Transport Reversal for Model Reduction of Hyperbolic Partial Differential Equations*

Donsub Rim†, Scott Moe‡, and Randall J. LeVeque‡

Abstract. Snapshot matrices built from solutions to hyperbolic partial differential equations exhibit slow decay in singular values, whereas fast decay is crucial for the success of projection-based model reduction methods. To overcome this problem, we build on previous work in symmetry reduction [Rowley and Marsden, *Phys. D*, 142 (2000), pp. 1–19] and propose an iterative algorithm that decomposes the snapshot matrix into multiple shifting profiles, each with a corresponding speed. Its applicability to typical hyperbolic problems is demonstrated through numerical examples, and other natural extensions that modify the shift operator are considered. Finally, we give a geometric interpretation of the algorithm.

1. Introduction. Reduced order models (ROMs) can emulate the behavior of high-dimensional models (HDMs) with small computational cost. Therefore, once ROMs can be constructed, prohibitively expensive problems in control design or uncertainty quantification (UQ) can be tackled by using them as surrogate models in place of HDMs. Proper orthogonal decomposition (POD) and its variants [6, 9, 16, 17, 12, 29, 28, 35, 39] have been successfully applied to various partial differential equations (PDEs), including those arising in fluid dynamics [7, 11, 19, 25, 30, 41]. However, these projection-based methods can be ineffective when applied to compressible flow problems governed by hyperbolic PDEs. This difficulty is well-known and was noted in [1], where a dictionary-based model reduction method was developed, and in [8] where a fail-safe h-adaptive algorithm was introduced.

We will illustrate the main obstacle with a simple example. Consider the initial boundary value problem for the advection equation, whose solution u in the domain $\Omega \equiv (0, 1)$ satisfies the PDE

$$(1.1) \qquad\qquad u_t + cu_x = 0 \qquad \text{in } \Omega,$$

along with the periodic boundary condition and the initial condition

$$(1.2) \qquad u(0, t) = u(1, t) \quad \text{for } t \in [0, T], \qquad u(x, 0) = u_0(x) \equiv \delta(x).$$

†Current affiliation: Department of Applied Physics and Applied Mathematics, Columbia University, New York, 10027 (dr2965@columbia.edu).
    ‡Department of Applied Mathematics, University of Washington, Seattle, WA 98195 (smoe@uw.edu, rjl@uw.edu).

We assume $c = 1$ here. Let us seek a solution using the finite volume method with upwind flux [23]. We set the grid points $x_j = jh$ for $j = 0, 1, \ldots, N$ and let $h \equiv 1/N$, and define the cells $\mathcal{C}_j \equiv [x_{j-1/2}, x_{j+1/2}]$, where $x_{j+1/2} \equiv x_j + h/2$. Denote by u_j^n the approximation to the cell average of the solution at time t_n

$$(1.3) \qquad u_j^n \approx \frac{1}{h} \int_{x_{j-1/2}}^{x_{j+1/2}} u(x, t_n) \, \mathrm{d}x,$$

and also denote by $\mathbf{u}^n$ the vector $(u_j^n)_{j=1}^N \in \mathbb{R}^N$. Taking a fixed time-step of size $\Delta t = h$, it is easy to see that the finite volume solution at time t_n is just the scaled standard basis vector $\mathbf{u}^n = \mathbf{e}_n/h$. That is,

$$(1.4) \qquad \mathbf{u}^n = \mathbf{K}^n \mathbf{u}^0, \quad \text{where} \quad \mathbf{K} \equiv \begin{bmatrix} 0 & 0 & \cdots & 0 & 1 \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix} \quad \text{and} \quad \mathbf{u}^0 = \frac{1}{h} \mathbf{e}_1.$$

This solution has no error apart from the discretization of the initial data and thereafter reproduces the cell averages (1.3) exactly.

The *snapshot matrix* $\mathbf{A}$, taken at times $\{t_n\}_{n=0}^{N-1}$, is given by

$$(1.5) \qquad \mathbf{A} \equiv \begin{bmatrix} \mathbf{u}^0 & \mathbf{u}^1 & \cdots & \mathbf{u}^{N-1} \end{bmatrix} = \frac{1}{h} \mathbf{I}, \quad \text{where } \mathbf{I} \text{ is the identity in } \mathbb{R}^{N \times N}.$$

In POD, we take the singular value decomposition (SVD) of the matrix $\mathbf{A}$. It is easy to see that $\mathbf{A}$ has singular values $\sigma_1 = \cdots = \sigma_N = 1/h$. We then truncate the rank-1 expansion of $\mathbf{A}$ after some R terms. Usually, we choose the smallest R such that, for a given tolerance $\varepsilon \ll 1$, the remainder satisfies

$$(1.6) \qquad \sum_{j=R+1}^N \sigma_j^2 \Big/ \sum_{j=1}^N \sigma_j^2 < \varepsilon, \quad \text{in this case } 1 - R/N < \varepsilon.$$

The left-hand side decreases linearly in R, so R must be large even for a moderately small ε to satisfy (1.6). If $\varepsilon < h$, it would require $R = N$, so that all singular values and corresponding singular vectors must be kept as reduced basis vectors. (Often $\mathbf{A}$ is preprocessed by subtracting from each column its mean, so that each column has zero mean. Doing so here would make σ_N equal to zero, but other singular values will not be changed, leaving the obstacle intact.)

This slow decay is commonly observed in snapshot matrices taken from hyperbolic problems. Therefore, existing projection-based methods quickly face a difficulty. The approach we adopt to overcome this is to focus on the low-dimensional *hyperbolic* behavior of the solution and treat it separately, to the extent possible. Simply put, we wish to construct the Lagrangian frame of reference. To do so directly is a challenging problem of its own right, so instead we devise a numerical method for utilizing this frame indirectly for our special purposes.

This main idea coincides with the so-called symmetry reduction that was studied in [32, 33] and similar ideas that appeared in the references therein. The target for reduction in that context is a continuous symmetry group G acting on a manifold M. In our setting, M is the L^2 inner product space of periodic functions, and G is the group of spatial translations. To reduce G, *template fitting* [18] is used to map the full dynamics of u to the quotient space M/G. Given a snapshot $u(x, t)$ at time t and a *template* $u_0(x)$, both periodic in $[0, 1]$, template fitting posits the minimization problem

$$(1.7) \qquad \min_a \int_0^1 |u(x - a, t) - u_0(x)|^2 \ \mathrm{d}x.$$

This minimization resembles the orthogonal Procrustes problem [13], which deals with data given in the form of sample points, rather than in discretized function values over a grid. If u is smooth, one obtains the equation for the minimum a_*, $\langle u(x, t), u_0'(x + a_*) \rangle = 0$, that defines the dynamics of $a_*(t)$. The orthogonality condition then allows one to identify the quotient space M/G with an affine space intersecting u_0 called a *slice* denoted by S_{u_0}. To summarize, for each given dynamics $u(t)$ in M, corresponding *slice dynamics* $r(\tau)$ in S_{u_0} can be found. After a reduction for $r(\tau)$ is found in S_{u_0}, there are reconstruction equations that can be used to recover the original dynamics $u(t)$ [33]. The main advantage is that $r(\tau)$ in the space S_{u_0} may yield low-dimensional structure more readily, even when $u(t)$ itself does not. This key property is the inspiration for this work.

Following the template fitting approach, we propose generalizations which expand its applicability. In [33] the dynamics of the infinitesimal action for the reconstruction were formulated, and then the system was integrated numerically. Here we consider the direct discretization of (1.7), and then devise a greedy algorithm we call the *transport reversal*. The algorithm terminates when the snapshot matrix can be well-approximated by the superposition of multiple transport dynamics. The main ideas are (1) the projection onto the *template* or the *pivot* for scaling, (2) the use of *cut-off vectors* to modify the pivot, and (3) enforcing of regularity in the minimization problem to obtain smooth transport dynamics. The details appear in section 2.

In the subsequent sections, we consider two extensions of the shift operator. The upwind flux is used to extend the shift numbers to real numbers in section 3.1. Then an extension to the case where the speed c in (1.1) varies with respect to the spatial variable is introduced in section 3.4. Nothing prevents these extensions from being used in conjunction with the iterative transport reversal algorithm introduced in the preceding section. In section 4, we present some geometric interpretations.

Transport reversal shares features with the shifted proper orthogonal decomposition introduced in [31]. It can be related to the dynamic mode decomposition (DMD) [34, 38] in the sense that the periodic shift operator in (1.4) is a linear operator generating the dynamics on the state space $\mathbb{R}^N$, but the objectives differ. Here we assume that a specific dynamic, namely, transport, is present in the data, whereas DMD aims to discover the spectral properties of the Koopman operator derived from the data itself.

The discovery of the hyperbolic structure through this algorithm is only a first step toward building a ROM for hyperbolic PDEs. Using this output to build a ROM requires tackling further issues that will be pursued in future work. Once ROMs can be constructed for any

parameter value, they can be used to explore the solution behavior in parameter space. In many practical applications the parameter space is high-dimensional, so one needs a strategy for constructing a global model that is not sensitive to the number of dimensions. To this end, various interpolation methods incorporating adaptive and greedy strategies have been introduced [2, 3, 4, 24, 26]. The algorithms in this paper allows one to apply these methods in conjunction with *displacement interpolation* (see, e.g., [40]) thereby incorporating the Lagrangian frame into the approximation procedure. The approach given here may well supplement not only existing model reduction methods, but also UQ methods such as the generalized polynomial chaos [27].

The transport reversal extends naturally to the multidimensional setting. The key component in the extension is the use of the intertwining property of the Radon transform [15]. In exploiting this remarkable property, one obtains a multidimensional extension of the large time-step [LTS] method [20, 21, 22], and therefore the multidimensional analogue of the transport reversal algorithm. The scope of this paper does not permit a detailed account of this important extension. A thorough treatment will appear elsewhere, based on the one-dimensional theory presented in this paper.

2. Transport reversal. In this section, we discretize and generalize the template fitting problem (1.7). The main objective of this section is to extend the idea of template fitting to more complex situations that arise from important examples of hyperbolic problems. We will introduce new enhancements that will result in a greedy iteration we call the transport reversal algorithm.

To motivate the discussion, let us revisit the problem (1.1). Recall the finite volume solution $\mathbf{u}^n$, the matrix of shifts with periodic boundary conditions $\mathbf{K}$ (1.4), and the snapshot matrix $\mathbf{A}$ (1.5). With this notation, the columns of $\mathbf{A}$ can be rewritten in terms of the Krylov subspace generated by $\mathbf{K}$ in using the fact that $\mathbf{u}^n = \mathbf{K}^n \mathbf{u}^0$,

$$\mathbf{A} = \begin{bmatrix} \mathbf{u}^0 & \mathbf{K}\mathbf{u}^0 & \cdots & \mathbf{K}^{N-1}\mathbf{u}^0 \end{bmatrix}.$$

Suppose we preprocess $\mathbf{A}$ to obtain $\mathring{\mathbf{A}}$,

$$(2.1) \qquad \mathring{\mathbf{A}} \equiv \begin{bmatrix} \mathbf{I} & \mathbf{K}^{-1} & \mathbf{K}^{-2} & \cdots & \mathbf{K}^{-(N-1)} \end{bmatrix} \odot \begin{bmatrix} \mathbf{u}^0 & \mathbf{u}^1 & \mathbf{u}^2 & \cdots & \mathbf{u}^{N-1} \end{bmatrix},$$

where the notation $\odot$ denotes componentwise multiplication between a list of matrices and a list of column vectors. It follows that

$$(2.2) \qquad \mathring{\mathbf{A}} = \begin{bmatrix} \mathbf{u}^0 & \cdots & \mathbf{u}^0 \end{bmatrix} = \begin{bmatrix} \frac{1}{h} & \frac{1}{h} & \cdots & \frac{1}{h} \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix} = \begin{bmatrix} N & N & \cdots & N \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix},$$

and also that $\mathring{\mathbf{A}}$ has the trivial SVD

$$(2.3) \qquad \mathring{\mathbf{A}} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^* = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \begin{bmatrix} N\sqrt{N} \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{N}} & \frac{1}{\sqrt{N}} & \cdots & \frac{1}{\sqrt{N}} \end{bmatrix}.$$

Hence the singular values of $\mathring{\mathbf{A}}$ are $\sigma_1 = N\sqrt{N}$ and $\sigma_2 = \cdots = \sigma_N = 0$. In short, when SVD is applied to $\mathring{\mathbf{A}}$ rather than $\mathbf{A}$ there is only one nonzero singular value, yielding a reduced basis with a single element $\{\mathbf{u}_0\}$. By shifting each snapshot by an appropriate number of grid cells (reversing the transport due to the hyperbolic equation) they all line up. This procedure can be seen as a straightforward discretization of (1.7), and we formulate its generalization as follows.

Definition 2.1 (shift numbers). *Let* $\mathbf{A} \in \mathbb{R}^{N \times M}$ *be a real matrix and* $\mathbf{b} \in \mathbb{R}^N$ *a real vector we will call the* pivot*. Denote by* $\mathbf{a}_j$ *the* j*th column of* $\mathbf{A}$*, and then define the integers* $\nu_j \in \mathbb{Z}_N$ *to be the minimizers*

$$(2.4) \qquad \nu_j = \mathrm{argmin}_{\omega \in \mathbb{Z}_N} \|\mathbf{a}_j - \mathbf{K}^\omega \mathbf{b}\|_2^2 \quad \text{for } j = 1, 2, \ldots, M.$$

Whenever the minimization is not unique, we choose one closest to 0.

We call $\{\nu_j\}$ *the* shift numbers *and organize them in a vector* $\boldsymbol{\nu} \equiv (\nu_j)_{j=1}^M$*. We denote the computation of* $\boldsymbol{\nu}$ *in* (2.4) *as*

$$(2.5) \qquad \boldsymbol{\nu} = \mathcal{C}(\mathbf{A}; \mathbf{b}).$$

In (2.4) we are merely shifting the entries of the pivot $\mathbf{b}$ to match $\mathbf{a}_j$ as much as possible. We remark that the choice of the ℓ^2-norm in the minimization problem (2.4) is not unique and other norms could have been used instead. However, we will use the ℓ^2-nrom throughout this paper for simplicity. Next, we introduce some more notation regarding the computation $\mathcal{C}$.

Notation 2.2. *Pivot operations.*
- *Let* $\mathcal{C}(\mathbf{A}; j) \equiv \mathcal{C}(\mathbf{A}; \mathbf{a}_j)$*, when the pivot is a column of* $\mathbf{A}$*.*
- *For* $\mathbf{B} \in \mathbb{R}^{N \times N}$*, let* $(\mathcal{C}(\mathbf{A}; \mathbf{B}))_j \equiv \mathcal{C}(\mathbf{a}_j, \mathbf{b}_j)$*. That is, in case* $\mathbf{b}$ *in* (2.4) *depends on the column index* j *so that the pivot is allowed to change for each column, we supply the matrix* $\mathbf{B} \in \mathbb{R}^{N \times M}$ *to indicate that its* j*th column* $\mathbf{b}_j$ *will be used as the pivot for computing* ν_j*.*
- *Given* $\ell : \{1, \ldots, M\} \to \{1, \ldots, M\}$*, let* $(\mathcal{C}(\mathbf{A}; \ell))_j \equiv \mathcal{C}(\mathbf{a}_j; \mathbf{a}_{\ell(j)})$*. We define a pivot map* ℓ *that designates the pivot for each column, and supply it to* $\mathcal{C}$*.*
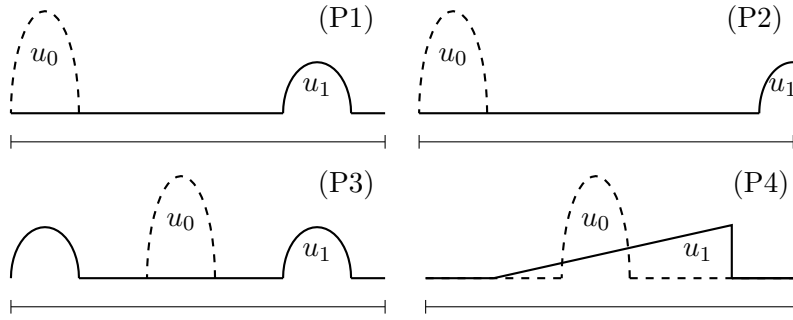
The shift numbers $\boldsymbol{\nu}$ contain the information on how many entries each column of the matrix should be shifted. So $\boldsymbol{\nu}$ describes a transport operation to be acted on each column, which will be summarized in the operator defined below.

Definition 2.3 (transport with periodic boundary conditions). *Given a matrix* $\mathbf{A} \in \mathbb{R}^{N \times M}$ *and a vector of shift numbers* $\boldsymbol{\nu} \in \mathbb{R}^M$*, the* transport with periodic boundary conditions $\mathcal{T}$ *is defined as*

$$(2.6) \qquad \mathcal{T}(\mathbf{A}; \boldsymbol{\nu}) \equiv \begin{bmatrix} \mathbf{K}^{\nu_1} & \mathbf{K}^{\nu_2} & \cdots & \mathbf{K}^{\nu_M} \end{bmatrix} \odot \begin{bmatrix} \mathbf{a}_1 & \mathbf{a}_2 & \cdots & \mathbf{a}_M \end{bmatrix}.$$

If a vector $\mathbf{b} \in \mathbb{R}^N$ *is given instead of a matrix, we let*

$$(2.7) \qquad \mathcal{T}(\mathbf{b}; \boldsymbol{\nu}) \equiv \begin{bmatrix} \mathbf{K}^{\nu_1}\mathbf{b} & \mathbf{K}^{\nu_2}\mathbf{b} & \cdots & \mathbf{K}^{\nu_M}\mathbf{b} \end{bmatrix}.$$

Figure 1. *Illustration of solution behavior for the four hyperbolic problems* (P1), (P2), (P3), *and* (P4). u_0 *denotes the initial profile, drawn in dashed lines, and u_1 denotes the solution at some future time, in solid lines.*

It is easy to see that $\mathcal{T}(\cdot; \boldsymbol{\nu})$ and $\mathcal{T}(\cdot : -\boldsymbol{\nu})$ are exact inverses of each other. That is, for fixed $\boldsymbol{\nu}$,

$$\text{(2.8)} \qquad \mathcal{T}(\mathcal{T}(\mathbf{A}; -\boldsymbol{\nu}); \boldsymbol{\nu}) = \mathbf{A} \quad \text{for } \mathbf{A} \in \mathbb{R}^{N \times M}.$$

The key observation in the example above (2.3) is that the SVD of $\mathcal{T}(\mathbf{A}; -\boldsymbol{\nu})$ with $\boldsymbol{\nu} = \mathcal{C}(\mathbf{A}; \mathbf{a}_1)$ results in faster decay in singular values than that of $\mathbf{A}$. (The dynamics in $\mathcal{T}(\mathbf{A}; -\boldsymbol{\nu})$ represents the reduced dynamics $r(\tau)$ in symmetry reduction.) Therefore one approximates $\mathcal{T}(\mathbf{A}; -\boldsymbol{\nu})$ by a low-rank representation $\tilde{\mathbf{A}}$ via the usual truncation of rank-1 expansion. If we apply the forward transport to $\tilde{\mathbf{A}}$, that is, compute $\mathcal{T}(\tilde{\mathbf{A}}; \boldsymbol{\nu})$, it will be a better approximation of $\mathbf{A}$ compared to the direct low-rank approximation of $\mathbf{A}$. This idea has been illustrated also in [31]. The effectiveness of this approach, along with extensions, will be discussed further in section 3.

Template fitting in the simple manner described above works very well for snapshots produced by solving the constant coefficient advection equation with periodic boundary conditions, but is not sufficient to well represent snapshots generated by solving other hyperbolic equations. For more general wave propagation problems the solution at later times will not simply be a shifted version of the initial data. The shape of the waves may vary, and waves may leave the domain or reflect at a boundary rather than behaving periodically. Moreover for a hyperbolic system of equations there are different families of waves that propagate at different speeds and perhaps in different directions. Nonetheless, many important wave propagation problems exhibit coherent waves that modulate slowly relative to their propagation speed, which suggests that a more complex version of template fitting and reversal may still be very effective in reducing the dimension of the data.

We give four specific examples of the difficulties mentioned above that are typical of more general hyperbolic PDEs. Illustrations of the solution behavior for each problem are shown in Figure 1.

(P1) Advection equation with source term,

$$\text{(2.9)} \qquad u_t + u_x = -\gamma u \quad \text{in } (0, 1) \quad \text{with } \gamma > 0,$$

where $u(x, 0)$ is a nonnegative density pulse. The pulse diminishes in height over time,

and this decrease cannot be well represented by translation alone. This is an inherent limitation of (2.4).

(P2) Advection equation

$$u_t + u_x = 0 \quad \text{in } (0, 1) \tag{2.10}$$

with absorbing boundary conditions, $u_x = -u_t$ at the right boundary $x = 1$, and $u(x, 0)$ a density pulse. In (2.4) periodic shift $\mathbf{K}$ assumes periodic boundary conditions, so there is little hope of capturing this absorption.

(P3) Acoustic equations in a homogeneous medium,

$$\begin{bmatrix} p \\ u \end{bmatrix}_t + \begin{bmatrix} 0 & K_0 \\ 1/\rho_0 & 0 \end{bmatrix} \begin{bmatrix} p \\ u \end{bmatrix}_x = 0 \quad \text{in } (0, 1) \tag{2.11}$$

with periodic boundary conditions and the initial conditions in which $p(x, 0)$ is an acoustic pulse and $u(x, 0) = 0$. For the state variable p, the initial profile splits into two, both scaled by half, and propagates in opposite directions. A single minimization problem (2.4) cannot be used to represent the two different speeds.

(P4) Burgers' equations

$$u_t + u u_x = 0 \quad \text{in } (0, 1), \tag{2.12}$$

again with a density pulse as the initial condition. When the initial profile changes shape dramatically, translation alone cannot yield a good approximation.

In the next four subsections we present a series of enhancements that are designed to deal with these four common features of hyperbolic problems.

- *Projection of pivot.* In section 2.1 we allow the pivot vector to be both translated, as in section 2, and rescaled in order to better match a snapshot. This can be done via a projection.
- *Cut-off vectors.* In section 2.2 we introduce the idea of cutting off a section of the (translated and projected) pivot vector, necessary to handle outflow boundary conditions and also to better model shock formation.
- *Greedy iteration and pivoting.* In section 2.3 we address the fact that a single pivot vector (even after translation, projection, and cut-off) is not likely to capture all of the features of later snapshots. We thus introduce an iterative algorithm that allows fitting with a combination of several different versions of the pivot vector (with different choices of the translation, projection, and cut-off). We extend further to allow the choice of a new pivot vector once the point of diminishing returns has been reached.
- *Regularization of shift numbers.* In section 2.4 we suggest an improved way of choosing the shift numbers to better reflect the expectation that waves should propagate smoothly in time, which can help to better reveal the wave propagation structure of the solution and may be important for applications of the reduced model.

Algorithm 1 summarizes the final iterative algorithm. Section 2 concludes with some numerical experiments illustrating the effectiveness of this algorithm on the acoustics and Burgers' equations. Some further extensions and improvements are then pursued in section 3.

2.1. Projection of pivot. Consider the situation in the problem (P1) above, where the initial profile diminishes in height with time, while being transported at constant speed. This is illustrated in Figure 1 (P1). The minimization problem in (2.4) does not take the scaling into account. We introduce a scaling by projecting the jth column onto the pivot. The projection $\mathcal{P}$ is defined as

$$(2.13) \qquad \mathcal{P}(\mathbf{a}_j; \mathbf{b}) \equiv \begin{cases} \dfrac{\mathbf{b}\mathbf{b}^T}{\|\mathbf{b}\|^2}\mathbf{a}_j & \text{if } \|\mathbf{b}\| > 0, \\ \mathbf{0} & \text{otherwise, if } \|\mathbf{b}\| = 0. \end{cases}$$

Now, we replace the functional in the minimization problem (2.4) by measuring the difference between the jth column $\mathbf{a}_j$ and the transported-and-projected vector $\mathcal{P}(\mathbf{a}_j; \mathbf{K}^\omega \mathbf{b})$. That is, we solve the minimization problem

$$(2.14) \qquad \nu_j = \mathrm{argmin}_{\omega \in \mathbb{Z}_N} \|\mathbf{a}_j - \mathcal{P}(\mathbf{a}_j; \mathbf{K}^\omega \mathbf{b})\|_2^2.$$

We denote this computation of the shift numbers in a concise form,

$$(2.15) \qquad \boldsymbol{\nu} = \mathcal{C}(\mathbf{A}; \mathbf{b}, \mathcal{P}),$$

by supplying the projection map $\mathcal{P}$. The scaling (2.13) must also be stored, and we organize it in the vector $\mathbf{h}$,

$$(2.16) \qquad h_j = \mathcal{P}(\mathbf{a}_j; \mathbf{K}^{\nu_j}\mathbf{b}) \quad \text{and} \quad \mathbf{h} = \begin{bmatrix} h_1 & \cdots & h_M \end{bmatrix}.$$

We denote this concisely by writing $\mathbf{h} = \mathcal{P}(\mathbf{A}; \mathbf{b}, \boldsymbol{\nu})$.

The operator $\mathcal{T}$ must also be generalized to take into account the scaling $\mathbf{h}$,

$$(2.17) \qquad \mathcal{T}(\mathbf{A}; \boldsymbol{\nu}, \mathbf{h}) \equiv \begin{bmatrix} h_1\mathbf{K}^{\nu_1} & \cdots & h_M\mathbf{K}^{\nu_M} \end{bmatrix} \odot \begin{bmatrix} \mathbf{a}_1 & \cdots & \mathbf{a}_M \end{bmatrix}.$$

The vector version of $\mathcal{T}$ is modified similarly,

$$(2.18) \qquad \mathcal{T}(\mathbf{b}; \boldsymbol{\nu}, \mathbf{h}) \equiv \begin{bmatrix} h_1\mathbf{K}^{\nu_1}\mathbf{b} & h_2\mathbf{K}^{\nu_2}\mathbf{b} & \cdots & h_M\mathbf{K}^{\nu_M}\mathbf{b} \end{bmatrix}.$$

2.2. Cut-off vectors. In (P2) we encounter a wave profile that is absorbed at the right boundary. If we were to apply the minimization problem with projection (2.14), the vanishing pulse would be partly represented by a translating profile that is decreasing in height. Still, some part of the profile will hit the boundary to the right, and since $\mathbf{K}$ assumes a periodic boundary condition, the profile will appear at the left boundary as well. While one can modify $\mathbf{K}$ to account for this behavior, this will cause significant changes in the minimization problem (2.4), and the existence of the exact inverse in (2.8) may be lost. Moreover, another problem arises in the Burgers' equation (P4). The initial pulse is deformed to the extent that its transported profile may not represent the shape of the shock wave adequately, even with scaling.

As a step toward remedying both issues, we introduce the cut-off (or support) vector $\boldsymbol{\rho}$. Roughly speaking, $\boldsymbol{\rho}$ will designate the location of the domain where the projection $\mathcal{P}(\mathbf{a}_j; \mathbf{K}^\omega \mathbf{b})$, for given $\omega \in \mathbb{Z}_N$, is a good approximation of $\mathbf{a}_j$. We will denote by $\mathcal{S}$ the

operator that yields the cut-off, for given two column vectors $\mathbf{a}_j$ and $\mathbf{b}$. To be more specific, the ith component ρ_i of $\boldsymbol{\rho}$ is defined as

$$(2.19) \qquad \rho_i = (\mathcal{S}(\mathbf{a}_j; \mathbf{b}))_i \equiv \begin{cases} 1 & \text{if } \text{sign}(a_{ij} - b_i) \cdot \text{sign}(a_{ij}) \geq 0 \\ & \qquad \text{and } |a_{ij} - b_i| \leq |a_{ij}|, \\ 0 & \text{otherwise.} \end{cases}$$

The intention is to use $\rho_i b_i$ to approximate a_i. The first condition $\text{sign}(a_{ij} - b_i) \cdot \text{sign}(a_{ij}) \geq 0$ ensures that the cut-off pivot does not overshoot the profile, and the second condition $|a_{ij} - b_i| \leq |a_{ij}|$ makes sure that the approximation has the same sign as the original vector.

We will project $\mathbf{a}_j$ onto $\mathbf{b}$ for scaling before we apply $\mathcal{S}$. That is, $\mathcal{P}(\mathbf{a}_j; \mathbf{b})$ will be input above in (2.19) in place of $\mathbf{b}$. To simplify the notation, we will use the shorthand

$$(2.20) \qquad \mathcal{S}(\mathbf{a}_j; \mathbf{b}, \mathcal{P}) \equiv \mathcal{S}(\mathbf{a}_j; \mathcal{P}(\mathbf{a}_j; \mathbf{b})).$$

Now the minimization (2.14) is further updated: we shift $\mathbf{b}$ and scale it using the projection $\mathcal{P}$, we cut-off using $\mathcal{S}$, and then we compare with $\mathbf{a}_j$. The new minimization problem becomes

$$(2.21) \qquad \nu_j = \text{argmin}_{\omega \in \mathbb{Z}_N} \|\mathbf{a}_j - \mathcal{S}(\mathbf{a}_j; \mathbf{K}^\omega \mathbf{b}, \mathcal{P}) \odot \mathcal{P}(\mathbf{a}_j; \mathbf{K}^\omega \mathbf{b})\|_2^2.$$

Here $\odot$ denotes the componentwise multiplication between two vectors in $\mathbb{R}^N$. As before, we define a shorter notation for this computation of shift numbers,

$$(2.22) \qquad \boldsymbol{\nu} = \mathcal{C}(\mathbf{A}; \mathbf{b}, \mathcal{P}, \mathcal{S}).$$

Furthermore, this generalization makes it necessary to store the vectors $\boldsymbol{\rho}_j$ corresponding to each ν_j, that is,

$$(2.23) \qquad \boldsymbol{\rho}_j = \mathcal{S}(\mathbf{a}_j; \mathbf{K}^{\nu_j} \mathbf{b}, \mathcal{P}).$$

We store these as columns of the matrix

$$(2.24) \qquad \mathbf{P} = \begin{bmatrix} \boldsymbol{\rho}_1 & \cdots & \boldsymbol{\rho}_M \end{bmatrix},$$

and we also write $\mathbf{P} = \mathcal{S}(\mathbf{A}; \mathbf{b}, \boldsymbol{\nu}, \mathcal{P})$. In implementing the algorithm $\mathbf{P}$ is computed simultaneously with $\boldsymbol{\nu}$, but we will keep this implicit notation. $\mathbf{P}$ can be stored as an array of Boolean data-type, so the storage requirement is not significant. It can be even more reduced should the pivot $\mathbf{b}$ be sparse, but the details will not be pursued here.

Finally, the transport operator $\mathcal{T}$ must also be extended to incorporate the cut-off function, which we can do by letting

$$(2.25) \qquad \begin{aligned} \mathcal{T}(\mathbf{A}; \boldsymbol{\nu}, \mathbf{h}, \mathbf{P}) &\equiv \begin{bmatrix} h_1 \boldsymbol{\rho}_1 \odot \mathbf{K}^{\nu_1} & \cdots & h_M \boldsymbol{\rho}_M \odot \mathbf{K}^{\nu_M} \end{bmatrix} \odot \begin{bmatrix} \mathbf{a}_1 & \cdots & \mathbf{a}_M \end{bmatrix} \\ &= \begin{bmatrix} h_1 \boldsymbol{\rho}_1 \odot \mathbf{K}^{\nu_1} \mathbf{a}_1 & \cdots & h_M \boldsymbol{\rho}_M \odot \mathbf{K}^{\nu_M} \mathbf{a}_M \end{bmatrix}, \end{aligned}$$

and the vector version $\mathcal{T}(\mathbf{b}; \boldsymbol{\nu}, \mathbf{h}, \mathbf{P})$ is defined similarly. Now we are ready to combine these computations in a greedy iteration.

2.3. Greedy iteration and pivoting. Recall that the rank-1 expansion arising from the SVD can be seen as an iterative procedure in which a greedy rank-1 update is made in each iteration. Here we define a similar update for the minimizations above, by attempting to capture transport structure iteratively. The necessity of multiple iterations can be illustrated by the acoustic equation (P3). The initial pulse splits into two and travels at two different speeds. In this case the speeds have equal magnitude with opposite sign, as sketched in Figure 1 (P3). However, they could be of the same sign and may also vary with time. One minimization problem using $\mathcal{C}$ and $\mathcal{T}$ defined above cannot approximate this behavior adequately. Furthermore, in the Burgers' equation (P4) the profile is deformed heavily, so that transporting one pivot once, even with projections and cut-offs, cannot capture the substantial change in shape.

Therefore, we iterate on the previously defined computations as follows. First, let $\mathbf{R}_1 \equiv \mathbf{A}$ and choose a pivot $\mathbf{b}_1$, say, the first column $\mathbf{a}_1$ of $\mathbf{A}$. $\mathbf{R}_k$ will denote the residual, and index k will be used for the iteration number. We compute the shift numbers $\boldsymbol{\nu}_1$, the scaling $\mathbf{h}_1$, and the cut-offs $\mathbf{P}_1$,

$$(2.26) \qquad \boldsymbol{\nu}_1 \equiv \mathcal{C}(\mathbf{R}_1; \mathbf{b}_1, \mathcal{P}, \mathcal{S}), \quad \mathbf{h}_1 \equiv \mathcal{P}(\mathbf{R}_1; \mathbf{b}_1, \boldsymbol{\nu}_1), \quad \text{and} \quad \mathbf{P}_1 \equiv \mathcal{S}(\mathbf{R}_1; \mathbf{b}_1, \boldsymbol{\nu}, \mathcal{P}).$$

Now, we subtract off the first rough approximation from the snapshots,

$$(2.27) \qquad \mathbf{R}_2 \equiv \mathbf{R}_1 - \mathcal{T}(\mathbf{b}_1; \boldsymbol{\nu}_1, \mathbf{h}_1, \mathbf{P}_1).$$

This forms one iteration. We remark that the conditions in (2.19) prevent $\mathbf{R}_2$ from developing oscillations.

Next, we compute $\boldsymbol{\nu}_2, \mathbf{h}_2$, and $\mathbf{P}_2$ by replacing $\mathbf{R}_1$ above by $\mathbf{R}_2$. We repeat, so that all transport patterns using the pivot $\mathbf{b}_1$ are removed from the data. This reaches a point of diminishing returns after some iterations, and we monitor the progress at the kth iteration by computing the ratio $\|\mathbf{R}_k\|/\|\mathbf{R}_{k-1}\|$. One may set a threshold τ_1 so that

$$(2.28) \qquad \text{if } \frac{\|\mathbf{R}_k\|_F}{\|\mathbf{R}_{k-1}\|_F} > \tau_1, \text{ then update the pivot } \mathbf{b}_\ell \text{ to } \mathbf{b}_{\ell+1}.$$

There are many different options in choosing the next pivot $\mathbf{b}_{\ell+1}$. For example, one may proceed to a pivot that is orthogonal to the previous pivot. Here we simply choose $\mathbf{b}_\ell = \mathbf{r}_{\ell,k}$, where $\mathbf{r}_{\ell,k}$ is the ℓth column of $\mathbf{R}_k$.

The algorithm halts when $\|\mathbf{R}_k\|_F < \tau_0$ for a given tolerance τ_0.

Let us organize the shift numbers $\boldsymbol{\nu}_k$ at each iteration in $\mathbf{V}$,

$$(2.29) \qquad \mathbf{V} \equiv \begin{bmatrix} \boldsymbol{\nu}_1 & \cdots & \boldsymbol{\nu}_K \end{bmatrix},$$

where K denotes the index of the last iteration. We do the same for the cut-offs $\mathbf{P}_k$ and collect them in $\mathbf{Q}$,

$$(2.30) \qquad \mathbf{Q} \equiv \begin{bmatrix} \mathbf{P}_1 & \cdots & \mathbf{P}_K \end{bmatrix},$$

and then similarly collect $\mathbf{h}_k$ in $\mathbf{H}$,

$$(2.31) \qquad \mathbf{H} \equiv \begin{bmatrix} \mathbf{h}_1 & \cdots & \mathbf{h}_K \end{bmatrix}.$$

2.4. Regularization of shift numbers. The shift numbers $\boldsymbol{\nu}_k$ encode the transport motion of a profile over time, and we expect the speed of the transport to be relatively smooth. While the greedy iteration may yield a good approximation to $\mathbf{A}$, the components of $\boldsymbol{\nu}_k$ may vary wildly. Hence it is reasonable to enforce some regularity when computing $\boldsymbol{\nu}_k$. That is, the shift number should change smoothly over time. We achieve this by adding a penalty term in the minimization problem (2.21) when $j > 1$ to try to keep $|\nu_{(j-1)k} - \nu_{jk}|$ small:

$$(2.32) \qquad \nu_{jk} = \operatorname{argmin}_{\omega \in \mathbb{Z}_N} \|\mathbf{a}_j - \mathcal{S}(\mathbf{a}_j; \mathbf{K}^\omega \mathbf{b}, \mathcal{P}) \odot \mathcal{P}(\mathbf{a}_j; \mathbf{K}^\omega \mathbf{b})\|_2^2 + \lambda \left|\omega - \nu_{(j-1)k}\right|^2$$

with a regularization parameter λ. It may be desirable to add additional higher-order regularity terms, that is, second order finite difference term for $\boldsymbol{\nu}_k$. Other penalty terms regarding the regularity of $\mathbf{P}_k$ as well as $\mathbf{h}_k$ can be summed into (2.32) also.

The regularization is crucial, since the smooth evolution of the shift numbers across snapshots is needed for *displacement interpolation* in the sense used in optimal transport (see, e.g., [40]), which effectively tracks the transport structure. Note also that we may encode the smooth evolution efficiently by polynomial interpolation or regression. This could be taken into consideration much earlier on, when the snapshots are generated: one may store snapshots at Chebyshev grid points in the time variable, to facilitate accurate interpolation, and then enforce high regularity in the shift numbers.

The output of the regularized version of the algorithm can also be viewed as a greedy solution to an optimal transport problem, where one seeks to minimize the cost of transporting an initial state to the final state over admissible transport maps. The cost function here is particularly simple and is given by the regularization terms, for example, the term penalizing the total displacement in the case of (2.32).

A simplified pseudocode of the transport reversal is given in Algorithm 1.

2.5. Numerical example for transport reversal. Here we apply the transport reversal algorithm to two of the problematic scenarios given above: the acoustic equation (P3) and the Burgers' equation (P4). The tolerances and regularization parameters are chosen rather heuristically. λ is set adaptively according to the variation of the functional in the minimization problem without the penalty terms (2.21): we set λ in (2.32) as $2.5/(CN)$, where

$$(2.33) \qquad \begin{aligned} C \equiv &\max_{\omega \in \mathbb{Z}_N} \|\mathbf{a}_j - \mathcal{S}(\mathbf{a}_j; \mathbf{K}^\omega \mathbf{b}, \mathcal{P}) \odot \mathcal{P}(\mathbf{a}_j; \mathbf{K}^\omega \mathbf{b})\|_2^2 \\ &- \min_{\gamma \in \mathbb{Z}_N} \|\mathbf{a}_j - \mathcal{S}(\mathbf{a}_j; \mathbf{K}^\gamma \mathbf{b}, \mathcal{P}) \odot \mathcal{P}(\mathbf{a}_j; \mathbf{K}^\gamma \mathbf{b})\|_2^2. \end{aligned}$$

The L^2-norm used for measuring the error here refers to the two-dimensional L^2-norm over spatial and temporal variables, $\|\cdot\|_F/\sqrt{NM}$ for a matrix in $\mathbb{R}^{N \times M}$.

2.5.1. Acoustic equation. The snapshot of the p variable for the acoustic equation is given in Figure 2. No pivoting was required up to the given maximum number of iterations $K = 15$. The L^2-norm of the residual at the final iteration was 2.1841×10^{-3}. The corresponding shift numbers for each iteration are shown in Figure 3. The diagonal pattern is clearly visible, which indicates that the method is capturing the two profiles being transported at constant speed through the domain. However, there is some ambiguity between the two profiles when they pass each other near snapshot number 60. Notice how in the computed shift numbers for

---

Algorithm 1 Transport reversal algorithm.

1: **procedure** $\mathrm{TR}(\mathbf{A}, K, \tau_0, \tau_1)$
                   ▷ input matrix $\mathbf{A}$, max. no. of iterations K, and tolerances τ_0, τ_1
2:   $\ell \leftarrow 1$ ▷ pivot number
3:   $k \leftarrow 0$ ▷ iteration count
4:   $r_{\mathrm{old}} \leftarrow \|\mathbf{A}\|_F$
5:   $\mathbf{R} \leftarrow \mathbf{A}$ ▷ initialize residual $\mathbf{R}$
6:   $\mathbf{b}_\ell \leftarrow \mathbf{R}(:,\ell)$ ▷ choose first column of $\mathbf{R}$ as pivot
7:   **while** $(r_{\mathrm{old}} > \tau_0 \text{ and } k \le K)$ **do**
8:     $k \leftarrow k + 1$
9:     $(\boldsymbol{\nu}_k, \mathbf{h}_k, \mathbf{P}_k) \leftarrow (\mathcal{C}(\mathbf{R}; \mathbf{b}_\ell, \mathcal{P}, \mathcal{S}), \mathcal{P}(\mathbf{R}; \mathbf{b}_\ell, \boldsymbol{\nu}_k), \mathcal{S}(\mathbf{R}; \mathbf{b}_\ell, \boldsymbol{\nu}_k, \mathcal{P}))$
                        ▷ computation is done concurrently
10:     $\mathbf{R} \leftarrow \mathbf{R} - \mathcal{T}(\mathbf{b}_\ell; \boldsymbol{\nu}_k, \mathbf{h}_k, \mathbf{P}_k)$
11:     $r_{\mathrm{new}} \leftarrow \|\mathbf{R}\|_F$
12:     **if** $r_{\mathrm{new}}/r_{\mathrm{old}} > \tau_1$ **then**
13:       $\ell \leftarrow \ell + 1$ ▷ pivoting
14:       $\mathbf{b}_\ell \leftarrow \mathbf{R}(:,\ell)$ ▷ update pivot to be the ℓth column of the new $\mathbf{R}$
15:     **end if**
16:     $r_{\mathrm{old}} \leftarrow r_{\mathrm{new}}$
17:   **end while**
18:   $\mathbf{B} \leftarrow [\mathbf{b}_1, \cdots, \mathbf{b}_\ell]$
19:   $\mathbf{V} \leftarrow [\boldsymbol{\nu}_1, \cdots, \boldsymbol{\nu}_k]$
20:   $\mathbf{H} \leftarrow [\mathbf{h}_1, \cdots, \mathbf{h}_k]$
21:   $\mathbf{Q} \leftarrow [\mathbf{P}_1, \cdots, \mathbf{P}_k]$
22:   **return** $\mathbf{B}, \mathbf{V}, \mathbf{H}, \mathbf{Q}$
                  ▷ output pivots, shift numbers, scalings, and cut-offs
23: **end procedure**

$k = 1$ shown in Figure 3 the profile is transported first to the right, and then the direction is reversed around the snapshot 60, rather than keeping straight. This behavior can be changed by adding higher-order finite difference terms of $\boldsymbol{\nu}_k$ as penalty term in (2.32) so that the second derivative of the shift numbers are kept small.

2.5.2. Burgers' equation. Now we apply transport reversal to the snapshot matrix from the Burgers' equation (P4). The initial condition and its final snapshot is shown in Figure 4. The entire snapshot matrix and its approximate reconstruction are shown in Figure 5. The total number of iterations was $K = 30$ and pivoting occured once at iteration 19.

The L^2-norm of the residual at the final iteration was 9.6333×10^{-5}. The corresponding shift numbers extracted for iterations 1–15 are shown in Figure 6. The shift numbers computed here also face an ambiguity between the two separate humps at times, in a similar manner to the acoustic equations example. Adding more regularity will remove this ambiguity.

The first two shift numbers in Figure 6 correspond to the movement of the pivot that attempts to match the deforming hump to the left and to the right. Since the left and the
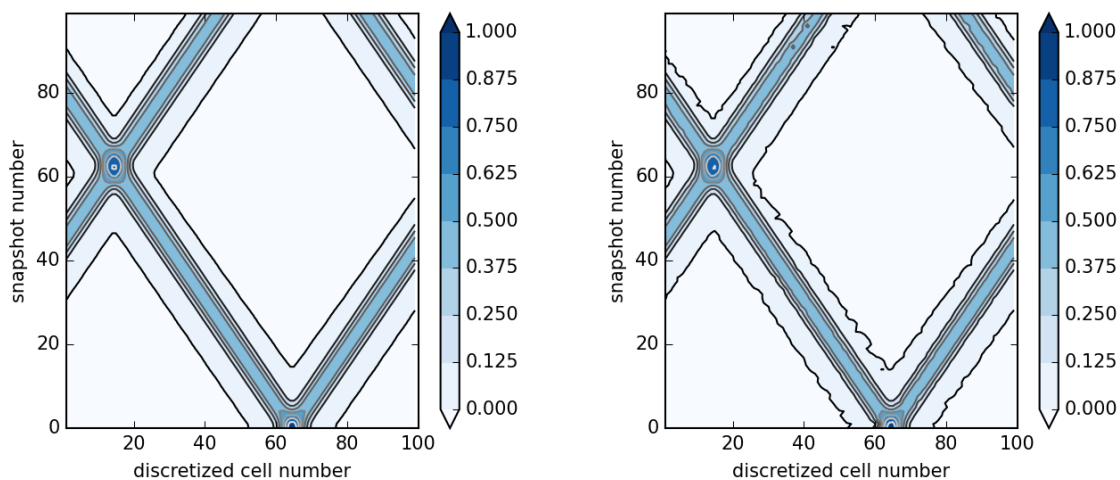
Figure 2. *Snapshot matrix of the p variable in the acoustic equation* (P3) *(left) and its approximation via the transport reversal algorithm (right). The L^2-norm of the difference is 2.1841×10^{-3}.*
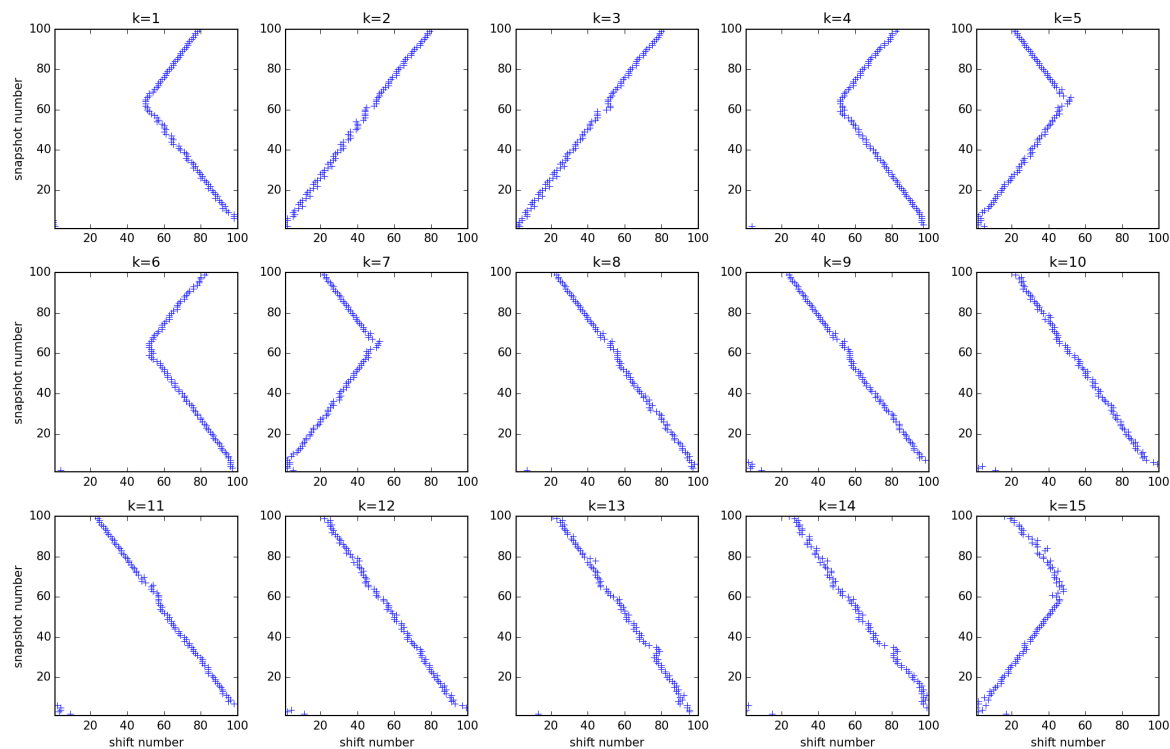
Figure 3. *Shift numbers ν_k* (2.29) *for each iteration k, for the acoustic equations example. Single pivot (the initial condition) was used.*
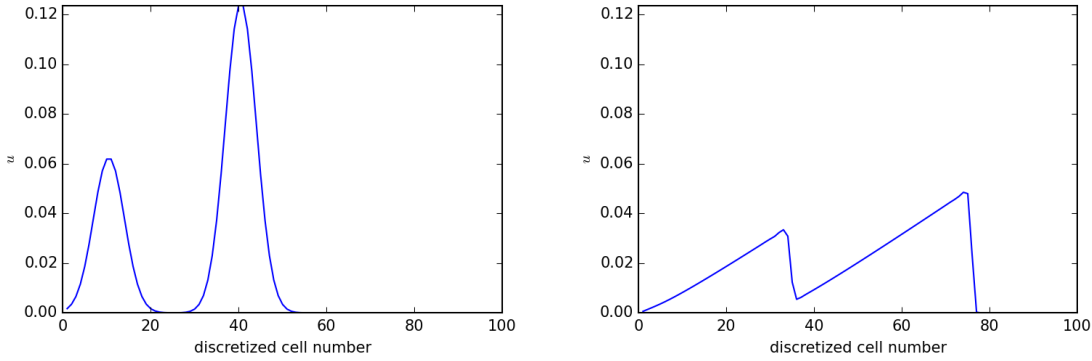
Figure 4. *Initial condition of the Burgers' equation* (P4) *(left) and its final snapshot (right).*
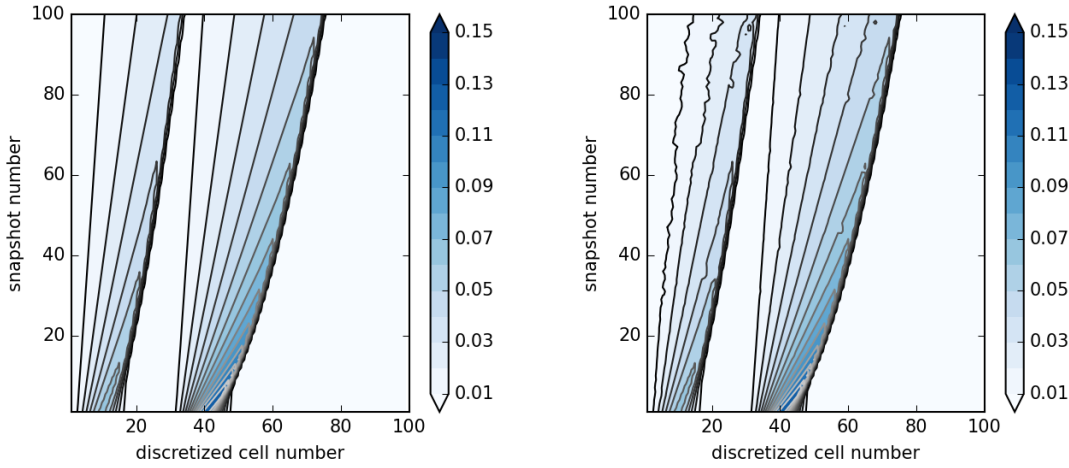

Figure 5. *Snapshot matrix of the Burgers' equation* (P4) *(left) and its approximation via the transport reversal algorithm (right). L^2-norm of the difference is 9.6333×10^{-5}.*

right humps move at slightly different speeds, transporting the initial condition at constant speed is only able to match one of them. It is helpful to isolate the contributions from the first two iterations from the algorithm. That is, we observe

$$(2.34) \qquad\qquad h_{jk}\boldsymbol{\rho}_{jk} \odot \mathbf{K}^{\nu_{jk}}\mathbf{a}_1 \quad \text{ for } k = 1, 2 \text{ and } j = 15, 50.$$

These contributions are shown in Figure 7. Note how in the first iteration the initial profile is cut off to match the hump to the right. In the second iteration, the hump to the right in the initial condition is cut off to match the left shock in the snapshot. This illustrates the flexibility provided by the cut-off vectors $\boldsymbol{\rho}$ for capturing the deformation occurring in the profile.

3. Extensions of the shift operator. In this section, we consider extensions of the shift matrix $\mathbf{K}$ defined in (1.4). This matrix is the discrete analogue of the translation operator
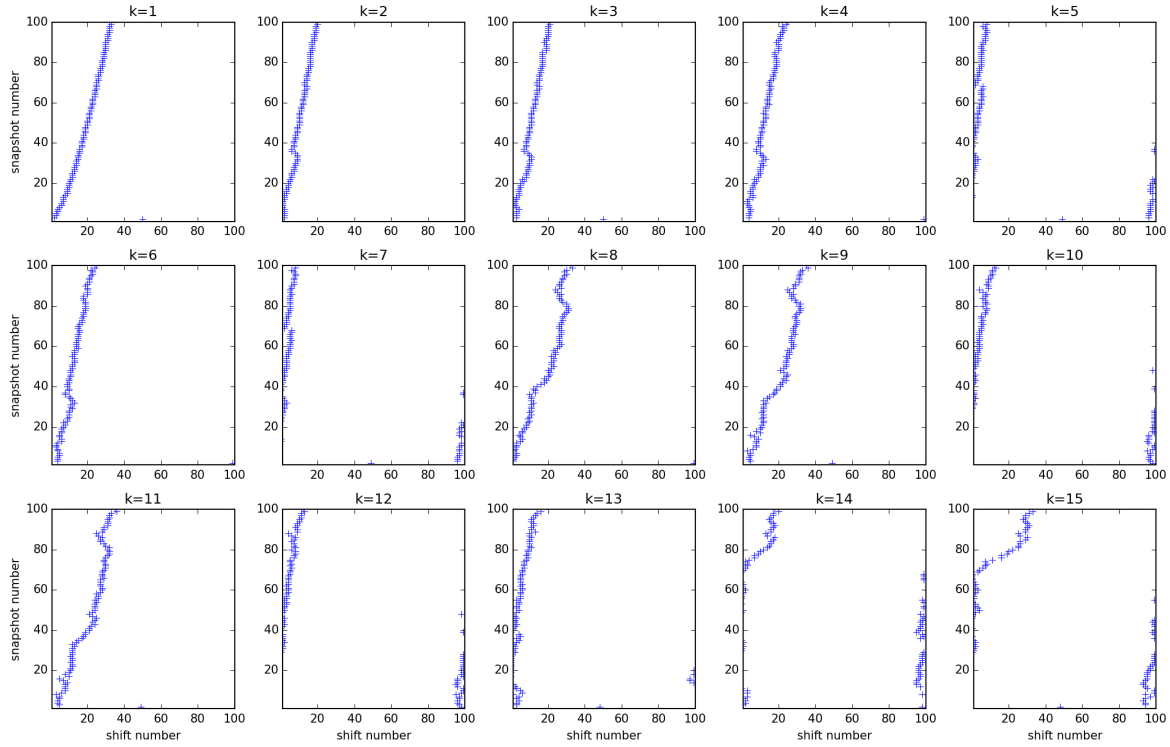
Figure 6. *Shift numbers ν_k (2.29) for iterations $k = 1$ to 15, for the Burgers' equations example. The first pivot (the initial condition) was used for all iterations shown here.*

in (1.7) and is the defining component of the minimization problem (2.4) that enables the detection of moving waves. Throughout this section, we consider the natural extensions of the powers $\mathbf{K}^\omega$ ($\omega \in \mathbb{Z}_N$) that were used repeatedly in the the minimization problems (2.4), (2.14), (2.21), and (2.32) in the previous section.

First, we extend $\mathbf{K}^\omega$, for which ω was restricted to a set of integers $\mathbb{Z}_N$, to real numbers $\tilde{\omega} \in \mathbb{R}$. This is done using a linear interpolation between $\mathbf{K}^\omega$ and $\mathbf{K}^{\omega+1}$, for the integer $\omega = \lfloor \tilde{\omega} \rfloor$. This yields a continuous operator $\mathcal{K}(\tilde{\omega})$ defined over the real numbers. This operator can be used in place of $\mathbf{K}^\omega$ in the minimization problems of the previous section, and therefore transport reversal can be extended to $\mathbb{R}$. Since $\mathcal{K}(\tilde{\omega})$ now causes some numerical diffusion due to its interpolation, a reconstruction procedure is introduced to sharpen the profile. This extension is developed in sections 3.1–3.3.

Another extension allows the advection velocity c in the advection equation (1.1) to depend on the spatial variable. This extension results in the operator $\mathcal{K}_c(\tilde{\omega})$ with a prescribed velocity field c. To avoid excessive diffusion whenever $\mathcal{K}_c(\tilde{\omega})$ is used, we apply the large-time step (LTS) method. We also briefly discuss the pivoting procedure that becomes necessary for systems of PDEs. This appears in section 3.4

We emphasize that the extensions that appear in this section can be used in the greedy iteration introduced in the previous section. This would require only that one change the operator $\mathbf{K}$ above to $\mathcal{K}$ or $\mathcal{K}_c$.
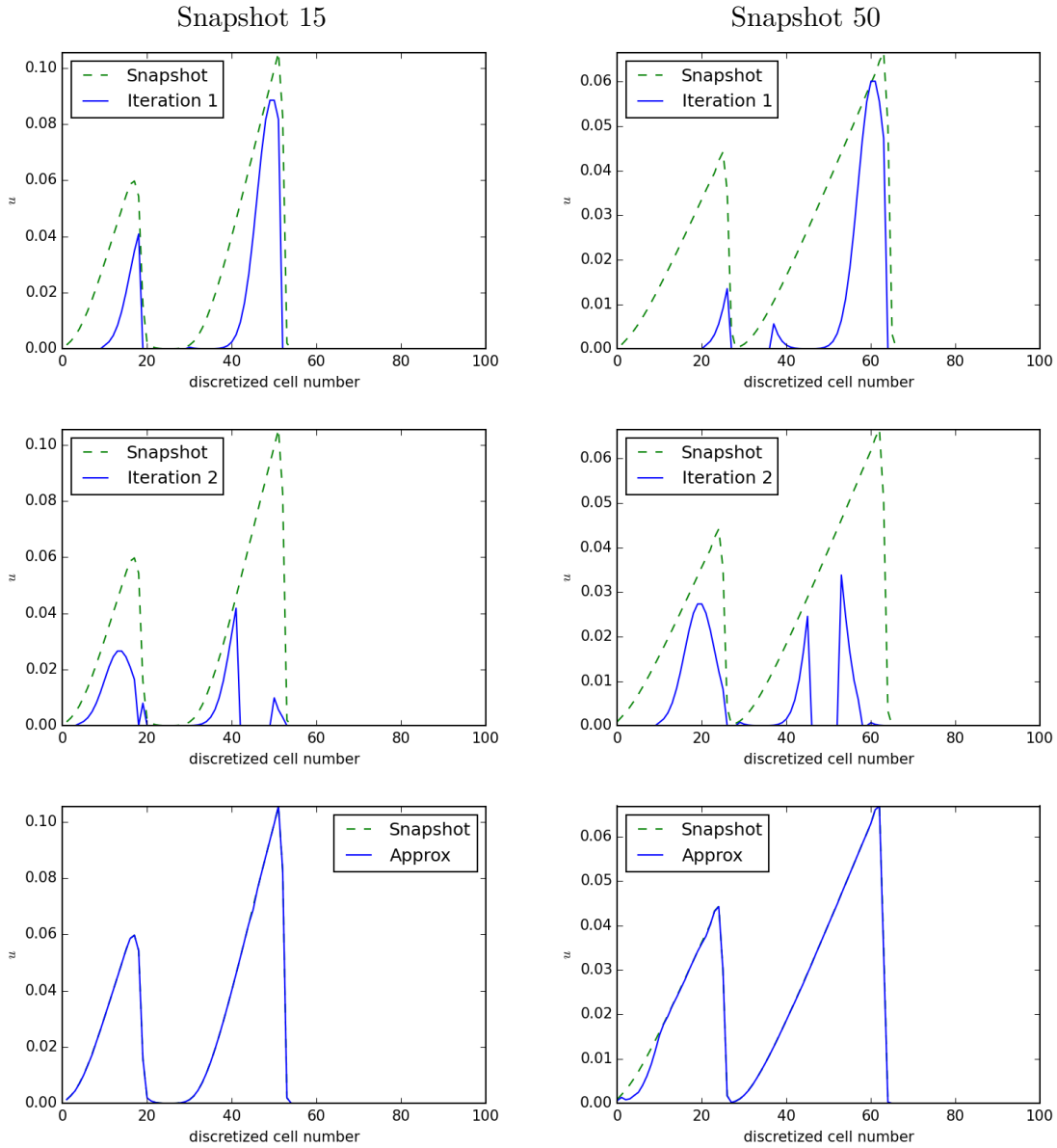
Figure 7. *Contributions from the first two iterations of the transport reversal (rows 1 and 2) for snapshots 15 and 50 (columns 1 and 2) given by the expression (2.34). Final approximation at iteration 30 is given in the last row. The contribution from first iteration alone attempts to capture the taller shock to the right (row 1), and that of the second iteration captures the shorter one to the left (row 2).*

3.1. Extension of $\mathbf{K}^\omega$ by upwind flux. First let us recall the finite volume upwind flux, which will motivate our definitions below. The finite volume update of the advection equation (1.1) is given by

$$(3.1) \qquad u_j^{n+1} = u_j^n - \frac{\Delta t}{\Delta x} \left(f_{j+1/2}^n - f_{j-1/2}^n \right).$$

The upwind flux is defined by $f_{j-1/2}^n \equiv c u_{j-1}^n$, and letting $\nu \equiv c\Delta t / \Delta x$ be the *shift number*, the time-step (3.1) can be is expressed as a linear interpolation between u_{j-1}^n and u_j^n,

$$(3.2) \qquad u_j^{n+1} = \left(\frac{c\Delta t}{\Delta x} \right) u_{j-1}^n + \left(1 - \frac{c\Delta t}{\Delta x} \right) u_j^n = \nu u_{j-1}^n + (1 - \nu) \, u_j^n.$$

We write the update in (3.1) as a matrix multiplication.

Definition 3.1. *Let* $\mathbf{K}$ *be the permutation matrix in* (1.4). *Define the matrix* $\mathbf{K}(\nu) \equiv (1 - \nu)\mathbf{I} + \nu\mathbf{K}$. *Let us also define the* discretized Laplacian $\mathbf{L}_h \equiv (\mathbf{K} + \mathbf{K}^T - 2\mathbf{I})/h^2$, *where* $h = 1/N$.

We list some basic properties of the matrix $\mathbf{K}(\nu)$.

Lemma 3.2. $\mathbf{K}(\nu)$ *satisfies*
(a) $\mathbf{K}(\nu)\mathbf{K}(\omega) = \mathbf{K}(\omega)\mathbf{K}(\nu)$.
(b) $\mathbf{K}(\nu)\mathbf{K}(\omega)^T = \mathbf{K}(\omega)^T \mathbf{K}(\nu)$.
(c) *For* $0 \leq \nu, \omega \leq 1$ *and* $\nu + \omega \leq 1$, $\mathbf{K}(\nu)\mathbf{K}(\omega) = \mathbf{K}(\nu + \omega) + \mathcal{O}(1/N^2)$, *where the constant for* $\mathcal{O}(1/N^2)$ *is a shifted discrete Laplacian (see the paragraph below).*
(d) *For* $\mathbf{u} \in \mathbb{R}^N$, $\sum_{j=1}^N (\mathbf{u})_j = \sum_{j=1}^N (\mathbf{K}(\nu)\mathbf{u})_j$.

Proof. The proof follows directly from definitions and is given in Appendix A. ∎

The notation $\mathcal{O}(1/N^2)$ here and throughout the paper is to be interpreted as follows. When a matrix term is to be acted on vectors $\mathbf{v}$ that are discretizations of twice differentiable functions on a grid of size $\mathcal{O}(N)$ (so that its discrete Laplacian $\mathbf{L}_h\mathbf{v}$ converges), the resulting term is of size $\mathcal{O}(1/N^2)$.

Note that $\mathbf{K}(\nu)^T$ is obtained if we use the one-sided flux in (3.1) on advection with velocity -1 rather than 1 (so that this is actually still the upwind flux). Naturally, when the matrix $\mathbf{K}(\nu)^T\mathbf{K}(\nu)$ is multiplied to the left of a vector, it propagates the entries of the vector first in one direction and then back in the opposite direction. The resulting vector should be close to the initial one, in other words $\mathbf{K}(\nu)^T\mathbf{K}(\nu)$ must be close to the identity. This fact is summarized in the following lemma.

Lemma 3.3. $\mathbf{K}(\nu)^T\mathbf{K}(\nu) = \mathbf{K}(\nu)\mathbf{K}(\nu)^T$ *is an identity up to* $\mathcal{O}(1/N^2)$, *in which the residual is a multiple of discrete Laplacian with periodic boundary condition. It satisfies the bound*

$$(3.3) \qquad \left\| \mathbf{K}(\nu)^T\mathbf{K}(\nu) - \mathbf{I} \right\|_2 \leq 4\nu(1 - \nu).$$

Proof. By definition and recalling $h = 1/N$,

$$(3.4) \qquad \mathbf{K}(\nu)^T\mathbf{K}(\nu) - \mathbf{I} = \nu(1 - \nu)\left(\mathbf{K} + \mathbf{K}^T - 2\mathbf{I} \right) = \frac{\nu(1 - \nu)}{N^2}\mathbf{L}_h.$$

Now, $\|\mathbf{K} + \mathbf{K}^T - 2\mathbf{I}\|_2 \leq 4$ by von Neumann analysis

$$|\lambda_\xi| = \left| e^{i2\pi\xi(x+h)} - 2e^{i2\pi\xi x} + e^{i2\pi\xi(x-h)} \right|$$
$$= \left| e^{i2\pi\xi x} \left(e^{i2\pi\xi h} - 2 + e^{-i2\pi\xi h} \right) \right| = 2\left| (\cos(2\pi\xi h) - 1) \right| \leq 4. \qquad ∎$$

The inequality (3.3) holds for any N, and it is merely an estimate for the total numerical diffusion due to the upwind flux, resulting from both $\mathbf{K}(\nu)$ and $\mathbf{K}(\nu)^T$. This marks a point of departure from the continuous setting studied in [32, 33], as it indicates that the translational actions discretized in such a way no longer form a group; the inverse (3.4) and multiplication (Lemmas 3.2(c) and 3.5(c)) are both only approximate and their residuals indicate the presence of numerical diffusion.

Note that $\mathbf{K}(\nu)$ was defined in Definition 3.1 for any $\nu \in \mathbb{R}$. However, when ν is viewed as the Courant number, the Courant–Friedrichs–Lewy condition imposes a necessary condition for stability of the upwind method [23], which requires $\nu \in [0, 1]$ in this case. This prohibits the use of $\mathbf{K}(\nu)$ when ν is outside of the unit interval, at least superficially. There is a straightforward generalization of this ν to be any real number $\tilde{\nu} \in \mathbb{R}$ by first shifting exactly an integer number of times (determined by the integer part s of $\tilde{\nu}$), which is accomplished by multiplying by $\mathbf{K}^s$, and then applying $\mathbf{K}(\nu)$, where ν is the remaining fractional part.

This leads us to the next definition.

Definition 3.4. *Given a shift number $\tilde{\nu} \in \mathbb{R}$, let s and ν be its integral part and the fractional part, $s \equiv \lfloor \tilde{\nu} \rfloor$ and $\nu \equiv \tilde{\nu} - s$, respectively. We define the matrix $\mathcal{K}(\tilde{\nu})$ as follows:*

$$(3.5) \qquad \mathcal{K}(\tilde{\nu}) \equiv \begin{cases} \mathbf{K}^s \mathbf{K}(\nu) & \text{if } s \geq 0, \\ \left(\mathbf{K}^T\right)^s \mathbf{K}(\nu) & \text{if } s < 0. \end{cases}$$

We will also use the notation, for $s \in \mathbb{Z}$ and $\omega \in \mathbb{R}$,

$$(3.6) \qquad \mathcal{K}(s, \omega) \equiv \begin{cases} \mathbf{K}^s \mathbf{K}(\omega) & \text{if } s \geq 0, \\ \left(\mathbf{K}^T\right)^s \mathbf{K}(\omega) & \text{if } s < 0. \end{cases}$$

Then it follows that $\mathcal{K}(\tilde{\nu})^T = \mathcal{K}(-\tilde{\nu})$ since if one writes out the integral and fractional parts $\tilde{\nu} = s + \nu$ and $-\tilde{\nu} = -(s+1) + (1-\nu)$,

$$(3.7) \qquad \mathcal{K}(\tilde{\nu})^T = \left(\mathbf{K}^T\right)^{s+1} \left((1-\nu)\mathbf{I} + \nu\mathbf{K}\right) = \left(\mathbf{K}^{s+1}\right)^T \mathbf{K}(1-\nu) = \mathcal{K}(-\tilde{\nu}).$$

All of Lemma 3.2 follows through easily.

Lemma 3.5. *Let $\mathcal{K}(\tilde{\nu})$ be as above. Let $s \equiv \lfloor \tilde{\nu} \rfloor$, $\nu \equiv \tilde{\nu} - s$ and $r \equiv \lfloor \tilde{\omega} \rfloor$, $\omega \equiv \tilde{\omega} - r$. Then it satisfies the following.*
  (a) *$\mathcal{K}(\tilde{\nu})\mathcal{K}(\tilde{\omega}) = \mathcal{K}(\tilde{\omega})\mathcal{K}(\tilde{\nu})$.*
  (b) *$\mathcal{K}(\tilde{\nu})\mathcal{K}(\tilde{\omega})^T = \mathcal{K}(\tilde{\omega})^T\mathcal{K}(\tilde{\nu})$.*
  (c) *For $\tilde{\nu}, \tilde{\omega}$ such that $0 \leq \nu, \omega \leq 1$ and $\nu + \omega \leq 1$, we have $\mathcal{K}(\tilde{\nu})\mathcal{K}(\tilde{\omega}) = \mathcal{K}(\tilde{\nu} + \tilde{\omega}) + \mathcal{O}(1/N^2)$, where the constant for $\mathcal{O}(1/N^2)$ is a shifted discrete Laplacian.*
  (d) *For $\mathbf{u} \in \mathbb{R}^N$, $\sum_{j=1}^{N} (\mathbf{u})_j = \sum_{j=1}^{N} (\mathcal{K}(\tilde{\nu})\mathbf{u})_j$.*

Proof. We omit the proof, as it is similar to that of Lemma 3.2. ∎

This is a generalization of the upwind method beyond the constraint of the CFL condition, and can be viewed as a special case of the LTS method [20], which can also be extended to nonlinear systems [21, 22].

3.2. Transport reversal in $\mathbb{R}$**.** Using the definitions above, we generalize the minimization problem (2.4) to be applied to a snapshot matrix $\mathbf{A}$. We will use the notation $\mathbf{D} \equiv \mathbf{K} - \mathbf{I}$ and $\mathbf{L} \equiv \mathbf{K} + \mathbf{K}^T - 2\mathbf{I}$.

Lemma 3.6. *Suppose* $\mathbf{a}, \mathbf{b} \in \mathbb{R}^N$ *are nonconstant.*
If we let $\nu = \mathrm{argmin}_{\omega \in \mathbb{R}} \|\mathbf{b} - \mathbf{K}(\omega)^T \mathbf{a}\|_2^2$, *then*

$$(3.8) \qquad \nu = \frac{1}{2}\left(1 - 2\frac{\mathbf{a}^T \mathbf{D} \mathbf{b}}{\mathbf{a}^T \mathbf{L} \mathbf{a}}\right).$$

Proof. Let

$$\mathcal{J}(\omega) \equiv \left\|\mathbf{b} - \mathbf{K}(\omega)^T \mathbf{a}\right\|_2^2 = \left(\mathbf{b} - \mathbf{K}(\omega)^T \mathbf{a}\right)^T \left(\mathbf{b} - \mathbf{K}(\omega)^T \mathbf{a}\right)$$
$$= \mathbf{b}^T \mathbf{b} - 2\mathbf{a}^T \mathbf{K}(\omega)\mathbf{b} + \mathbf{a}^T \mathbf{K}(\omega)\mathbf{K}(\omega)^T \mathbf{a}.$$

Taking a derivative,

$$\mathcal{J}'(\omega) = -2\mathbf{a}^T \mathbf{K}'(\omega)\mathbf{b} + \mathbf{a}^T \mathbf{K}'(\omega)\mathbf{K}(\omega)^T \mathbf{a} + \mathbf{a}^T \mathbf{K}(\omega)\mathbf{K}'(\omega)^T \mathbf{a}.$$

Letting $\mathcal{J}'(\nu) = 0$ and expanding, we have

$$\nu = \frac{1}{2}\left(1 - 2\frac{\mathbf{a}^T(\mathbf{K} - \mathbf{I})\mathbf{b}}{\mathbf{a}^T(\mathbf{K} + \mathbf{K}^T - 2\mathbf{I})\mathbf{a}}\right) = \frac{1}{2}\left(1 - 2\frac{\mathbf{a}^T \mathbf{D} \mathbf{b}}{\mathbf{a}^T \mathbf{L} \mathbf{a}}\right).$$

The nullspace of $\mathbf{L}$ is the span of constant vectors, so the denominator on the right-hand side does not vanish. ∎

Recall that the CFL condition required that $\nu \in [0, 1]$, and note that ν given by (3.8) is not guaranteed to lie in this stability region. The minimization (2.4) is now extended to the case when $\mathbf{K}$ is replaced by the matrix $\mathcal{K}(\tilde{\nu})$,

$$(3.9) \qquad \min_{\tilde{\omega} \in \mathbb{R}} \left\|\mathbf{a} - \mathcal{K}(\tilde{\omega})^T \mathbf{b}\right\|_2^2.$$

It is immediate that this problem is symmetric with respect to the vectors $\mathbf{a}$ and $\mathbf{b}$ up to $\mathcal{O}(1/N^2)$. That is, the problem can be rewritten using (3.4),

$$(3.10) \qquad \min_{\tilde{\omega} \in \mathbb{R}}\left(\left\|\mathbf{b} - \mathcal{K}(\tilde{\omega})^T \mathbf{a}\right\|_2^2 + \mathcal{O}\left(1/N^2\right)\right).$$

The problem (3.9) is only of one variable $\tilde{\omega}$ lying in an interval $[0, N]$, although this can be viewed as a nonconvex minimization problem in $\mathbb{R}^N$ as we will see in section 4. With the partitioning $\{[m, m+1] : m = 0, \ldots, N-1\}$ of $[0, N]$, a recursive relation can be found for the formula (3.8) in terms of j, restricting the variable $\tilde{\omega}$ to a set of $2N - 1$ positive reals.

Lemma 3.7. *Suppose* $\mathbf{a}, \mathbf{b} \in \mathbb{R}^N$ *and let* $\nu_s^{\pm}$ *be defined as*

$$(3.11) \qquad \nu_s^+ \equiv \mathrm{argmin}_{\omega \in \mathbb{R}} \left\|\mathbf{b} - \mathcal{K}(s, \omega)^T \mathbf{a}\right\|_2^2, \quad \nu_s^- \equiv \mathrm{argmin}_{\omega \in \mathbb{R}} \left\|\mathbf{b} - \mathcal{K}(-s, \omega)^T \mathbf{a}\right\|_2^2.$$

Then we have the relations

$$(3.12) \qquad \nu_{s+1}^+ = \nu_s^+ + \frac{(\mathbf{K}^s \mathbf{b})^T \mathbf{L} \mathbf{a}}{\mathbf{a}^T \mathbf{L} \mathbf{a}}, \quad \nu_{s+1}^- = \nu_s^- + \frac{((\mathbf{K}^T)^s \mathbf{b})^T \mathbf{L} \mathbf{a}}{\mathbf{a}^T \mathbf{L} \mathbf{a}}.$$

Proof. The proof is easy and is given in Appendix A. ∎

Filtering out $\nu_s^\pm$ that do not satisfy the CFL condition, we let

$$\hat{\nu}_s^\pm \equiv \begin{cases} \pm s + \nu_s^\pm & \text{if } \nu_s \in [0,1] \\ \pm s & \text{otherwise.} \end{cases}$$

Then the minimization problem (3.9) only requires comparison of at most $2N - 1$ discrete values,

$$(3.13) \qquad \tilde{\nu} = \operatorname{argmin}_{\tilde{\omega} \in W} \left\| \mathbf{b} - \mathcal{K}(\tilde{\omega})^T \mathbf{a} \right\|_2^2 \text{ where } W \equiv \left\{ 0, \hat{\nu}_0^+, 1, \hat{\nu}_1^+, \dots, N-1, \hat{\nu}_{N-1}^+ \right\},$$

or equivalently, $W = \{0, \hat{\nu}_{N-1}^-, 1, \hat{\nu}_{N-2}^-, \dots, N-1, \hat{\nu}_0^-\}$.

In many examples the data $\mathbf{a}$ and $\mathbf{b}$ have localized features. This fact can be incorporated into our computation of (3.9) by assuming that $\mathbf{a}$ and $\mathbf{b}$ are sparse representations that reflect these features well, thereby W. Reduction of W beyond this may be possible by using discrete Fourier transforms and exploiting isotropy that might exist in $\mathbf{a}$ or $\mathbf{b}$ (see Proposition 4.1.)

Definition 3.8 (transport reversal in $\mathbb{R}$). *Given a matrix $\mathbf{A} \in \mathbb{R}^{N \times M}$, let $\mathbf{a}_j$ denote the jth column of $\mathbf{A}$, and let $\mathbf{b} \in \mathbb{R}^N$ be a given pivot. Let*

$$(3.14) \qquad \tilde{\nu}_j \equiv \operatorname{argmin}_{\tilde{\omega} \geq 0} \left\| \mathbf{a}_j - \mathcal{K}(\tilde{\omega})^T \mathbf{b} \right\|_2^2 \quad \text{for } j = 1, \dots, M.$$

This computation is denoted by $\tilde{\boldsymbol{\nu}} = \tilde{\mathcal{C}}(\mathbf{A}; \mathbf{b})$.

We define the transport $\mathcal{T}$ of $\mathbf{A}$,

$$(3.15) \qquad \mathcal{T}(\mathbf{A}; \tilde{\boldsymbol{\nu}}) \equiv \begin{bmatrix} \mathcal{K}(\tilde{\nu}_1) & \mathcal{K}(\tilde{\nu}_2) & \cdots & \mathcal{K}(\tilde{\nu}_M) \end{bmatrix} \odot \begin{bmatrix} \mathbf{a}_1 & \mathbf{a}_2 & \cdots & \mathbf{a}_M \end{bmatrix}.$$

Let $\tilde{\nu} \equiv \tilde{\mathcal{C}}(\mathbf{A}; \mathbf{b})$ and $\mathring{\mathbf{A}} \equiv \mathcal{T}(\mathbf{A}; -\tilde{\boldsymbol{\nu}})$. Now, the orthogonality of the eigenvectors of $\mathring{\mathbf{A}}^T \mathring{\mathbf{A}}$ is not strictly preserved under the action of $\mathcal{K}(\tilde{\nu})$, but holds up to $\mathcal{O}(1/N^2)$. This is an analogue of [5, Proposition 3] and is stated as follows.

Proposition 3.9. *If φ is an eigenvector of $\mathring{\mathbf{A}}^T \mathring{\mathbf{A}}$ with eigenvalue λ, then φ is also an eigenvector of $(\mathcal{K}(\tilde{\nu})\mathring{\mathbf{A}})^T (\mathcal{K}(\tilde{\nu})\mathring{\mathbf{A}})$ to the same λ for every $\tilde{\nu} \in \mathbb{R}$, up to $\mathcal{O}(1/N^2)$.*

Proof. The proof follows immediately from (3.4) in Lemma 3.3. ∎

3.3. Sharpening procedure. Once $\mathring{\mathbf{A}} \equiv \mathcal{T}(\mathbf{A}; -\tilde{\boldsymbol{\nu}})$ is computed, we can apply the SVD to construct a reduced basis representation of $\mathring{\mathbf{A}}$. Let us denote this low-rank representation by $\tilde{\mathbf{A}}$, and columns of $\tilde{\mathbf{A}}$ by $\tilde{\mathbf{a}}_j$. For a reconstruction of $\mathbf{A}$ itself, we compute the forward transport, $\mathcal{T}(\tilde{\mathbf{A}}; \tilde{\boldsymbol{\nu}})$. While this yields an acceptable reconstruction, the numerical diffusion arising from the upwind flux causes $\mathcal{O}(1/N^2)$ amount of smearing. This numerical diffusion has a particular structure (3.4) in the form of a discrete Laplacian $\mathbf{L}_h$. This can be utilized to improve the accuracy by applying a postprocessing procedure motivated as follows.

Suppose we are given a column $\mathbf{a}$, to which we apply the reversal then reconstruction as above. Then the reconstruction, which we denote by $\mathbf{b}$, satisfies the equation $\mathcal{K}(\tilde{\nu})\mathcal{K}(\tilde{\nu})^T \mathbf{a} = \mathbf{b}$. Recall that

$$\mathcal{K}(\tilde{\nu})\mathcal{K}(\tilde{\nu})^T = \mathbf{I} + \alpha \mathbf{L}_h, \quad \text{where } \alpha = \frac{\nu(1 - \nu)}{N^2} \text{ and } \mathbf{L}_h \equiv \frac{1}{h^2} \left(\mathbf{K} + \mathbf{K}^T - 2\mathbf{I} \right).$$

Thus we can recover $\mathbf{a}$ from $\mathbf{b}$ by solving a discretized Helmholtz equation augmented with a set of boundary conditions. For example, we can use the first and last values of $\mathbf{a}$,

$$(3.16) \qquad (\mathbf{I} + \alpha \mathbf{L}_h)\, \mathbf{u} = \mathbf{b}, \quad \text{satisfying} \quad u_1 = a_1 \quad \text{and} \quad u_N = a_N.$$

This inversion acts to remove the diffusive error caused by grid interpolation (3.2). This can also be seen as a sharpening procedure, once rewritten as

$$\frac{\mathbf{u} - \mathbf{b}}{k} = \beta \mathbf{L}_h \mathbf{u}, \quad \beta \equiv -\frac{\alpha}{k}.$$

Due to the negative sign of β, here $\mathbf{u}$ is shown as the single time-step solution to the backward heat equation with step size k (a parameter that has been introduced for illustrative purpose).

We will denote this solution operator to (3.16) by $(\mathbf{I} + \alpha \mathbf{L}_h)^{-1}$. Letting $\alpha_j \equiv \nu_j(1-\nu_j)/N^2$, we apply this sharpening procedure for each column of the reconstructed $\tilde{\mathbf{A}}$, that is,

$$(3.17) \qquad \left[(\mathbf{I} + \alpha_1 \mathbf{L}_h)^{-1} \mathcal{K}(\tilde{\nu}_1) \quad \cdots \quad (\mathbf{I} + \alpha_{M-1} \mathbf{L}_h)^{-1} \mathcal{K}(\tilde{\nu}_{M-1}) \right] \odot \left[\tilde{\mathbf{a}}_1 \quad \cdots \quad \tilde{\mathbf{a}}_M \right].$$

As mentioned in remarks following Lemma 3.3, this procedure aims to address the fact that the discretized advection or translation no longer forms a symmetry group exactly. The reversal and reconstruction procedure will be demonstrated numerically in section 3.4.3.

3.4. Variable speed transport reversal and linear systems. In the previous section we have introduced a transport procedure amounting to a long-time solution of a constant speed advection equation. Now we consider a generalization of the reversal problem (1.7) when the advection speed $c > 0$ in (1.1) is allowed to depend on the spatial variable x. For simplicity, c will be represented as a piecewise constant function over a uniform grid.

In the previous section, the transport reversal (3.14) has largely been a discretization of the continuous minimization problem (1.7) with some numerical error (3.3). But in considering the variable speed setting, additional differences between the discrete and the continuous case come to the fore. Consider c which has the following property: there exists $\overline{\omega} > 0$ and a grid $\{x_j\}_{j=0}^N$ and grid-sizes $\Delta x_j = x_{j+1} - x_j$,

$$(3.18) \qquad |\overline{\omega} - \omega_j| < \delta \ll 1, \quad \text{where } \omega_j = \frac{c(x_{j+1/2})}{\Delta x_j} \quad \text{for } j = 0, \ldots, N.$$

That is, even if $c(x_{j+1/2})$ varies, care can be taken to adjust size of the cells Δx_j so that the shift number $\nu_j = \omega_j \Delta t$ behaves like a constant multiple of Δt for all cells. Then, the discretized problem is identical to the constant speed case and the techniques introduced in the previous section apply directly, so the extension to variable speed c satisfying (3.18) is trivial. Let us give a simple example of c and $\{x_j\}_{j=0}^N$ that satisfies this property. Consider the advection equation (1.1) and suppose c took on two values and the grid $\{x_j\}$ was constructed as follows:

$$c(x) = \begin{cases} 1 & \text{if } 0 \le x < \frac{1}{2}, \\ \frac{1}{2} & \text{if } \frac{1}{2} \le x \le 1, \end{cases} \quad \text{and} \quad x_j = \begin{cases} \frac{3}{2N} j & \text{if } j < N/3, \\ \frac{3}{4N} j + \frac{1}{4} & \text{if } j \ge N/3. \end{cases}$$

If we choose N to be a multiple of 3, then (3.18) holds with $\bar{\omega} = 2N/3$ and $\delta = 0$.

However, the property (3.18) is not easily guaranteed, especially when dealing with systems, when the characteristic variables have different speeds, or when c is allowed to depend on time, as in the nonlinear case. Here we address the general variable speed case in section 3.4.1, even if c does not satisfy the property (3.18).

3.4.1. Variable speed reversal. We proceed to generalize the transport reversal procedure by employing the LTS method [20, 21, 22]. This is also reminiscent of Lagrangian methods such as the particle-in-cell method [10, 14] or its variant the material point method [37, 36]. The LTS method allows long-time reversal of the given wave profile without incurring excessive numerical diffusion, mimicking the behavior of the matrix $\mathcal{K}(\tilde{\omega})$.

The given vector $(u_j^n)_{j=1}^N$ will be considered to represent a discretization of a function $u(x)$ lying on a uniform grid $\{x_j\}_{j=0}^N$ of the domain $\Omega = [0, 1]$. Let us define the jumps $\Delta_j^n \equiv u_j^n - u_{j-1}^n$, where the index j is defined modulo N.

Now, the grid points will serve as particles or material points, and their positions will evolve with time. We index the time-dependence by ℓ, letting $\{x_j^\ell\}_{j=0}^N$ denote the grid points at time t_ℓ. Then we evolve the grid points as a function of time, $x_j = x_j(t)$, according to the ordinary differential equation

$$(3.19) \qquad \begin{cases} \dot{x}_j = c(x_j), \\ x_j(0) = x_j^0 \end{cases} \qquad \text{for } j = 0, 1, \ldots, N$$

with periodic boundary conditions. We will evolve backward in time, as is natural for the reversal procedure. The problem will be solved up to time $t^L < 0$. The solution at t^L is given by

$$(3.20) \qquad x_j(t^L) = x_j(0) + \int_0^{t^L} c\left(x_j(t)\right) \, \mathrm{d}t.$$

Recall $c(x)$ was assumed to be piecewise constant, so we let $c_j \equiv c(x_{j+1/2})$ and we compute (3.20) explicitly.

Let us be given time-steps $0 = t^0 > t^1 > \cdots > t^L$ with $\Delta t^\ell = t_\ell - t_{\ell-1} < 0$. Define Δt_{ij}^ℓ as the amount of time $x_j(t)$ lies in the ith cell $\mathcal{C}_i$ during the time interval $[t_\ell, t_{\ell-1}]$, $\Delta t_{ij}^\ell \equiv -|\{t \in [t_{\ell-1}, t_\ell] : x_j(t) \in \mathcal{C}_i\}|$. Naturally, this is a partition of the time interval $[t_\ell, t_{\ell-1}]$ so $\Delta t^\ell = \sum_{i=0}^N \Delta t_{ij}^\ell$. Then the solution x_j^ℓ at time $t_\ell = t_0 + \sum_{k=1}^\ell \Delta t^k$ is given by
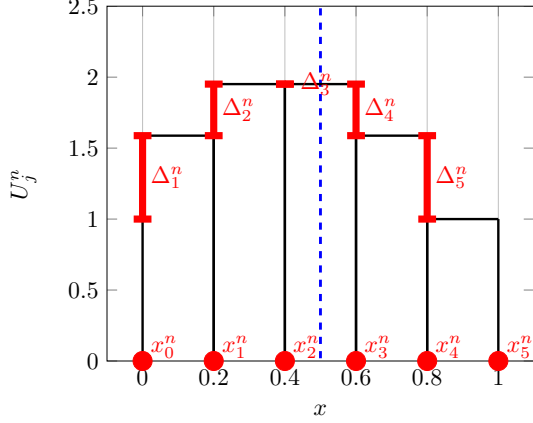
$$(3.21) \qquad x_j^{\ell+1} = x_j^\ell + \sum_{i=1}^N c_i \Delta t_{ij}^\ell \qquad \mod 1.$$

Once time-stepping has reached the final time t^L, we can update the cell average by computing the total flux for each cell,
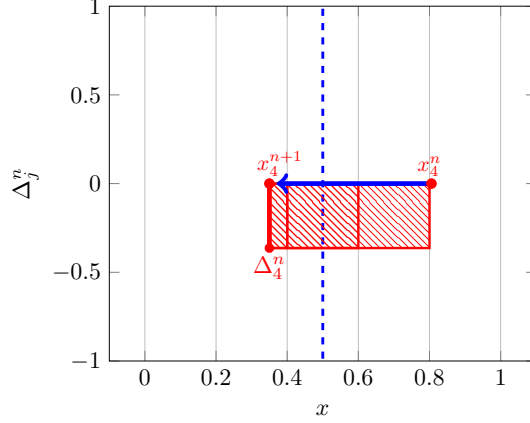
$$(3.22) \qquad u_j^{n+1} = u_j^n + \sum_{i=1}^N \Delta_i^n c_i \Delta t_{ij}^\ell.$$

The procedure is sketched in Figure 8.

Given u_j^n, place a material point at each grid point x_j^0 (red dots) and compute the jumps Δ_j^n at these points (red lines). Assign the jumps to the material points.

Advect the material points, computing the total flux caused by each jump, for each cell (e.g, jump for x_4^n below will change the volume of cells $\mathcal{C}_2, \mathcal{C}_3, \mathcal{C}_4$).

Figure 8. *An illustration of the variable speed advection with periodic boundary conditions. The dotted blue line at $x = 0.5$ denotes the interface where c changes. After the fluxes for each cell are computed, we update u_j^n by computing the total change of volume as in (3.22), resulting in u_j^{n+1}.*

Although the time-steps Δt^ℓ are not technically necessary since this ODE can be solved to any time in one step, we define it in order to maintain an analogy to the constant speed case. Let T denote the period of the solution x_j to (3.19), so that $x_j(t + T) = x_j(t)$. Recall that shift number $\tilde{\nu} \in \mathbb{R}$ satisfied the periodicity condition $\mathcal{K}(\tilde{\nu} + N) = \mathcal{K}(\tilde{\nu})$, which should correspond to periodicity $x_j(t + T) = x_j(t)$. Notice that T satisfies the relationship

$$(3.23) \qquad \bar{c} \equiv \frac{|\Omega|}{T} = \frac{1}{|\Omega|} \int_\Omega c(x) \, dx,$$

where $|\Omega|$ denotes the measure of Ω. Given $\tilde{\nu}$ and c we let the time-step Δt satisfy

$$\tilde{\nu} = \frac{N \Delta t}{T} = \bar{c} \frac{\Delta t}{|\Omega| / N} = \left(\sum_{j=0}^{N} c_j \Delta x_j^0 \right) \frac{\Delta t}{\sum_{j=0}^{N} \Delta x_j / N}.$$

Using this definition, we denote the reversal procedure above by the operator $\mathcal{K}_c(\tilde{\nu})$. Note in particular that $\mathcal{K}_c(\tilde{\nu} + N) = \mathcal{K}_c(\tilde{\nu})$ holds with high numerical accuracy. Now the minimization problem to be used in the variable speed reversal can be written down. Replacing the transpose in (3.9) by a negation of the argument using (3.7), we have

$$(3.24) \qquad \min_{\tilde{\omega} \in \mathbb{R}} \| \mathbf{a}_j - \mathcal{K}_c(-\tilde{\omega}) \mathbf{b} \|_2^2.$$

Now we define the reversal for the variable speed case.

Definition 3.10 (variable speed transport reversal).   *Given* $\mathbf{A} \in \mathbb{R}^{N \times M}$ *and a variable speed* $c : [0, N] \to \mathbb{R}$, *let* $\mathbf{a}_j$ *denote the jth column of* $\mathbf{A}$. *Let* $\mathbf{b} \in \mathbb{R}^N$ *be a given* pivot. *Then compute*

$$(3.25) \qquad\qquad \tilde{\nu}_j = \operatorname{argmin}_{\tilde{\omega} \in \mathbb{R}} \|\mathbf{a}_j - \mathcal{K}_c(-\tilde{\omega})\mathbf{b}\|_2^2 \quad \text{for } j = 1, \ldots, M.$$

This computation is denoted by $\tilde{\boldsymbol{\nu}} = \tilde{\mathcal{C}}_c(\mathbf{A}; \mathbf{b})$, *where* $(\tilde{\boldsymbol{\nu}})_j = \tilde{\nu}_j$.

We define the transport *of* $\mathbf{A}$ *with speed c, denoted by*

$$(3.26) \qquad\qquad \mathcal{T}_c(\mathbf{A}; \boldsymbol{\nu}) \equiv \begin{bmatrix} \mathcal{K}_c(\tilde{\nu}_1) & \cdots & \mathcal{K}_c(\tilde{\nu}_M) \end{bmatrix} \odot \begin{bmatrix} \mathbf{a}_1 & \cdots & \mathbf{a}_M \end{bmatrix}.$$

Also define the transport reversal *of* $\mathbf{A}$, *distinguished by the sign of the shift numbers* $\tilde{\boldsymbol{\nu}}$,

$$(3.27) \qquad\qquad \mathcal{T}_c(\mathbf{A}; -\boldsymbol{\nu}) \equiv \begin{bmatrix} \mathcal{K}_c(-\tilde{\nu}_1) & \cdots & \mathcal{K}_c(-\tilde{\nu}_M) \end{bmatrix} \odot \begin{bmatrix} \mathbf{a}_1 & \cdots & \mathbf{a}_M \end{bmatrix}.$$

The orthogonality condition in Proposition 3.9 still holds with small error. Also, no simple relation such as (3.12) are found, and the sharpening procedure (3.17) cannot be easily applied. This is due to the loss of convexity to be discussed in section 4.

3.4.2. Reversal for linear systems with pivoting. Let us now consider the reversal for snapshot matrices arising from linear systems of equations. We will focus on the acoustic equation (P4),

$$(3.28) \qquad\qquad \begin{bmatrix} p \\ u \end{bmatrix}_t + \begin{bmatrix} 0 & K \\ 1/\rho & 0 \end{bmatrix} \begin{bmatrix} p \\ u \end{bmatrix}_x = 0.$$

Parameters ρ and K are the density and the bulk modulus of compressibility of the material, respectively. Eigendecomposition of the matrix yields eigenpairs,

$$(3.29) \qquad\qquad \lambda^1 = u - c, \quad \mathbf{r}_1 = \begin{bmatrix} -\rho c \\ 1 \end{bmatrix} \quad \text{and} \quad \lambda^2 = u + c, \quad \mathbf{r}_2 = \begin{bmatrix} \rho c \\ 1 \end{bmatrix},$$

where $c \equiv \sqrt{K/\rho}$. We can rewrite (3.28) in terms of new variables r_1 and r_2 by projecting the state vector $[p, u]^T$ onto the eigenspace spanned by the two vectors in (3.29). When c is constant, the system can be completely decoupled, and two advection equations can be solved separately. However, when c depends on the spatial variable x, the eigendecomposition also depends on x. This implies that even after the eigendecomposition, there is a coupling between the variables r_1 and r_2 across space if ρc varies, so that the wave profiles will evolve. For example, r_1 may initially be identically zero at initial time but suddenly develop nonempty support as soon as a wave profile in r_2 passes through an interface and is partially reflected. An example of this kind is shown in Figure 12 below.

Therefore we need to dynamically change the pivot vector $\mathbf{b}$ in (3.24) appropriately to other columns as the wave profile evolves. Let us recall the *pivot map* $\ell : \{1, \ldots, M\} \to \{1, \ldots, M\}$, which takes each column $\mathbf{a}_j$ to its corresponding pivot $\mathbf{a}_{\ell(j)}$. Then we may define reversal with pivoting as follows.

Definition 3.11 (variable speed transport reversal with pivoting).   *Let the matrix* $\mathbf{A} \in \mathbb{R}^{N \times M}$ *be given, a pivot map* $\ell : \{1, \ldots, M\} \to \{1, \ldots, M\}$, *and a variable speed* $c : [0, N] \to \mathbb{R}$. *Then let*

$$(3.30) \qquad\qquad \tilde{\nu}_j \equiv \operatorname{argmin}_{\tilde{\omega} \in \mathbb{R}} \|\mathbf{a}_j - \mathcal{K}_c(-\tilde{\omega})\mathbf{a}_{\ell(j)}\|_2^2 \quad \text{for } j = 1, \ldots, M.$$

This computation is denoted by $\tilde{\boldsymbol{\nu}} = \tilde{\mathcal{C}}_c(\mathbf{A}; \ell)$, *where* $(\tilde{\boldsymbol{\nu}})_j = \tilde{\nu}_j$.
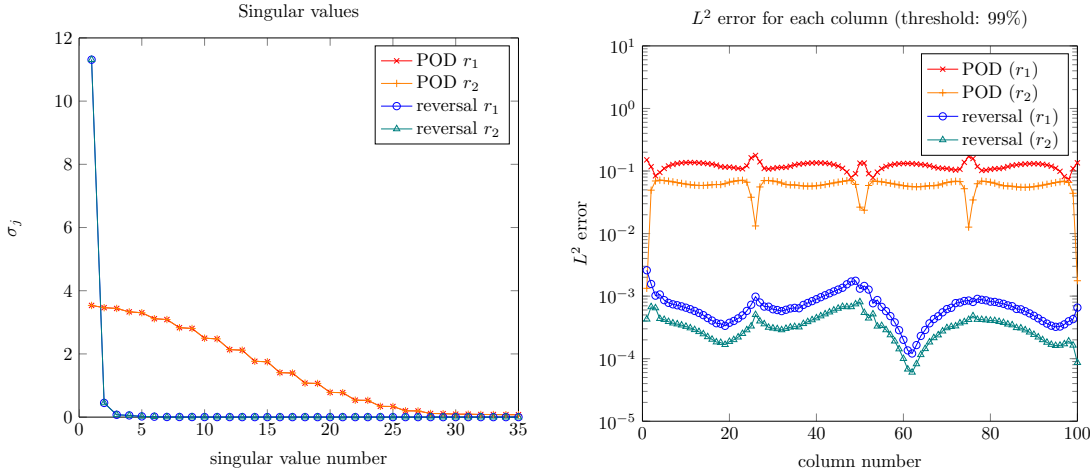
Figure 9. *Left: Fast decay of singular values of snapshots in the variables r_1 and r_2 for the acoustic equation with homogeneous media. The largest 35 singular values are shown. Three singular values represent 99% of the threshold for both variables. Right: L^2 error for each column of the reconstruction for eigenvector variables r_1 and r_2.*

The proper pivoting criterion will depend on the problem at hand, and for acoustic equations with heterogeneous media, pivoting when there is large relative change in the ℓ^2-norm difference between the previous and current column was sufficient. See Example 3.6 for numerical results using this particular pivoting criterion.

3.4.3. Numerical experiments. We apply the transport reversal and reconstruction procedure outlined in this section to the acoustic equation, in both homogeneous and heterogeneous media. We do not introduce the iterative procedure from section 2, and consider relatively simple examples to focus on the effect of the extensions of the shift operator.

3.5. Acoustic equation in homogeneous media. We apply the reversal (3.15) to the constant speed acoustic equation (3.28) with $K \equiv K_0$ and $\rho \equiv \rho_0$ with periodic boundary conditions. For the initial conditions, p_0 is given to be a Gaussian hump and u_0 to be identically zero.

The 100 snapshots were taken from a 100-cell solution. After an eigendecomposition of the state vectors (3.29) we transform the state variables p and u, to the characteristic variables r_1 and r_2. Then we apply the reversal procedure (3.15) to the two snapshot matrices corresponding to these variables.

The decay of the singular values for $\mathring{\mathbf{A}}$ is clearly much more rapid, as seen from Figure 9. The threshold of 99% is achieved with only 3 basis vectors. The reconstruction is plotted against the snapshot itself in Figure 10, and they are nearly identical. The POD reconstruction is also plotted. The L^2-errors from the two reconstructions are compared in Figure 9, where the reversal consistently outperforms the naïve POD.

3.6. Acoustic equation with heterogeneous media. Now we consider the acoustic equation (3.28) in heterogeneous media, with two different materials. An interface will be located at $x = 0.5$. Letting ℓ designate the *left* part of the domain $(0, 0.5)$ and r the *right* part of the
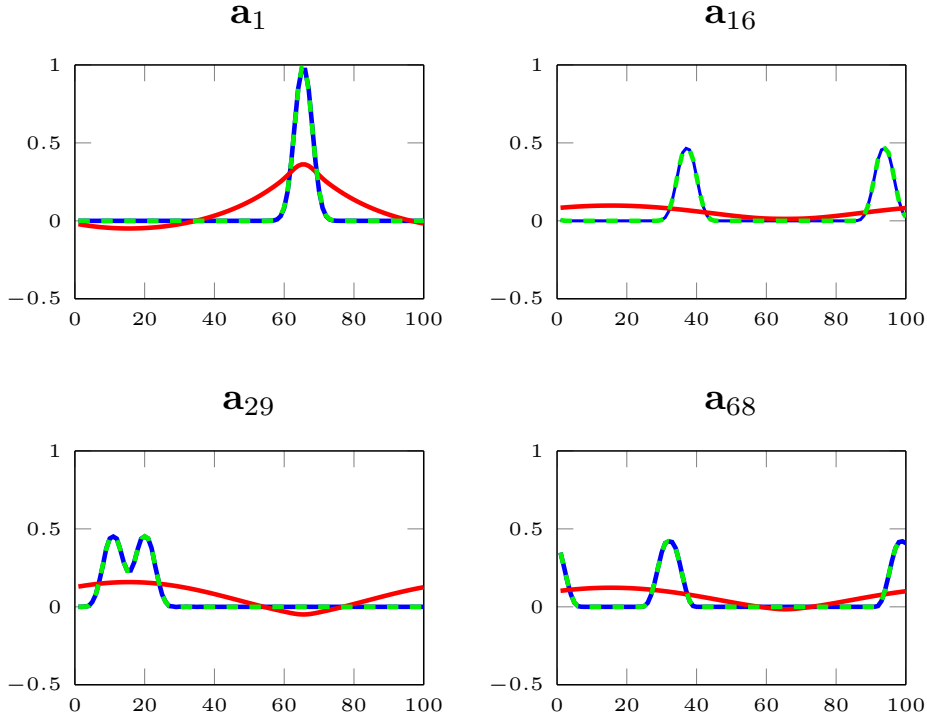
Figure 10. *Reconstruction of the solution to acoustic equation with homogeneous media, for state variable p. The snapshot is given in dashed green, the reversal reconstruction in blue, and POD reconstruction in red. Three reduced basis vectors were used for both reversal and POD.*

domain $(0.5, 1)$, suppose we have the parameters K and ρ vary depending on the part of the domain. Here we let $\rho_\ell = 1$, $K_\ell = 1$ and $\rho_r = 4$, $K_r = 1$, so that $c_\ell = 1$ and $c_r = 0.5$. We again impose periodic boundary conditions, and this creates two more interfaces, at $x = 0$ and 1. The initial conditions p_0 and u_0 are both Gaussian humps of identical shape traveling toward the interface at $x = 0.5$.

The 100 snapshots were taken from a 100-cell solution. An eigendecomposition of the state-vectors transform the variables u and p, to r_1 and r_2 as in the previous example. Then we apply variable speed reversal with pivoting (3.30) on each of the matrices for these variables.

Here finding a suitable pivot map becomes necessary. We track the change of the profile by computing the relative ℓ^2-norm difference between the previous and the current column, and pivots to the current column when it exceeds some threshold γ. That is, the pivot map is given by

$$\ell(j) = \max_k \left\{ k \in \mathbb{Z} : 0 \le k \le j, \frac{\|\mathbf{a}_{j+1} - \mathbf{a}_j\|_2}{\|\mathbf{a}_j\|_2} \ge \gamma \right\}.$$

For this example, setting $\gamma = 0.15$ was appropriate.

The achieved decay in singular values, along with the L^2-errors for each snapshot, are shown in Figure 11. Note how the error for the reversal is concentrated near the interface. Away from the interface, the traveling wave solution is much more accurately captured with
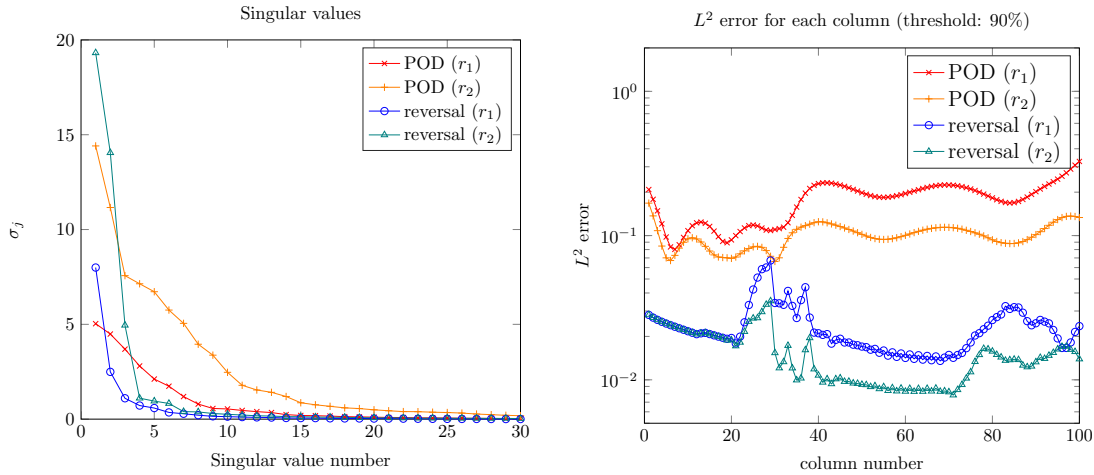
Figure 11. *Faster decay of singular values of r_1 and r_2 for the acoustic equation* (1.1) *with heterogeneous media (left). The largest* 30 *singular values are shown. Five and seven singular values represent* 90% *threshold for these variables, respectively. L^2-error for each column of reconstruction for characteristic variables r_1 and r_2. The error is concentrated near the snapshots in which the wave profile is undergoing quick change near the interface.*

the reversal. The decay of the singular values can also be interpreted in this context. While the decay is clearly more rapid than the POD modes, the difference is not as striking when compared to the case of homogeneous media. The singular modes whose corresponding singular values belong to this trailing part represent the rapidly changing shape of the wave profile near the interface. The slower decay is attributable to the presence of these modes.

A few sample reversal reconstruction are plotted along with the snapshot itself and the POD reconstruction in Figure 12. Seven and five reduced basis vectors, for r_1 and r_2, respectively, were used for the reconstruction. The accuracy of the reconstruction visibly deteriorates near the interface.

4. Geometric interpretation. In this section, we present some geometric interpretations of the reversal procedure introduced in the sections 2 and 3.4 that arise naturally. Recall the matrices $\mathbf{K}(\nu)$ (Definition 3.1), $\mathcal{K}(\tilde{\nu}) = \mathcal{K}(s, \nu)$ (3.5) and variable speed operator $\mathcal{K}_c(\tilde{\nu})$ (3.24). Let us define

$$(4.1) \qquad \mathcal{I}_s(\mathbf{a}) \equiv \{\mathcal{K}(s, \nu)\mathbf{a} : \nu \in [0, 1]\} \text{ and } \mathcal{M}(\mathbf{a}) \equiv \bigcup_{s \in \mathbb{Z}} \mathcal{I}_s = \{\mathcal{K}(\tilde{\nu})\mathbf{a} : \tilde{\nu} \in \mathbb{R}\}.$$

Observe that $\mathcal{I}_s(\mathbf{a})$ is the convex hull of $\{\mathbf{K}^s\mathbf{a}, \mathbf{K}^{s+1}\mathbf{a}\}$. Therefore, given any column vector $\mathbf{a} \in \mathbb{R}^N$, $\mathcal{M}(\mathbf{a})$ is a union of one-dimensional intervals lying in $\mathbb{R}^N$ (4.1), although $\mathcal{M}(\mathbf{a})$ is not convex in $\mathbb{R}^N$ in general.

In the minimization problem (3.9) we are choosing a point on this polygon that is closest to $\mathbf{b}$. Since $\mathbf{K}$ is an isometric map, the vertices of the polygon lie on the N-sphere of radius $\|\mathbf{a}_j\|_2$. Note that when choosing a point in the interior of $\mathcal{I}_s(\mathbf{a})$, we are not preserving $\|\mathbf{a}\|_2$ but $\sum_{j=1}^N (\mathbf{a})_j$ due to mass conservation in Lemma 3.2(d). This is also reflected in the $\mathcal{O}(1/N^2)$ numerical diffusion term in $\mathcal{K}(\tilde{\nu})^T \mathcal{K}(\tilde{\nu}) - \mathbf{I}$ (3.3).
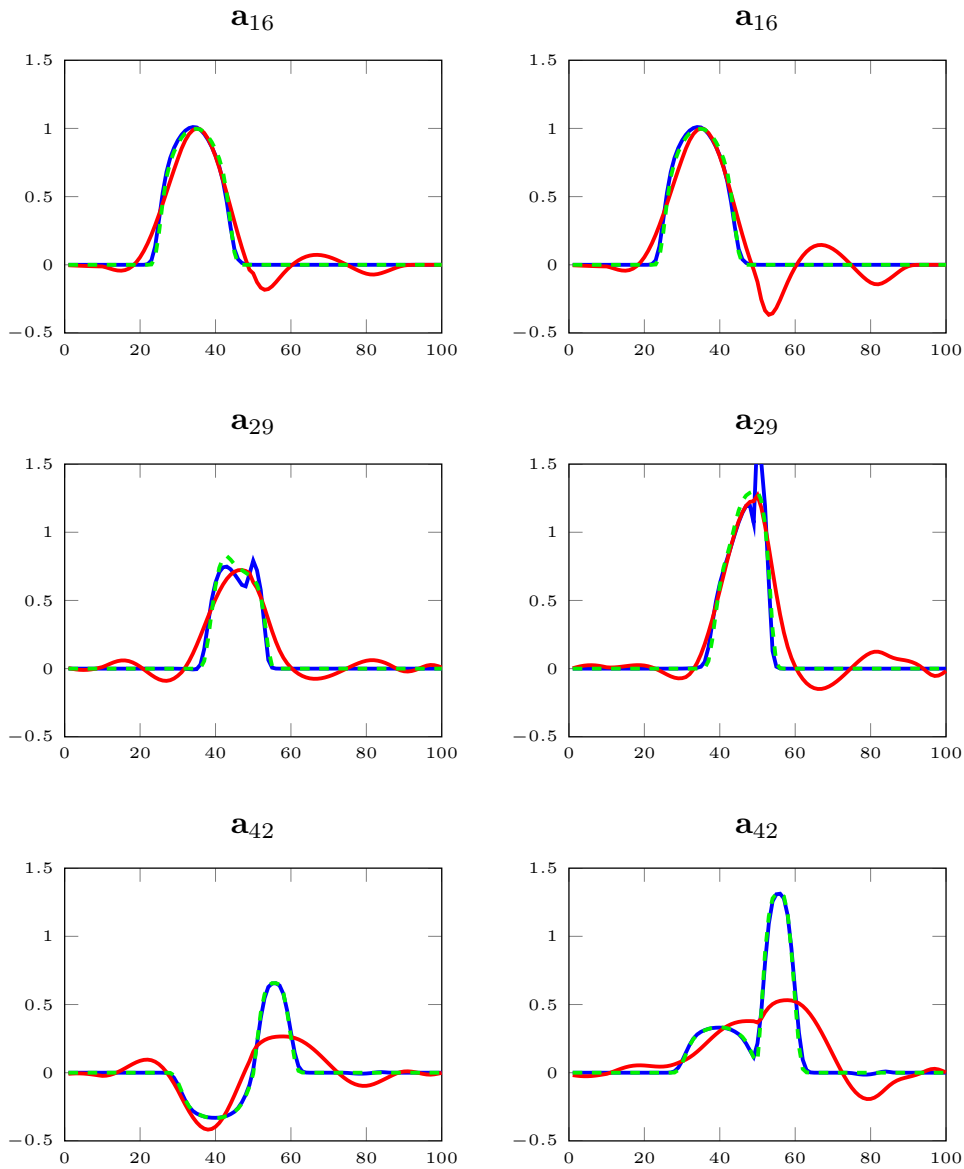
Figure 12. *Reconstruction of the solution to acoustic equation for state variable p (left column) and u (right column). The snapshot is given in dashed green, reversal reconstruction in blue, and POD reconstruction in red. Four reduced vectors were used for each reconstruction.*

When computing the reversal of a matrix $\mathbf{A}$, each column $\mathbf{a}_j$ is transformed along its corresponding polygon $\mathcal{M}(\mathbf{a}_j)$. The orientation of this polygon is determined by the column $\mathbf{a}_j$ itself. On the other hand, it is easy to see that the $\mathcal{M}(\mathbf{a})$ is always regular, since the angle between the tangent vectors along its edges are $-\mathbf{a}^T(\mathbf{K} + \mathbf{K}^T - 2\mathbf{I})\mathbf{a}/\|\mathbf{D}\mathbf{a}\|_2^2$.

As a special case, if $\mathbf{A} = \mathbf{I}$, all the regular polygons corresponding to each column vector of $\mathbf{A}$ coincide, and therefore the reversal is able to eliminate the functional (3.9). Then using

a single reduced basis vector of $\mathring{\mathbf{A}}$ suffices. This indicates that the decay of singular values of $\mathring{\mathbf{A}}$ depends on how well the polygons $\mathcal{M}(\mathbf{a}_j)$ are aligned with respect to each other.

It follows easily that the $\mathcal{M}(\mathbf{a})$ shrinks to a point as $\mathbf{a}$ approaches $\mathbf{1}$. So the problem of finding the optimal point in the $\mathcal{M}(\mathbf{a})$ becomes more constrained and then finally becomes trivially ill-posed when $\mathbf{a}$ or $\mathbf{b}$ is parallel to $\mathbf{1}$. It is also easy to see that the shift number (3.14) can always be found. That is, suppose $\mathbf{a}, \mathbf{b} \in \mathbb{R}^N$, and then there always exists a shift number $\tilde{\nu}$ minimizing $\|\mathbf{b} - \mathcal{K}(\tilde{\nu})^T \mathbf{a}\|_2$. $\|\mathbf{b} - \mathbf{c}\|_2^2$ for $\mathbf{c} \in \mathcal{M}(\mathbf{a})$ is a paraboloid on $\mathbb{R}^N$ restricted to a compact subset, so it yields a minimum in $\mathcal{M}(\mathbf{a})$. This minimum may not be unique, but the addition of a proper regularization term as in (2.32) will yield uniqueness for the problem.

Now, let us turn our attention to the dimension of the space spanned by vertices of $\mathcal{M}(\mathbf{a})$. The dimension depends on the periodicity of $\mathbf{a}$, in particular when the period of $\mathbf{a}$ is strictly smaller than N. In the presence of such smaller periods, also called isotropy [33], the minimization (3.9) can be further reduced; and the smaller the period of $\mathbf{a}$, the smaller the dimension should be. This is eventually related to the period of the functional (3.9), summarized in the next proposition and remarks that follow.

Note that a period of a function g defined on $\mathbb{R}$ is the smallest number $L > 0$ such that $g(x + L) = g(x)$ for all $x \in \mathbb{R}$.

Proposition 4.1. *Suppose we are given vectors $\mathbf{a}, \mathbf{b} \in \mathbb{R}^N$ both not parallel to $\mathbf{1}$. Defining $g(\tilde{\nu}) \equiv \mathbf{b} - \mathcal{K}(\tilde{\nu})^T \mathbf{a}$, let us denote by L the period of g, $\mathcal{F}$ the discrete Fourier transform, and gcd the greatest common divisor. Then*

$$(4.2) \qquad L = \frac{N}{\gcd\left[\operatorname{supp} \mathcal{F}(\mathbf{a}) \setminus \{0\}\right]}.$$

Proof. The proof is given in Appendix A. ∎

Given this dimension L in (4.2), we can reduce the discrete set W in (3.12) by considering only the first $2L - 1$ values. The proposition characterizes the isotropy in a discrete case, whose continuous version was mentioned but not detailed in [33].

In the variable speed case, the polygon $\mathcal{M}$ no longer retains its regularity. Let us define a variable speed counter part to (4.1),

$$(4.3) \qquad \mathcal{I}_{c,s}(\mathbf{a}) \equiv \{\mathcal{K}_c(\tilde{\nu})\mathbf{a} : \nu \in [s, s+1]\} \text{ and } \mathcal{M}_c(\mathbf{a}) \equiv \bigcup_{s \in \mathbb{Z}} \mathcal{I}_s = \{\mathcal{K}_c(\tilde{\nu})\mathbf{a} : \nu \in \mathbb{R}\}.$$

Mass is not preserved and $\mathcal{I}_{c,s}$ is not guaranteed to be convex. The variable speed introduces more vertices to the polygon $\mathcal{M}_c(\mathbf{a})$ and $\mathcal{I}_{c,s}(\mathbf{a})$ is itself now a union of more convex hulls. This makes the minimization problem (3.24) more challenging and causes the sharpening procedure (3.17) to run into difficulties, outside simple special cases for which one may impose more boundary conditions (3.16) near the interface.

5. Conclusion and future work. This paper introduced a greedy algorithm that extracts the transport structure from the snapshot matrix by building on the template fitting strategy. Extensions of the algorithm though the generalizations of the shift operators were also considered. Numerical experiments show that the algorithm can capture complex hyperbolic behaviors in examples where shocks and interfaces are present.

The objective of this approach is to construct a reduced order model of fully nonlinear hyperbolic problems, for use in high-dimensional applications arising in UQ and control design. In future work, the problem of postprocessing the output from the transport algorithm for use with existing projection-based model reduction methods will be investigated. Extension of the algorithm to the multidimensional setting is currently under development.

Appendices.

A. Proofs to Lemmas. Proofs of Lemmas 3.2 and 3.7 and Proposition 4.1 are given below.

Proof of Lemma 3.2.
(a) It follows from the definition since $\mathbf{I}, \mathbf{K}$ commute,

$$\mathbf{K}(\nu)\mathbf{K}(\omega) = (1 - \nu)(1 - \omega)\mathbf{I} + \nu(1 - \omega)\mathbf{K} + \omega(1 - \nu)\mathbf{K} + \nu\omega\mathbf{K}^2$$
$$= ((1 - \omega)\mathbf{I} + \omega\mathbf{K})\,((1 - \nu)\mathbf{I} + \nu\mathbf{K}) = \mathbf{K}(\omega)\mathbf{K}(\nu).$$

(b) Again from the definition,

$$\mathbf{K}(\nu)^T\mathbf{K}(\omega) = ((1 - \nu)\mathbf{I} + \nu\mathbf{K}^T)\,((1 - \omega)\mathbf{I} + \omega\mathbf{K})$$
$$= ((1 - \omega)\mathbf{I} + \omega\mathbf{K})\,((1 - \nu)\mathbf{I} + \nu\mathbf{K}^T) = \mathbf{K}(\omega)\mathbf{K}(\nu)^T.$$

(c) We have

$$\mathbf{K}(\nu)\mathbf{K}(\omega) = (1 - \nu - \omega)\mathbf{I} + (\nu + \omega)\mathbf{K} + \frac{\nu\omega}{N^2}\frac{\mathbf{I} - 2\mathbf{K} + \mathbf{K}^2}{1/N^2}.$$

The matrix in the last term, $\mathbf{I} - 2\mathbf{K} + \mathbf{K}^2 = \mathbf{K}(\mathbf{K}^T + \mathbf{K} - 2\mathbf{I}) = \frac{1}{N^2}\mathbf{K}\mathbf{L}_h$, is a shifted $\mathbf{L}_h$.

(d) This follows from the fact that the sum of the rows of $\mathbf{K}(\nu)$ is equal to $\begin{bmatrix} 1 & \cdots & 1 \end{bmatrix}$. ∎

Proof of Lemma 3.7. Consider the $(+)$ case, and the $(-)$ case follows similarly.

$$\nu_s^+ = \mathrm{argmin}_{\omega\in\mathbb{R}} \left\|\mathbf{b} - (\mathbf{K}(\omega)\mathbf{K}^s)^T\mathbf{a}\right\|_2^2 = \mathrm{argmin}_{\omega\in\mathbb{R}} \left\|\mathbf{b} - \mathbf{K}(\omega)^T\left(\mathbf{K}^T\right)^s\mathbf{a}\right\|_2^2.$$

Using this in formula (3.8),

$$\nu_s^+ = \frac{1}{2}\left(1 - 2\frac{\mathbf{b}^T\left(\mathbf{K} - \mathbf{I}\right)\left(\mathbf{K}^T\right)^s\mathbf{a}}{\mathbf{a}^T\left(\mathbf{K} + \mathbf{K}^T - 2\mathbf{I}\right)\mathbf{a}}\right).$$

Then we have

$$\nu_{s+1}^+ = \frac{1}{2}\left(1 - 2\frac{\mathbf{b}^T\mathbf{D}\left(\mathbf{K}^T\right)^{s+1}\mathbf{a}}{\mathbf{a}^T\mathbf{L}\mathbf{a}}\right) = \nu_s^+ + \frac{\mathbf{b}^T\mathbf{L}\left(\mathbf{K}^T\right)^s\mathbf{a}}{\mathbf{a}^T\mathbf{L}\mathbf{a}} = \nu_s^+ + \frac{(\mathbf{K}^s\mathbf{b})^T\mathbf{L}\mathbf{a}}{\mathbf{a}^T\mathbf{L}\mathbf{a}}.$$
∎

Proof of Proposition 4.1. We will call L a period of the vector $\mathbf{a}$ if L is the smallest number such that $a_{j+L} = a_j$, where the indices are computed modulo N.

It is easy to see that if $\mathbf{a}$ has period L, then $g(\tilde{\nu} + L) = g(\tilde{\nu})$.

Suppose g has period L, and then L must be an integer. If L has nonzero fractional part denoted by α, then

$$g(L) = \mathbf{b} - \left(\mathbf{K}^T\right)^{L-\alpha} \mathbf{K}(\alpha)^T \mathbf{a} = \mathbf{b} - \mathbf{a} = g(0).$$

So we must have that $(\mathbf{K}^T)^{L-\alpha}\mathbf{K}(\alpha)^T\mathbf{a} = \mathbf{a}$. Taking the 2-norm on both sides,

$$\left\|\left(\mathbf{K}^T\right)^{L-\alpha} \mathbf{K}(\alpha)^T\mathbf{a}\right\|_2^2 = \mathbf{a}^T\mathbf{K}(\alpha)^T\mathbf{K}(\alpha)\mathbf{a} = \mathbf{a}^T\mathbf{a} - \frac{\alpha(1-\alpha)}{N^2}\mathbf{a}^T\mathbf{L}\mathbf{a} > \|\mathbf{a}\|_2^2$$

for nonconstant $\mathbf{a}$ since $-\mathbf{L}$ is positive semidefinite with nullspace equal to that of constant vectors. Hence α cannot be in $(0,1)$, so it must be zero.

Now since L is an integer modulo N let us assume $L > 0$ without loss of generality, and then

$$\mathbf{b} - \left(\mathbf{K}^T\right)^{\ell} \mathbf{a} = \mathbf{b} - \left(\mathbf{K}^T\right)^{\ell+L} \mathbf{a} \quad \text{for some} \quad \ell < N,$$

which implies $\mathbf{K}^L\mathbf{a} = \mathbf{a}$, so $\mathbf{a}$ has period L. Also, since L is smallest number satisfying this equality, L divides N.

Therefore we only need to find the dividend, for which we simply take the discrete Fourier transform and compute the greatest common divisor of the nonzero frequencies. This yields (4.2). ∎

REFERENCES

[1] R. ABGRALL AND D. AMSALLEM, *Robust model reduction by L^1-norm minimization and approximation via dictionaries: Application to linear and nonlinear hyperbolic problems*, Adv. Model. Simul. Eng. Sci., 3 (2016), pp. 1–16.

[2] D. AMSALLEM AND C. FARHAT, *An interpolation method for adapting reduced-order models and application to aeroelasticity*, AIAA J., 46 (2008), pp. 1803–1813.

[3] D. AMSALLEM AND C. FARHAT, *An online method for interpolating linear parametric reduced-order models*, SIAM J. Sci. Comput., 33 (2011), pp. 2169–2198.

[4] D. AMSALLEM, M. ZAHR, Y. CHOI, AND C. FARHAT, *Design optimization using hyper-reduced-order models*, Struct. Multidisc. Optim., 51 (2015), pp. 919–940.

[5] N. AUBRY, W.-Y. LIAN, AND E. TITI, *Preserving symmetries in the proper orthogonal decomposition*, SIAM J. Sci. Comput., 14 (1993), pp. 483–505.

[6] G. BERKOOZ, P. HOLMES, AND J. LUMLEY, *The proper orthogonal decomposition in the analysis of turbulent flows*, Annu. Rev. Fluid. Mech., 25 (1993), pp. 539–575.

[7] G. BERKOOZ AND E. TITI, *Galerkin projections and the proper orthogonal decomposition for equivariant equations*, Phys. Lett. A, 174 (1993), pp. 94–102.

[8] K. CARLBERG, *Adaptive h-refinement for reduced-order models*, Internat. J. Numer. Methods Engrg., 102 (2015), pp. 1192–1210.

[9] E. CHRISTENSEN, M. BRONS, AND J. SORENSEN, *Evaluation of proper orthogonal decomposition-based decomposition techniques applied to parameter-dependent nonturbulent flows*, SIAM J. Sci. Comput., 21 (2000), pp. 1419–1434.

[10] J. Dawson, *Particle simulation of plasmas*, Rev. Modern Phys., 55 (1983).

[11] A. Deane, I. Kevrekidis, G. Karniadakis, and S. Orszag, *Low-dimensional models for complex geometry flows: Applications to grooved channels and circular cylinders*, Phys. Fluids A, 3 (1991), pp. 2337–2354.

[12] R. Everson and L. Sirovich, *The Karhunen–Loève procedure for gappy data*, J. Opt. Soc. Am., 12 (1995), pp. 1657–1664.

[13] J. C. Gower and G. B. Dijksterhuis, *Procrustes Problems*, Vol. 3, Oxford University Press, Oxford, 2004.

[14] F. Harlow, *A machine calculation method for hydrodynamic problems*, Los Alamos Scientific Laboratory report LAMS, 1955.

[15] S. Helgason, *The Radon transform on $\mathbb{R}^n$*, Integral Geometry and Radon Transforms, Springer, New York, 2011, pp. 1–62.

[16] P. Holmes, J. Lumley, and G. Berkooz, *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*, Cambridge University Press, Cambridge, 1996.

[17] P. Holmes, J. Lumley, G. Berkooz, J. Mattingly, and R. Wittenberg, *Low-dimensional models of coherent structures in turbulence*, Phys. Rep., 287 (1997), pp. 337–384.

[18] M. Kirby and D. Armbruster, *Reconstructing phase space from PDE simulations*, Z. Angew. Math. Phys., 43 (1992), pp. 999–1022.

[19] K. Kunisch and S. Volkwein, *Control of Burgers' equation by a reduced order approach using proper orthogonal decomposition*, J. Optim. Theory Appl., 102 (1999), pp. 345–371.

[20] R. J. LeVeque, *Large time step shock-capturing techniques for scalar conservation laws*, SIAM J. Numer. Anal., 19 (1982), pp. 1091–1109.

[21] R. J. LeVeque, *Convergence of a large time step generalization of Godunov's method for conservation laws*, Comm. Pure Appl. Math., 37 (1984), pp. 463–477.

[22] R. J. LeVeque, *A large time step generalization of Godunov's method for systems of conservation laws*, SIAM J. Numer. Anal., 22 (1985), pp. 1051–1073.

[23] R. J. LeVeque, *Finite Volume Methods for Hyperbolic Problems*, Cambridge University Press, Cambridge, 2002.

[24] Y. Maday and B. Stamm, *Locally adaptive greedy approximations for anisotropic parameter reduced basis spaces*, SIAM J. Sci. Comput., 35 (2013), pp. A2417–A2441.

[25] H. M. Park and M. W. Lee, *An efficient method of solving the Navier–Stokes equations for flow control*, Internat. J. Numer. Methods Engrg., 41 (1998), pp. 1133–1151.

[26] A. Paul-Dubois-Taine and D. Amsallem, *An adaptive and efficient greedy procedure for the optimal training of parametric reduced-order models*, Internat. J. Numer. Methods Engrg, 102 (2015), pp. 1262–1292.

[27] R. Pulch and D. Xiu, *Generalised polynomial chaos for a class of linear conservation laws*, J. Sci. Comput., 51 (2011), pp. 293–312.

[28] M. Rathinam and L. Petzold, *Dynamic iteration using reduced order models: A method for simulation of large scale modular systems*, SIAM J. Numer. Anal., 40 (2002), pp. 1446–1474.

[29] M. Rathinam and L. Petzold, *A new look at proper orthogonal decomposition*, SIAM J. Numer. Anal., 41 (2003), pp. 1893–1925.

[30] S. Ravindran, *Reduced-order adaptive controllers for fluid flows using proper orthogonal decomposition*, J. Sci. Comput., 15 (2000), pp. 457–478.

[31] J. Reiss, P. Schulze, J. Sesterhenn, and V. Mehrmann, *The Shifted Proper Orthogonal Decomposition: A Mode Decomposition for Multiple Transport Phenomena*, (2015), arXiv:1512.01985.

[32] C. W. Rowley, I. G. Kevrekidis, J. E. Marsden, and K. Lust, *Reduction and reconstruction for self-similar dynamical systems*, Nonlinearity, 4 (2003), pp. 1257–1275.

[33] C. W. Rowley and J. E. Marsden, *Reconstruction equations and the Karhunen–Loève expansion for systems with symmetry*, Phys. D, 142 (2000), pp. 1–19.

[34] C. W. Rowley, I. Mezić, S. Bagheri, P. Schlatter, and D. S. Henningson, *Spectral analysis of nonlinear flows*, J. Fluid Mech., 641 (2009), pp. 115–127.

[35] N. Smaoui and D. Armbruster, *Symmetry and the Karhunen–Loève analysis*, SIAM J. Sci. Comput., 18 (1997), pp. 1526–1532.

[36] D. SULSKY, Z. CHEN, AND H. L. SCHREYER, *A particle method for history-dependent materials*, Comput. Methods Appl. Mech. Engrg., 118 (1994), pp. 179–196.

[37] D. SULSKY, A. R. YORK, AND H. SCHREYER, *Fluid-membrane interaction based on the material-point method*, Internat. J. Numer. Methods Engrg., 48 (2000), pp. 901–924.

[38] J. H. TU, C. W. ROWLEY, D. M. LUCHTENBURG, S. L. BRUNTON, AND J. N. KUTZ, *On dynamic mode decomposition: Theory and applications*, J. Comput. Dyn., 1(2014), pp. 391–421.

[39] K. VEROY AND A. PATERA, *Certified real-time solution of the parametrized steady incompressible Navier–Stokes equations: Rigorous reduced-basis a posteriori error bounds*, Internat. J. Numer. Methods Fluids, 47 (2005), pp. 773–788.

[40] C. VILLANI, *Optimal Transport: Old and New*, Vol. 338, Springer, New York, 2008.

[41] K. WILLCOX, *Unsteady flow sensing and estimation via the gappy proper orthogonal decomposition*, Comput. & Fluids, 35 (2006), pp. 208–226.