sidi M. Ould Kaber, Eitan Tadmor, Legendre pseudospectral viscosity method for nonlinear conservation laws, *SIAM J. Numer. Anal.* 30 (2) (1993) 321–342.
- [20] Andrew Majda, Stanley Osher, Propagation of error into regions of smoothness for accurate difference approximations to hyperbolic equations, *Commun. Pure Appl. Math.* 30 (6) (1977) 671–705.
- [21] Andrew Majda, James McDonough, Stanley Osher, The Fourier method for nonsmooth initial data, *Math. Comput.* 32 (144) (1987) 1041–1081.
- [22] Yurii Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course*, vol. 87, Springer, 2004.
- [23] Vidvuds Ozoliņš, Rongjie Lai, Russel Caflisch, Stanley Osher, Compressed modes for variational problems in mathematics and physics, *Proc. Natl. Acad. Sci. USA* 110 (46) (2013) 18368–18373.
- [24] Vidvuds Ozoliņš, Rongjie Lai, Russel Caflisch, Stanley Osher, Compressed plane waves—compactly supported multiresolution basis for the Laplace operator, *Proc. Natl. Acad. Sci. USA* 111 (5) (2013) 1691–1696.
- [25] Hayden Schaeffer, Stanley Osher, A low patch-rank interpretation of texture, *SIAM J. Imaging Sci.* 6 (1) (2013) 226–262.
- [26] Hayden Schaeffer, Russel Caflisch, Cory D. Hauck, Stanley Osher, Sparse dynamics for partial differential equations, *Proc. Natl. Acad. Sci. USA* 110 (17) (2013) 6634–6639.
- [27] Chi-Wang Shu, Stanley Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes, II, *J. Comput. Phys.* 83 (1) (1989) 32–78.
- [28] Chi-Wang Shu, Peter S. Wong, A note on the accuracy of spectral method applied to nonlinear conservation laws, *J. Sci. Comput.* 10 (3) (1995) 357–369.
- [29] Eitan Tadmor, Stability analysis of finite difference, pseudospectral and Fourier–Galerkin approximations for time-dependent problems, *SIAM Rev.* 29 (4) (1987) 525–555.
- [30] Eitan Tadmor, Convergence of spectral methods for nonlinear conservation laws, *SIAM J. Numer. Anal.* 26 (1) (1989) 30–44.
- [31] Eitan Tadmor, Knut Waagan, Adaptive spectral viscosity for hyperbolic conservation laws, *SIAM J. Sci. Comput.* 34 (2) (2012) A993–A1009.
- [32] Giang Tran, Hayden Schaeffer, William M. Feldman, Stanley J. Osher, An L1 penalty method for general obstacle problems, [arXiv:1404.1370](https://arxiv.org/abs/1404.1370), 2014.
- [33] Jonathan H. Tu, Clarence W. Rowley, Dirk M. Luchtenburg, Steven L. Brunton, J. Nathan Kutz, On dynamic mode decomposition: theory and applications, [arXiv:1312.0041](https://arxiv.org/abs/1312.0041), 2013.
- [34] Mathew O. Williams, Eli Shlizerman, Jon Wilkening, J. Nathan Kutz, The low dimensionality of time-periodic standing waves in water of finite and infinite depth, *SIAM J. Appl. Dyn. Syst.* 11 (3) (2012) 1033–1061.
- [35] Wotao Yin, Stanley Osher, Donald Goldfarb, Jerome Darbon, Bregman iterative algorithms for ℓ_1 -minimization with applications to compressed sensing, *SIAM J. Imaging Sci.* 1 (1) (2008) 143–168.