Uncertainty Quantification Problems in Tsunami Modeling and Reduced Order Models for Hyperbolic Partial Differential Equations

Donsub Rim

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2017

Reading Committee:

Randall J. LeVeque, Chair

Gunther Uhlmann, Chair

Aleksandr Y. Aravkin

Frank I. González

Program Authorized to Offer Degree:
Applied Mathematics

University of Washington

Abstract

Uncertainty Quantification Problems in Tsunami Modeling and Reduced Order Models for Hyperbolic Partial Differential Equations

Donsub Rim

Co-Chairs of the Supervisory Committee:
Professor Randall J. LeVeque
Department of Applied Mathematics

Professor Gunther Uhlmann
Department of Mathematics

In this thesis, we consider an uncertainty quantification (UQ) problem that arises from tsunami modeling, namely the probabilistic tsunami hazard assessment (PTHA) problem. The goal of PTHA is to compute the probability of inundation at coastal communities, and the uncertainty originates from the unknown slip distribution of potential tsunamigenic earthquakes. First, we show that the Karhunen-Loève (K-L) expansion can be used to generate a wide range of random earthquake scenarios that represent this uncertainty well. Then we propose a multi-resolution approach to estimate the inundation: since it is computationally expensive to accurately estimate the inundation resulting from each scenario by using only fine-grid runs, many cheap coarse-grid runs are used instead to bulid an approximation.

For physical models that involve hyperbolic partial differential equations (PDEs), dimensionality reduction techniques such as the K-L expansion or multi-resolution approaches face limitations due to the fact that snapshot matrices built from solutions often exhibit slow decay in singular values, whereas fast decay is crucial for the success of many projection-based model reduction methods. To overcome this problem, we build on previous work in symmetry reduction [Rowley and Marsden, Physica D (2000), pp. 1-19] and propose an iterative

algorithm we call *transport reversal* that decomposes the snapshot matrix into multiple shifting profiles, each with a corresponding speed in 1D. Its applicability to typical hyperbolic problems is demonstrated through numerical examples, and other natural extensions that modify the shift operator are considered.

Transport or wave phenomena are much more complicated in multiple spatial dimensions, and in our approach to extend the transport reversal algorithm to higher dimensions it becomes crucial to generalize the large time-step (LTS) operators [LeVeque, SIAM J. Numer. Anal. (1985), pp.1051-1073]. For this purpose, we introduce a dimensional splitting method using the Radon transform, that enables the transport reversal introduced above for 1D to be extended to higher spatial dimensions. This dimensional splitting is of interest in its own right, and its applications to the solution of acoustic equation, absorbing boundary condition and displacement interpolation are illustrated. This splitting method requires inverting the Radon transform, and a method for inversion using conjugate gradient algorithm will be discussed.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I wish to express gratitude to my advisors Randall J. LeVeque and Gunther Uhlmann, for warmly welcoming me into the world of mathematics. Even with their sharp intellect and piercing insight, they would never aim to impress or intimidate, but instead seek tirelessly to inspire and encourage. I am forever indepted to their guidance and encouragement, first in mathematics then in character. Whatever good taste, if any, I have in either category, is due to their nurturing influence.

Furthermore, I owe any modicum of success in my graduate career to many incredible individuals who have imparted so much knowledge and perspective. I thank Ioana Dumitriu whose boundless optimism and humility I could only strive to emulate. I am most fortunate to have met François Monard, who became an instant role model. It was a great pleasure to work with Frank I. Gonźalez and Loyce M. Adams. I thank Steven L. Brunton, who encouraged and motivated my work. I thank Anne Greenbaum for always agreeing to chat about my half-formed ideas. I thank Bernard Deconinck for teaching me to become tenacious. I thank Tatiana Toro for showing me her masterful view of analysis.

Despite having been surrounded by such great teachers, the journey would still have been perilous without good friends. I thank Niket Thakkar, Ben Lansdell, Örn Arnaldsson, Scott Moe, Mauricio del Razo Sarmina, Daniel Shapero, Pedro Maia, Chris Vogl, Peng Zheng, Gerandy Brito Montes de Oca, Trevor Caldwell, Ben Segal, Lowell Thompson, Andreas Freund, Jeremy Upsal, Brian de Silva, and many others.

I wish also to thank Eun-Jae Park and Carsten Carstensen for setting me on this path.

Lastly, I wish to thank my family, whose kind love and support made it all a breeze.

Chapter 1

Introduction

Recent decades have seen impressive developments in numerical algorithms for computing solutions to various partial differential equations (PDEs) [20, 35, 64, 107]. The effort has been truly interdisciplinary, where mathematical analysis, physical insight, and software implementation all played indispensable roles. The current state-of-the-art numerical methods allow users to conduct computational experiments that lie beyond the reach of conventional experimental methods. For example, in probabilistic tsunami hazard assessment (PTHA) extreme earthquake scenarios are studied through numerical simulation to assess the inundation risk in coastal communities [42].

Computational experiments can be readily repeated, so scientists may go beyond performing a single computation for an accurate answer, to asking questions about the statistical properties of the result. Since many physical models have uncertain parameters associated with them, a single numerical solution for one fixed parameter value cannot provide a complete description. For example, in tsunami modeling the uncertainty can be related to the parameters for the earthquake event, involving geophysical constraints on the slip deficit or the fault geometry, etc. When such uncertainty is significant, statistical estimation becomes extremely important. This may appear to be in no way new, and rather a universal fact present in all scientific experiments. However, in computational experiments there is a novel aspect, namely the interplay between the statistical estimation of the quantity of interest (QoI) and the numerical model itself. This intricate relationship is examined closely in the rapidly expanding field called uncertainty quantification (UQ) [103, 117].

In this thesis, we are concerned mainly with an apparently simple relationship concerning

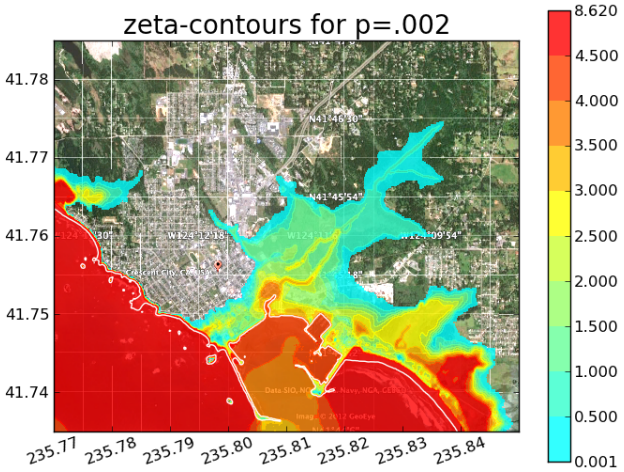

Figure 1.1: Probabilistic hazard map for Crescent City in Northern California, showing worst level of inundation in meters, for given annual probability of 0.2% (corresponding to 500-year return time). The map was constructed based on 15 earthquake scenarios. (This figure originally appeared in [44] and is reproduced here with permission.)

the computational cost. Statistial estimation of QoI often requires tens of thousands of samples to be accurate, and for each of these samples the numerical model must be run. Many realistic models tend to be computationally expensive, rendering the UQ task infeasible. Therefore, one needs to devise other approaches that allow the QoI to be obtained for each sample without requiring the evaluation of the expensive numerical model.

In particular, we will consider a UQ problem in which the underlying physical process is governed by nonlinear systems of hyperbolic PDEs, namely the probabilistic tsunami hazard assessment (PTHA) problem.

1.1 The probabilistic tsunami hazard assessment (PTHA) problem

The probabilistic tsunami hazard assessment (PTHA) problem aims to compute the probability of inundation at given spatial points of interest, which are usually chosen to be

geographical locations in coastal communities. This probability can be visually represented by hazard maps: an example of such a map is shown in Figure 1.1. Chapter 2 will be devoted to topics related to the PTHA, and the problem itself will be introduced in further detail in Section 2.3.

For our approach to PTHA, it is useful to assume a probability distribution over possible earthquake events. This distribution is inferred from the statistics of previous seismic events and other geophysical constraints (e.g., see [71]). Once the distribution is estimated, one next generates realistic earthquake scenarios whose statistical properties are consistent with the distribution. In Section 2.4, we show that the Karhunen-Loève (K-L) expansion can be applied to the spatial correlation of the slip distributions, then used to generate these random earthquakes [66]. Furthermore, the seafloor deformation caused by an earthquake acts as a low-pass filter so that higher-order terms in the K-L expansion can be judiciously neglected. This dramatically reduces the dimension of the random inputs.

In generating random earthquake scenarios for PTHA, it is important to cover extreme scenarios well, and we develop a simple methodology that accomplishes this via the K-L expansion. Furthermore, despite the dimensionality reduction provided by the truncation of the K-L expansion, the computational cost in executing the numerical model for tsunami propagation and inundation may still be significant. To alleviate this cost, we devise various *source-filtering* strategies. Given a set of extreme earthquake scenarios, we aim to select a smaller subset that can be used to approximate the hazard map resulting from the entire set. In particular, we describe a multi-resolution strategy in which cheap coarse-grid runs are used in place of expensive fine-grid runs. This multi-resolution strategy will be discussed in Section 2.5.

1.2 The transport reversal algorithm

Many practical problems in UQ such as the PTHA problem are computationally expensive to treat directly, often prohibitively so, due to the high dimensionality in the parameters. Naïve strategies require extensive computational resources, despite the improvements in the

efficiency of algorithms and the increases in computing power. Naturally, methods exploiting the low-dimensional structure hidden in the heavy computations become crucial. An approach of particular interest builds reduced order models (ROMs) of the high dimensional model (HDM) that are fast and accurate [10] then employs them in a Multi-Fidelity Monte Carlo (MFMC) framework [87], for example.

Projection-based methods for building ROMs have been successfully applied to UQ problems in computational fluid dynamics. Unfortunately, they are often not suitable for problems governed by hyperbolic PDEs. Projection-based ROMs typically project the numerical scheme to a smaller dimensional vector space of reduced-basis functions. However, even in a simple 1D setting, propagation of waves modeled by hyperbolic PDEs cannot be accurately captured by a small number of basis functions. This has been noted in recent literature [1, 21] where authors propose a dictionary-based method or a fail-safe method utilizing h-adaptivity. On the other hand, in an earlier work *template fitting* was used in the context of symmetry reduction [97, 98], in which the translation symmetry is removed from the solutions to the extent possible. The solution dynamics under this symmetry reduction is now called *slice dynamics*, and exhibits low-rank structure.

Template fitting is an elegant idea which can be adapted and generalized in order to deal with other typical behaviors that appear in the solutions to hyperbolic PDEs. These include extra-convective behaviors such as the presence of source terms, absorbing boundaries, nonlinearities or multiple wave speeds. Our recent work extends the template fitting to more general situations [96]. What we propose is an iterative algorithm, in which transport behavior is detected from the snapshots. It utilizes projections and cut-offs in a greedy manner until a given threshold of accuracy is achieved. Each iteration discovers a layer of transport structure, and their superposition yields a good approximation to the original snapshot. This algorithm reveals transport dynamics present in the snapshots. The output of the algorithm consists of multiple *pivots* that will be transported at varying speeds, *shift numbers* which encode the speeds, and *cut-off vectors* which are represented efficiently as Boolean arrays.

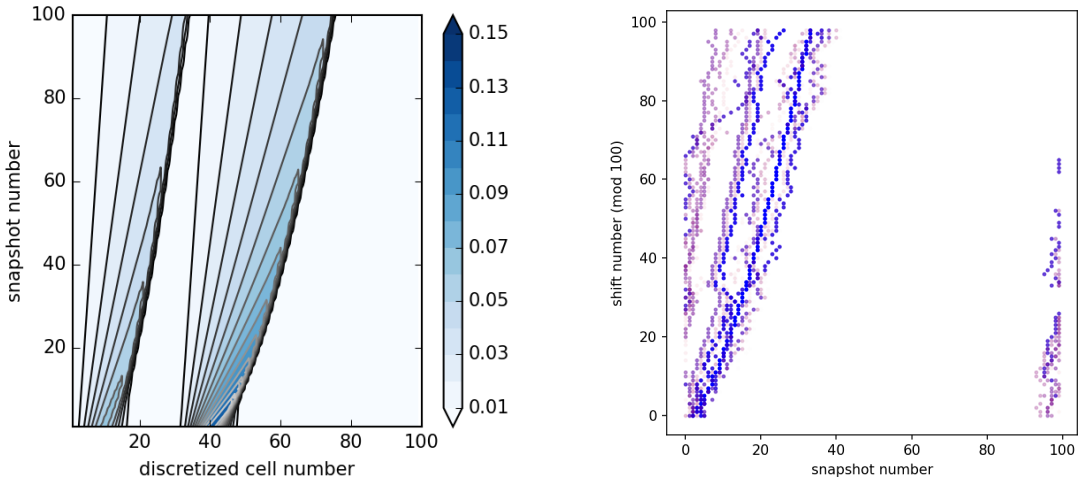

Figure 1.2: Snapshots from a solution of the Burger's equation (left) and the computed shift numbers (right). For further explanation see Chapter 3.

At the heart of the algorithm lies a minimization problem which is a direct generalization of the template fitting procedure, but with significant new features. An important ingredient is the addition of penalty terms which control the regularity of the shift numbers, enforcing the transport behavior to be smooth. The smoothness opens the possibility for *displacement interpolation* to be freely performed. This results in an improvement in accuracy compared to the usual linear interpolation, which is not appropriate for predicting transport phenomena.

The algorithm reflects a unique hyperbolic viewpoint of looking at the data. The singular value decomposition (SVD) extracts the singular vectors which best reflect the information from a *static* viewpoint. Transport reversal, in contrast, looks for information that is being transported across the state space. The extracted traveling information then can be used for *displacement interpolation* [113], as opposed to linear interpolation.

The details are provided in Chapter 3.

1.3 Dimensional splitting using the Radon transform

The transport reversal algorithm discussed in the previous section must be generalized to multi-dimensional settings to be useful for practical problems. This is a challenging problem which requires the development of a new tool. Since transport and wave phenomena are much more complicated in multiple spatial dimensions, template fitting can no longer effectively capture hyperbolic phenomena as in 1D.

In Chapter 4, we propose that the Radon transform be used in extending the transport reversal to the multi-dimensional setting. The Radon transform has a remarkable property, that it intertwines n-dimensional partial derivative with a univariate derivative. Thus transport and acoustic wave propagation in the spatial domain corresponds to a family of single-dimensional problems in the transformed domain. While this property is well-known [51, 80], the numerical analysis literature does not appear to exploit this property, to the best of our knowledge. Using the Radon transform, we obtain a new *dimensional splitting method*. This dimensional splitting is distinct from existing splitting methods, in that it allows a natural extension of the large time-step (LTS) method [60, 61, 62] to multiple spatial dimensions. It also has the potential to be useful as a dimensional splitting for fully nonlinear problems.

The splitting can be useful in extending transport reversal to multiple dimensions, since on the transformed domain one is left with 1D problems to which template fitting can still be effectively applied. In other words, the splitting transforms a multi-dimensional displacement interpolation problem to a family of single-dimensional ones.

Another compelling application of the dimensional splitting lies in the construction of absorbing boundaries. In many computational simulations, it is often desirable to impose absorbing boundaries. An obvious motivation is in simulating an infinite domain without incurring high computational cost. For the constant coefficient case, the absorbing boundary conditions (ABC) can be approximated [32]. Perfectly matched layers (PMLs) is now the popular method of choice for many problems, but still faces difficulties when a heterogenous

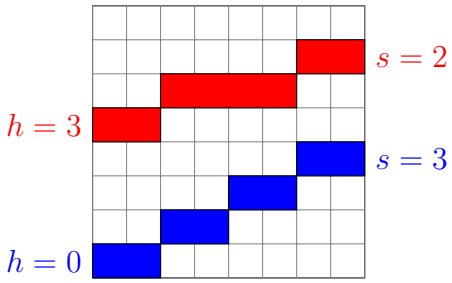

Figure 1.3: Two digital lines (or d-lines) on a 8×8 image determined by two parameters h and s.

medium extends to the boundary [84]. Dimensional splitting with the Radon transform can be useful in this context, as it is a decomposition into 1D planar waves, and therefore as a consequence each 1D wave can be treated separately at the boundary.

There are many different discretizations of the Radon transforms and its inverse. Among these, Filtered Backprojection (FBP) is arguably the most widely known [80]. In FBP, the high-frequency content is filtered during inversion, hence the sharp features are not preserved, making it unsuitable for nonlinear problems which exhibit shock formation. Instead, we turn to a completely discrete analogue, the Discrete Radon Transform (DRT), introduced in [18, 45]. DRT uses digital lines, or d-lines, rather than straight lines. Integration is replaced by simple sums along on these d-lines. Moreover, it is fast: d-lines are defined recursively so that Radon transform of a N-by-N image, as well as its backprojection, can be computed with $\mathcal{O}(N^2 \log N)$ floating point operations.

1.4 Outlook and future work

This thesis offers the groundwork for constructing ROMs for hyperbolic PDEs, but also leads to many open problems. There are still substantial challenges to be overcome before the ROMs can be freely constructed. For instance, the utilization of low-dimensional transport structures obtained through transport reversal from a parametrized PDE requires a more careful consideration of the dependence on parameters, even in the simple 1D setting. Moreover, the dimensional splitting via the Radon transform requires fast and optimized

implementation to be useful for practical problems. Nonetheless, there are concrete lines of approach for these issues, and they will be discussed in Chapter 5.

Chapter 2

The Probabilistic Tsunami Hazard Assessment (PTHA) Problem

In this chapter, we will first give an overview of a tsunami propagation model and the probabilistic tsunami hazard assessment (PTHA) problem in Sections 2.1, 2.2 and 2.3. Then we will discuss how the Karhunen-Loève (K-L) expansion could be used to generate random earthquakes for PTHA in Section 2.4, and propose a multi-resolution strategy that reduces the computational cost for generating hazard maps in Section 2.5.

2.1 Tsunami propagation model

Tsunamis are generated by earthquakes that deform the seafloor bathymetry and displace the column of seawater resting above it [65]. Naturally, the earthquake itself is an important component in the tsunami propagation model, and we will parameterize an earthquake with a simple rectangular fault-slip model. The rectangular fault geometry can be extended to triangles or general polygons allowing for a more flexible representation [27, 75, 106], but we will use rectangular faults throughout this thesis for simplicity. We will also restrict our discussion to instantaneous slip, as opposed to time-dependent slip: we will assume that the earthquake fault slips and deforms the seafloor instantaneously.

An illustration of a rectangular fault is shown in Figure 2.1. The fault is then subdivided into smaller rectangular *subfaults*, up to a desired resolution. Then we may assign the magnitude of the displacement (the *slip*) on each subfault, at some angle to the strike direction (the *rake*). A vector of real numbers designating the slip to each subfault will be called a *slip distribution*. Since we assumed that the fault will slip instantaneously, a slip distribution fully describes an earthquake event in our simplified setting. An example of a rectangular

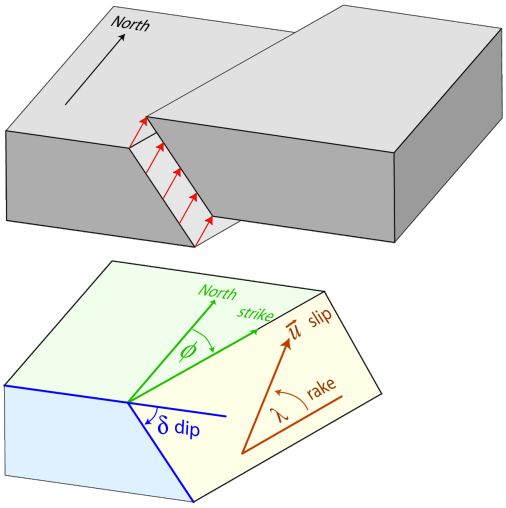

Figure 2.1: Rectangular fault oriented by angles strike, dip and rake.

fault geometry and a slip distribution is shown to the left in Figure 2.2.

The earthquake slip underneath the ocean causes a deformation of the seafloor. This is computed by the Okada model [82], and an example of a resulting deformation is shown to the right in Figure 2.2. Note that an equivalent computation can be done for triangular- or polygonal- shaped fault geometry with the model of Comninou and Dunders [27]. Both of these deformation models use Green's function formula for the homogenous linear elasticity equation for a half-space with free boundary conditions. This seafloor deformation displaces the water column lying on top. In this work, the deformation is assumed to equal the displacement of the top surface of the ocean, but in reality the dynamics of the seafloor deformation is a more complicated phenomenon involving solid-fluid interaction between the fluid mechanics and acoustics of the water and the seismic wave propagation that can be modeled with fully 3D simulations [114].

The tsunami propagation itself is modeled by the depth-integrated 2D shallow water equations with a source term. Let $\Omega \subset \mathbb{R}^2$ denote a rectangular domain large enough to

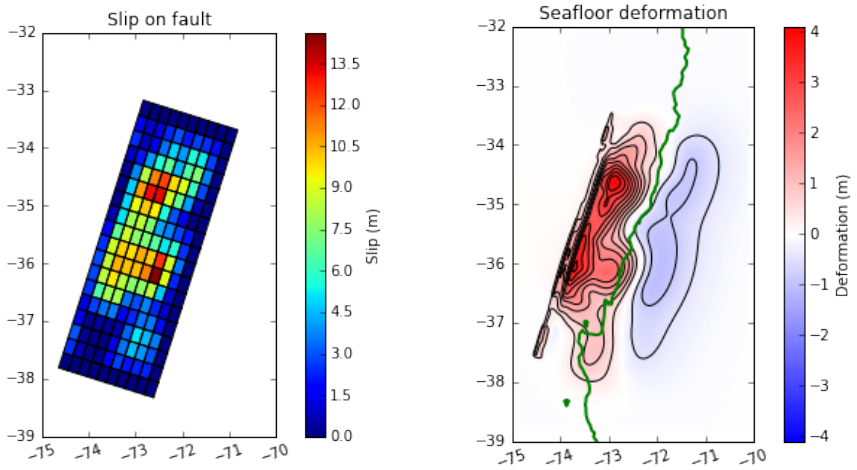

Figure 2.2: An example of slip distributed on a fault plane, from the USGS inversion of the 27 February 2010 event off Maule, Chile [110] (left), and the resulting sea floor deformation computed using the Okada model, with the coast line in green (right).

contain the surrounding geographical area, then the PDE is given by the system

$$\begin{bmatrix} h \\ hu \\ hv \end{bmatrix}_t + \begin{bmatrix} hu \\ hu^2 + \frac{1}{2}gh^2 \\ huv \end{bmatrix}_x + \begin{bmatrix} hv \\ huv \\ hv^2 + \frac{1}{2}gh^2 \end{bmatrix}_y = \begin{bmatrix} 0 \\ -ghB_x - Dhu \\ -ghB_y - Dhv \end{bmatrix} \quad \text{in } \Omega, \tag{2.1.1}$$

where h is the water height, u and v the velocity in x and y direction, respectively, and B the bathymetry. Also, g the gravitational constant and D is the drag coefficient

$$D(h, u, v) = n^2 g h^{-4/3} \sqrt{u^2 + v^2}. \tag{2.1.2}$$

The parameter n is the *Manning coefficient* which depends on the roughness of the bottom floor. A constant value of $n = 0.025$ is used here, following standard practice. We also will denote by η the sea-surface displacement $h + B$.

Longitude	Latitude	Depth	Strike	Length	Width	Dip
235.77	40.46	11.00	358.00	70.00	41.34	14.00
235.69	41.09	11.00	358.00	70.00	44.45	13.00
235.30	40.53	1.00	358.00	55.00	41.34	14.00
235.18	41.09	1.00	358.00	70.00	44.45	13.00
235.54	41.98	11.00	358.00	126.00	52.41	11.00
234.93	41.97	1.00	358.00	126.00	52.41	11.00
235.42	43.12	11.00	359.00	126.00	63.92	9.00
234.73	43.11	1.00	359.00	126.00	56.85	9.00

Table 2.1: Specification of the rectangular subfaults of the southern portion of Cascadia Subduction Zone (CSZ). Units are in degrees.

2.2 Numerical experiments on Cascadia Subduction Zone (CSZ)

We will be primarily concerned with earthquake events which occur in the southern portion of the Cascadia Subduction Zone (CSZ) shown in Figure 2.3. This fault geometry is parametrized by rectangular subfaults as designated in Table 2.1. Each of the rectangular fault is further subdivided into smaller approximately 8 km $\times$ 8 km square-shaped faults. After the subdivision, we obtain 540 such subfaults. An earthquake event on this fault geometry can be represented by a vector $\mathbf{s}$ of length 540. Suppose an earthquake event with given slip distribution $\mathbf{s} \in \mathbb{R}^{540}$ occurs, as shown on the left of Figure 2.4. As discussed in the previous section, this causes the seafloor deformation on the right of Figure 2.4. This in turn causes the sea surface displacement displayed on the left of Figure 2.5. The deformation is computed using the vertical displacement component from the Okada model at $1'$ resolution. ($1'$ in longitude is about 1.37 km at 42° latitude, and $1'$ in latitude is roughly 1.85 km.)

This yields the initial condition to the shallow water equations (2.1.1). The solution is computed numerically using GEOCLAW. The computational domain for the tsunami propagation is $\Omega = [-127.5^\circ, -123.5^\circ] \times [38.5^\circ, 44.5^\circ]$ initialized with $30' \times 30'$ square cells, and the propagation is computed through adaptive mesh refinement (AMR) as described in [13]. We use two different resolutions to compute the solutions, one with coarse-grid resolution,

	Fine grid		Coarse grid	
Level	Refinement ratio	Cell-length	Refinement ratio	Cell-length
1		$30'$		$30'$
2	6	$5'$	6	$5'$
3	5	$1'$	5	$1'$
4	6	$10''$	6	$10''$
5	5	$2''$		
6	3	$2/3''$		

Table 2.2: Two different sets of adaptive mesh refinement levels.

another with fine-grid resolution. The precise levels of refinement are displayed in Table 2.2

The initial sea-surface elevation η is shown on left of Figure 2.5. The tsunami then propagates towards the coast, as shown on the right of Figure 2.5, then inundates Crescent City, as displayed in Figure 2.6.

2.3 Probabilistic tsunami hazard assessment (PTHA)

The objective of probabilistic tsunami hazard assessment (PTHA) is to estimate the risk of tsunami hazards. In particular, we aim to estimate the probability that the maximum inundation at a particular geographical location in a coastal community will exceed some level. Hazard maps that indicate the annual probability of flooding exceeding various depths can provide much more information than a single worst-case inundation map [2, 36, 37, 43, 55, 69, 116]. PTHA, like probabilistic seismic hazard assessment (PSHA) which has a longer history, is an important tool in the development of emergency planning, as well as in the design of building codes able to withstand seismic or tsunami forces [23].

Here we focus on the inundation risk, represented by the flow depth, of onshore locations near Crescent City. We will not consider far-field scenarios and only consider earthquakes which occur on the southern portion of CSZ (Figure 2.3) as described in the previous section.

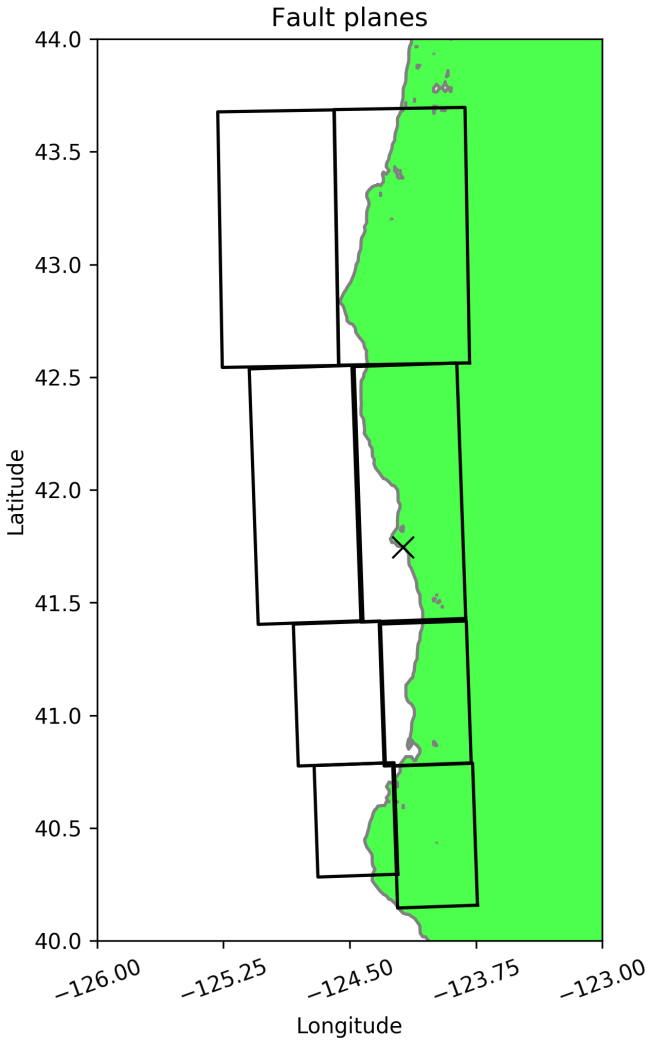

Figure 2.3: Fault planes in the southern portion of the Cascadia Subduction Zone. The X marks the location of Crescent City, $(-124.1838°, 41.7456°)$ in longitude-latitude coordinates.

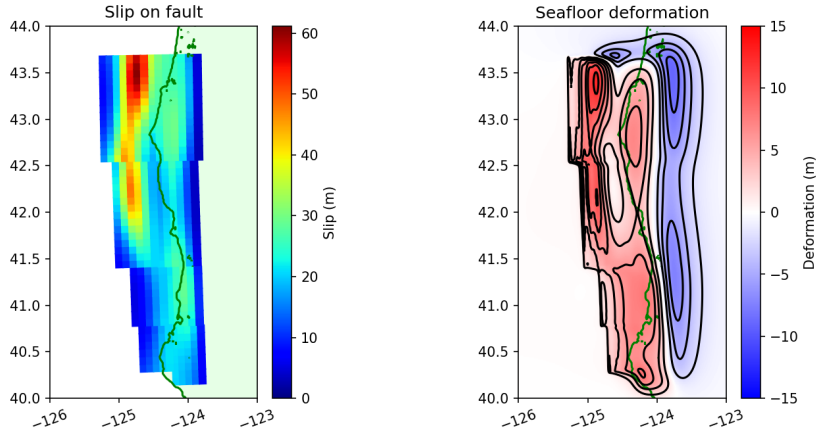

Figure 2.4: A hypothetical earthquake event with a given slip distribution (left) and the resulting seafloor deformation (right).

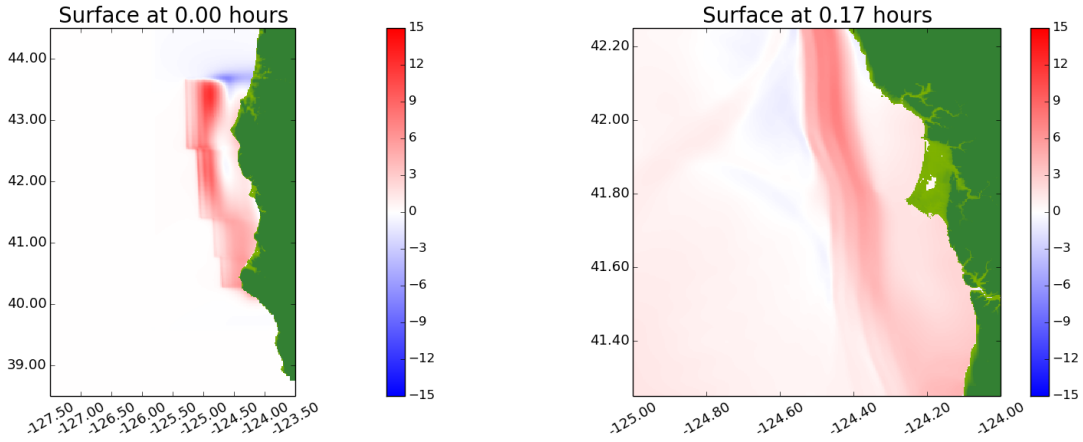

Figure 2.5: Initial sea surface elevation η (2.1.1) at $t = 2$s shown on the entire computational domain (left) and zoomed view of the tsunami propagating at $t = 0.17$h towards the coast (right).

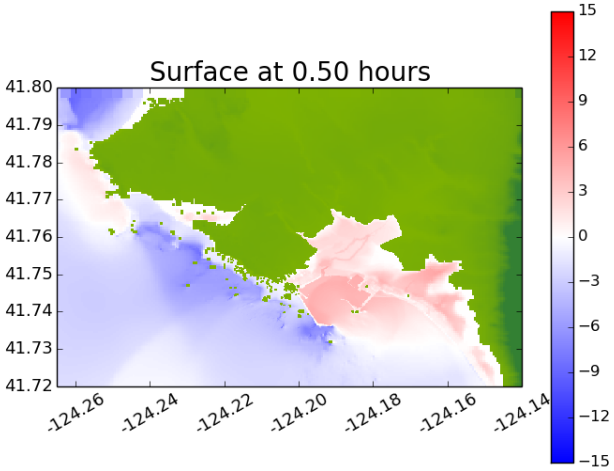

Figure 2.6: Inundation caused by the tsunami propagation shown in Figure 2.5 at $t = 0.50$h.

We will denote by ζ the maximum observed value over the full time period of a tsunami of either h or $B + h$ depending on the location. Recall that B is the bathymetry (2.1.1), and the set of points

$$\{(x, y) \in \Omega : B(x, y) > 0\} \tag{2.3.1}$$

are points initially onshore relative to the datum of Digital Elevation Model (DEM), usually Mean High Water (MHW). In an earthquake event, B itself undergoes change, therefore we must distinguish the bathymetry before and after the event. Since we assumed the deformation to occur instantaneously (Section 2.1) we only need to designate the initial bathymetry before the event by B_0.

$$\zeta(x, y) = \begin{cases} \max_t h(x, y, t) & \text{if } B_0(x, y) > 0 \quad \text{(the flow depth)}, \\ \max_t (h + B)(x, y, t) & \text{if } B_0(x, y) \leq 0 \quad \text{(sea surface elevation)}. \end{cases} \tag{2.3.2}$$

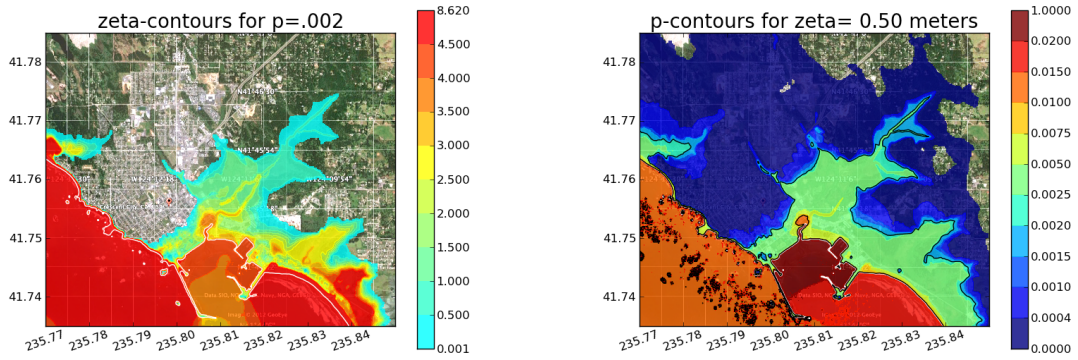

Figure 2.7: Examples of a zeta-contour (left) and a p-contour (right), drawn in the PTHA report for Crescent City. These maps appeared originally in [44] and is reproduced here with permission.

In PTHA, we wish to compute the *exceedance probability* given a threshold ζ_e,

$$P[\zeta > \zeta_e]. \tag{2.3.3}$$

This probability depends on two parameters, the spatial variable (x, y) and the given threshold ζ_e. One can represent this value by drawing a contour map of $P[\zeta > \zeta_e]$ for fixed values ζ_e, which we call a *probability-contour* or a *p-contour*. Another way to represent this information is to fix a probability $p \in (0, 1]$ then find the corresponding ζ_e value,

$$\zeta_p(x, y) = \max \{\zeta_e : P[\zeta > \zeta_e] \geq p\}. \tag{2.3.4}$$

Thus one can draw contour maps of ζ_p for each $p \in (0, 1]$. We call these contour maps *zeta-contours*. Examples of p- and zeta-contours are shown in Figure 2.7.

In practice, the probability (2.3.3) must be computed through simulations. We write this formally as follows. Suppose we are given a set $\mathcal{S}$ of earthquake scenarios, in which each

scenario is represented by the pair $(\mathbf{s}, p)$,

$$\mathcal{S} = \{(\mathbf{s}_j, p_j) : j = 1, \cdots, M\}. \tag{2.3.5}$$

$\mathbf{s}$ is a slip distribution parametrizing the earthquake event, and p is the annual probability associated with the event. Let us denote by $\zeta(\mathbf{s})$ the ζ value computed for the event with earthquake designated by $\mathbf{s}$. Also define the indicator function $I[\zeta(\mathbf{s}); \zeta_e]$ so that

$$I[\zeta(\mathbf{s}); \zeta_e] = \begin{cases} 1 & \text{if } \zeta(\mathbf{s}) > \zeta_e \\ 0 & \text{otherwise.} \end{cases} \tag{2.3.6}$$

Then the exceedance probability can be written in a computable form,

$$P[\zeta > \zeta_e] \approx \sum_{(\mathbf{s},p)\in\mathcal{S}} p\, I[\zeta(\mathbf{s}); \zeta_e]. \tag{2.3.7}$$

Therefore, if one could devise a set of scenarios $\mathcal{S}$ that represents the probability distribution of possible scenarios well, the computations (2.3.7) will be straightforward. However, finding the suitable set of scenarios $\mathcal{S}$ must also take into consideration the computational cost, since for each element in $\mathcal{S}$, a run of GEOCLAW must be performed. Furthermore, the probabilities of the form (2.3.3) are often used to model rare events, and are difficult to estimate; the region $\{\zeta > \zeta_e\}$ diminishes as ζ_e increases, and therefore one requires many scenarios to characterize the low-probability-high-risk events well. Statistical methods such as importance sampling [49] are needed to alleviate the computational burden.

This problem of finding the suitable set $\mathcal{S}$, depending on the fault geometry and probability distribution over potential slips, is discussed in the following sections.

2.4 Generating random earthquakes

In this section, we will introduce a methodology to generate hypothetical earthquakes with desired statistical properties, with a special focus on their use for tsunami modeling. The

method generates random earthquakes efficiently and is directly useful for PTHA, which requires performing tsunami simulations for many such earthquake scenarios. The set of all such scenarios was denoted by $\mathcal{S}$ in the previous section (2.3.5), and plays a direct role in assessing the risk arising from rare events. This is clear from the approximation (2.3.7). Generating a large number of hypothetical events can also be useful for testing inversion methods that incorporate tsunami data, such as the current DART buoy network or proposed early warning systems [76, 77]. This section is based on the joint publication with R. J. LeVeque, K. Waagan, F. I. González, and G. Lin [66].

The primary goal of this section is to introduce a general approach to generating hypothetical earthquakes, by producing random slip patterns on a pre-specified fault geometry. Similar techniques have been used in past studies, particularly for the generation of seismic waves in probabilistic seismic hazard assessment (PSHA), which has a longer history than PTHA. This is often done via a spectral representation of the slip pattern as a Fourier series with random coefficients that decay at a specified rate based on the desired smoothness and correlation length of the slip patterns (e.g., [40, 48, 59, 71, 78]). A random Fourier series can be trimmed down to the desired non-rectangular fault geometry, possibly with some tapering to zero slip at some edges of the fault. Different correlation lengths can be assumed in the strike and slip directions if the fault is rectangular and these directions are used as the horizontal coordinates in the Fourier representation.

Our approach is essentially the same on a rectangular fault but generalizes easily to other fault geometries by using a Karhunen-Loève expansion. This work was motivated in particular by the need to model events on the curving CSZ, which lies offshore North America and runs nearly 1200 km from Northern California up to British Columbia.

Recall that the fault is subdivided into many rectangular subfaults and a value of the slip s_i is assigned to the ith subfault. If here are N subfaults, then this defines a vector $\mathbf{s} \in \mathbb{R}^N$. Initially we assume that the moment magnitude M_w of the earthquake (which depends on the total slip summed over all subfaults) has been prescribed, and also that the desired mean

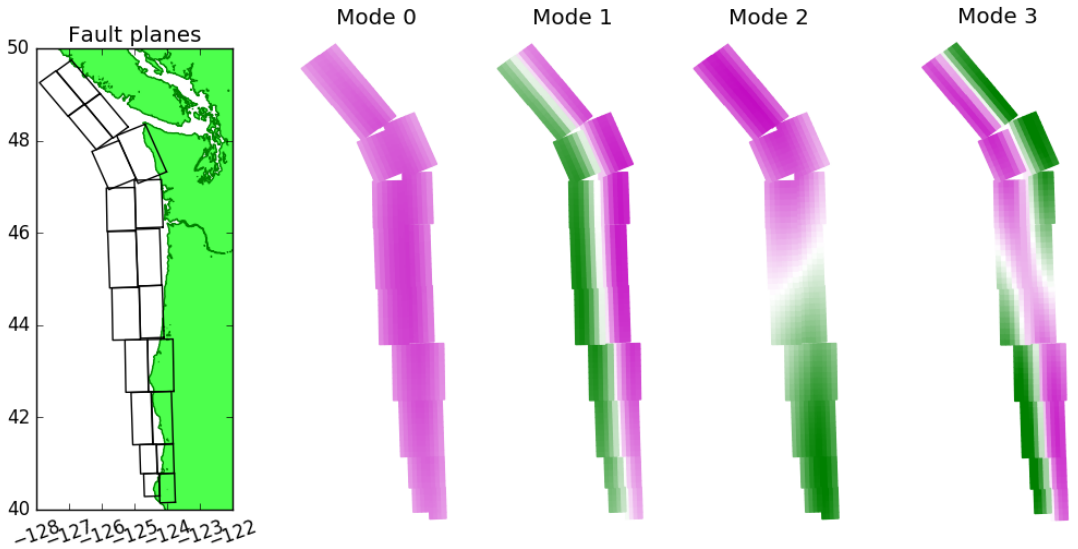

Figure 2.8: Subdivision of the Cascadia Subduction Zone into 20 subfaults. These are further divided into 865 subfaults to compute the modes shown, which are the first four eigenvectors of the 865×865 covariance matrix as might be used in a Karhunen-Lòeve expansion. Magenta and green are used to indicate positive and negative entries in the eigenmodes.

slip $\boldsymbol{\mu} \in \mathbb{R}^N$ and covariance matrix $\hat{\mathbf{C}} \in \mathbb{R}^{N \times N}$ are known. The mean slip is a vector with components $\mu_i = E[s_i]$, the expected value of the slip on the ith subfault, and the $N \times N$ covariance matrix $\hat{\mathbf{C}}$ has components $\hat{C}_{ij} = E[(s_i - \mu_i)(s_j - \mu_j)]$, which can also be expressed as the outer product $\hat{\mathbf{C}} = E[(\mathbf{s} - \boldsymbol{\mu})(\mathbf{s} - \boldsymbol{\mu})^T]$, where T denotes transposing the vector.

The *Karhunen-Loève (K-L) expansion* (e.g. [39, 54, 101]) is a standard approach to representing a random field as a linear combination of eigenvectors of the presumed covariance matrix $\hat{\mathbf{C}}$. If the matrix $\hat{\mathbf{C}}$ has eigenvalues λ_k (ordered with $\lambda_1 > \lambda_2 > \cdots > 0$) and corresponding eigenvectors $\mathbf{v}_k$, then the K-L expansion expresses the slip vector $\mathbf{s}$ as

$$\mathbf{s} = \boldsymbol{\mu} + \sum_{k=1}^{N-1} z_k \sqrt{\lambda_k} \mathbf{v}_k. \tag{2.4.1}$$

where the z_k are *independent* normally distributed random numbers $z_k \sim \mathcal{N}(0, 1)$ with mean 0 and standard deviation 1. This is described in more detail in Section 2.4.1 where we explain why this gives random slip patterns with the desired mean and covariance. This expansion makes it easy to generate an arbitrary number of realizations using standard software to generate $\mathcal{N}(0, 1)$ random numbers.

Figure 2.8 shows an example of the first four eigenmodes for the CSZ using this approach, where the N components of each eigenvector are represented on the fault geometry using a color map in which magenta is positive and green is negative. Note that Mode 0 is roughly constant over the fault, so adding a multiple of this mode modifies the total slip and hence the magnitude M_w. On the other hand, the other modes have both positive and negative regions and so adding a multiple of any of these tends to redistribute the slip (e.g., up-dip / down-dip with Mode 1 or between north and south with Mode 2). As with Fourier series, higher order eigenmodes are more oscillatory.

If the presumed correlation lengths are long and the covariance is a sufficiently smooth function of the distance between subfaults, then the eigenvalues λ_k decay rapidly (there is little high-frequency content) and so the K-L series can often be truncated to only a few terms, greatly reducing the dimension of the stochastic space that must be explored.

The K-L series approach could also be used to generate random slip patterns for generating seismic waves, e.g. for performing PSHA or testing seismic inversion algorithms. In this case high-frequency components of the slip are very important and the K-L expansion may not decay so quickly. However, for tsunami modeling applications the slip pattern on the fault is only used to generate the resulting seafloor deformation. This is a smoothing operation that suppresses high frequencies. Here we also explore this effect and show that truncating the expansion to only a few terms may be sufficient for many tsunami applications. Reducing the dimension of the stochastic space is important for efficient application of many sampling techniques that could be used for PTHA analysis.

In this section we focus on explaining the key ideas in the context of a one-dimensional fault model (with variation in slip only in the down-dip direction) and a two-dimensional example using the southern portion of the CSZ. However, we do not claim to have used the optimal parameters for modeling this particular fault.

The K-L expansion as described above generates a Gaussian random field, in which each subfault slip s_i has a normal distribution with mean μ_i and variance $\hat{C}_{ii}$ and together they have a joint normal distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\hat{\mathbf{C}}$. A potential problem with this representation is that when the variance is large it is possible for the slip s_i to be negative on some subfaults. Since we assume the rake is constant (e.g. 90 degrees for a subduction thrust event), this would correspond to subfaults that are slipping in the wrong direction. The same issue arises with Fourier series representations and can be dealt with by various means, for example by simply setting the slip to zero any place it is negative (and then rescaling to maintain the desired magnitude). This naturally changes the statistics of the resulting distributions.

Another approach is to instead posit that the random slip can be modeled by a joint lognormal distribution, for which the negative values are not allowed. Random slip patterns with a joint lognormal distribution can be generated by using the K-L expansion to first compute a Gaussian field and then exponentiating each component of the resulting vector to obtain the slip on each subfault. By choosing the mean $\boldsymbol{\mu}^g$ and covariance matrix $\hat{\mathbf{C}}^g$ for the

Gaussian field properly, the resulting lognormal will have the desired mean $\boldsymbol{\mu}$ and $\hat{\mathbf{C}}$ for the slip. This is discussed in Section 2.4.2 and used in the two-dimensional example in Section 2.4.3.

2.4.1 Expressing slip using a Karhunen-Loève expansion

If the earthquake fault is subdivided into N small rectangular subfaults with fixed geometry, then a particular earthquake realization can be described by specifying the slip on each subfault, i.e. by a vector $\mathbf{s} \in \mathbb{R}^N$ where s_i is the slip on the ith subfault. Note that we are assuming that only the slip varies from one realization to another; the geometry and rake (direction of slip on each subfault) are fixed, and the slip is instantaneous and not time-dependent. These restrictions could be relaxed at the expense of additional dimensions in our space of realizations.

Initially assume we wish to specify that the slip is a Gaussian random field with desired mean slip $\boldsymbol{\mu} \in \mathbb{R}^N$ and covariance matrix $\hat{\mathbf{C}} \in \mathbb{R}^{N\times N}$, which we write as $\mathbf{s} \sim \mathcal{N}(\boldsymbol{\mu}, \hat{\mathbf{C}})$. Then we compute the eigenvalues λ_k of $\hat{\mathbf{C}}$ and corresponding normalized eigenvectors $\mathbf{v}_k$ so that the matrix of eigenvectors $\mathbf{V}$ (with k-th column $\mathbf{v}_k$) and diagonal matrix of eigenvalues $\boldsymbol{\Lambda}$ satisfy $\hat{\mathbf{C}} = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^T$. Note that the covariance matrix is symmetric postive definite, so the eigenvalues are always positive real numbers and the eigenvectors can be chosen to be orthonormal, $\mathbf{V}^T\mathbf{V} = \mathbf{I}$.

Then the K-L expansion (2.4.1) can be written in matrix-vector form as

$$\mathbf{s} = \boldsymbol{\mu} + \mathbf{V}\boldsymbol{\Lambda}^{1/2}\mathbf{z}, \tag{2.4.2}$$

where $\mathbf{z} \in \mathbb{R}^N$ is a vector of independent identically distributed $\mathcal{N}(0,1)$ random numbers. Realizations generated via the K-L expansion have the right statistics since we can easily

compute that $E[\mathbf{s}] = \boldsymbol{\mu}$ (since $E[\mathbf{z}] = 0$) and

$$\begin{aligned} E[(\mathbf{s}-\boldsymbol{\mu})(\mathbf{s}-\boldsymbol{\mu})^T] &= E[\mathbf{V}\boldsymbol{\Lambda}^{1/2}\mathbf{z}\mathbf{z}^T\boldsymbol{\Lambda}^{1/2}\mathbf{V}^T] \\ &= \mathbf{V}\boldsymbol{\Lambda}^{1/2}E[\mathbf{z}\mathbf{z}^T]\boldsymbol{\Lambda}^{1/2}\mathbf{V}^T \\ &= \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^T = \hat{\mathbf{C}} \end{aligned} \tag{2.4.3}$$

using the fact that $\mathbf{V}$ and $\boldsymbol{\Lambda}$ are fixed and $E[\mathbf{z}\mathbf{z}^T] = \mathbf{I}$. Note that the $\mathbf{z}$ could be chosen from a different probability density with mean 0 and covariance matrix $\mathbf{I}$ and achieve the same covariance matrix $\hat{\mathbf{C}}$ with the K-L expansion, although the $\mathbf{s}$ would not have a joint normal distribution in this case.

2.4.2 Lognormally distributed slip

If we wish to generate slip realizations that have a joint lognormal distribution with a desired mean and covariance matrix, rather than a joint Gaussian distribution, we can first generate realizations of a joint Gaussian random field and then exponentiate each component. This approach will be used in the two-dimensional example below in Section 2.4.3.

In this case we first choose the desired mean $\boldsymbol{\mu}$ and covariance matrix $\hat{\mathbf{C}}$ for the slip, and then compute the necessary mean $\boldsymbol{\mu}^g$ and covariance matrix $\hat{\mathbf{C}}^g$ for the Gaussian to be generated by the K-L expansion, using the fact that if g is a random variable from $\mathcal{N}(\boldsymbol{\mu}^g, \hat{\mathbf{C}}^g)$, then $\exp(g)$ is lognormal with mean and covariance matrix given by:

$$\mu_i = \exp(\mu_i^g + \hat{C}_{ii}^g/2), \qquad \hat{C}_{ij} = \mu_i\mu_j(\exp(\hat{C}_{ij}^g) - 1). \tag{2.4.4}$$

Hence we can solve for

$$\begin{aligned} \hat{C}_{ij}^g &= \log(\hat{C}_{ij}/\mu_i\mu_j + 1), \\ \mu_i^g &= \log(\mu_i) - \frac{1}{2}\hat{C}_{ii}^g. \end{aligned} \tag{2.4.5}$$

We now find the eigenvalues λ_k and eigenvectors $\mathbf{v}_k$ of $\hat{\mathbf{C}}^g$. To generate a realization we

choose N values $z_k \sim \mathcal{N}(0,1)$ and then form the K-L sum

$$\mathbf{s}^g = \boldsymbol{\mu}^g + \sum_{k=1}^{N-1} z_k \sqrt{\lambda_k} \mathbf{v}_k. \tag{2.4.6}$$

We then exponentiate each component of $\mathbf{s}^g$ to obtain the slip values, which then have the desired joint lognormal distribution [38].

As described, this will generate realizations with total slip (and hence magnitude M_w) that vary around the mean. As in the Gaussian case, we can drop the nearly-constant $\mathbf{v}_0$ term from the sum to reduce this variation. We can also generally truncate the series to a much smaller number of terms and still capture most of the variation if the eigenvalues are rapidly decaying.

Now consider the special case where we make the same assumptions as in Section 2.4.1 that $\hat{C}_{ij} = \sigma_i \sigma_j C_{ij}$ where C is the desired correlation matrix and $\sigma_i = \alpha \mu_i$, while the mean μ_i was given by some some taper τ_i scaled by a scalar value $\bar{\mu}$. Then computing $\boldsymbol{\mu}^g$ and $\hat{\mathbf{C}}^g$ according to (2.4.5), we find that:

$$\begin{aligned} \hat{C}^g_{ij} &= \log(\alpha^2 C_{ij} + 1), \\ \mu^g_i &= \log(\bar{\mu}\tau_i) - \frac{1}{2}\log(\alpha^2 + 1) \end{aligned} \tag{2.4.7}$$

We see that the covariance matrix in this case depends only on the correlation matrix and the scalar α, not on the mean slip itself (and in particular is independent of the taper). We also find that $\exp(\mu^g_i) = \bar{\mu}\tau_i / \sqrt{\alpha^2 + 1}$ is simply a scalar multiple of the taper.

Using these assumptions and the fact that

$$\exp\left(\boldsymbol{\mu}^g + \sum_{k=1}^{N-1} z_k \sqrt{\lambda_k} \mathbf{v}_k\right) = \exp(\boldsymbol{\mu}^g) \exp\left(\sum_{k=1}^{N-1} z_k \sqrt{\lambda_k} \mathbf{v}_k\right),$$

it is easy to generate realizations that have exactly the desired magnitude: simply compute

$$\exp\left(\sum_{k=1}^{N-1} z_k \sqrt{\lambda_k}\mathbf{v}_k\right), \tag{2.4.8}$$

multiply the result by the desired taper, and then rescale by a multiplicative factor so that the area-weighted sum of the slips gives the total slip required for the desired seismic moment.

2.4.3 Cascadia Subduction Zone (CSZ)

We present an example in which the slip is allowed to vary in both directions along a fault surface. For illustration we use a subset of the Cascadia Subduction Zone from Figure 2.8, taking only the southern-most 8 fault segments, as illustrated in Figure 2.9. These are subdivided into 540 smaller fault planes for the purposes of defining the slip.

To define the 540×540 correlation matrix, we need to compute the pairwise "distance" between subfault i and subfault j. We can compute the Euclidean distance d_{ij}, but for this fault geometry we expect a longer correlation length in the strike direction than down-dip, so we wish to define

$$C_{ij} = \exp(-(d_{strike}(i,j)/r_{strike}) - (d_{dip}(i,j)/r_{dip})) \tag{2.4.9}$$

where $d_{strike}(i,j)$ and $d_{dip}(i,j)$ are estimates of the distance between subfaults i and j in the strike and dip direction, respectively, and r_{strike}, r_{dip} are the correlation lengths in each direction. We define $d_{dip}(i,j)$ using the difference in depth between the two subfaults and the dip angle δ as $d_{dip}(i,j) = d_{depth}/\sin(\delta)$, setting $d_{strike}(i,j) = \sqrt{d_{ij}^2 - d_{dip}(i,j)^2}$. We take the correlation lengths to be 40% of the fault length and width respectively, $r_{strike} = 130$km and $r_{dip} = 40$km. We again use an exponential autocorrelation function as defined in (2.4.9), but this could easily be replaced by a different ACF. We use the lognormal approach described in Section 2.4.2, with parameter $\alpha = 0.5$. Figure 2.9 shows the first 8 eigenmodes of $\hat{\mathbf{C}}^g$.

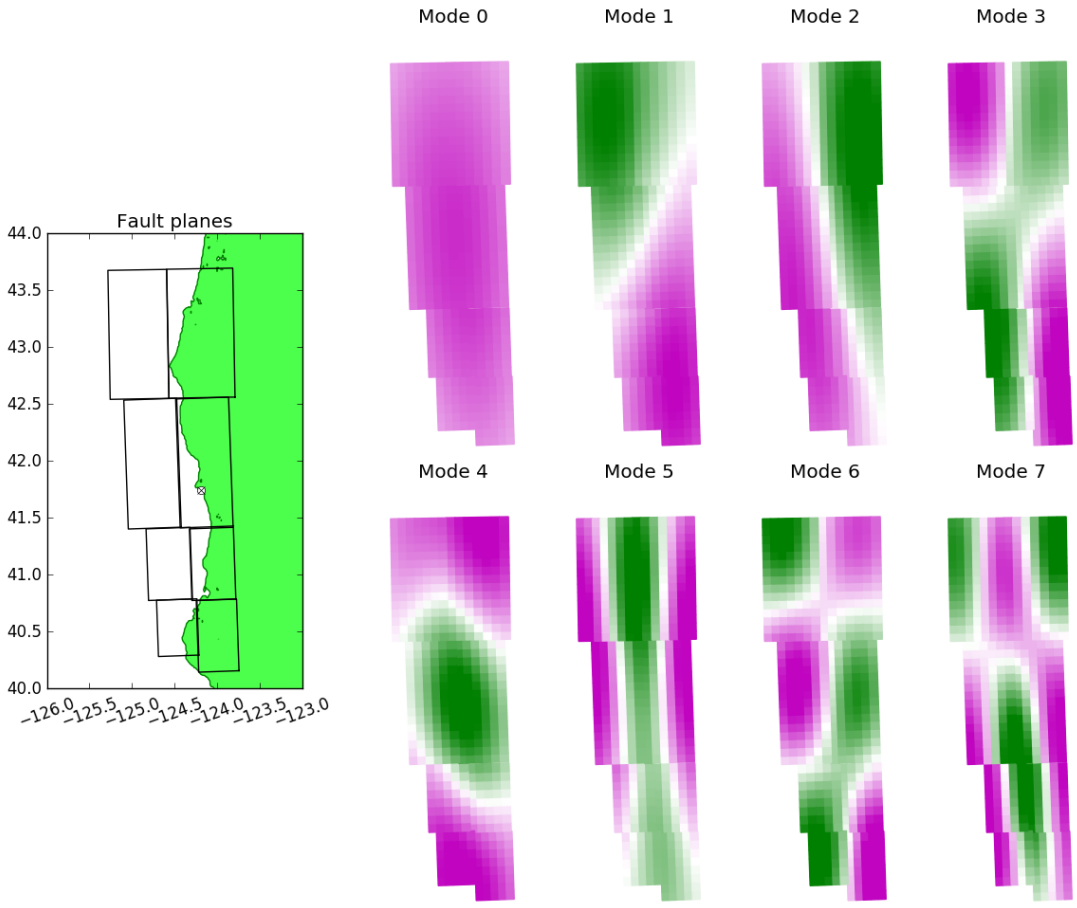

Figure 2.9: Southern portion of the CSZ fault showing location of Crescent City, CA and the 8 subfaults that are further subdivided into 540 subfaults. The first 7 eigenmodes of the resulting covariance matrix $\hat{\mathbf{C}}^g$ are also shown.

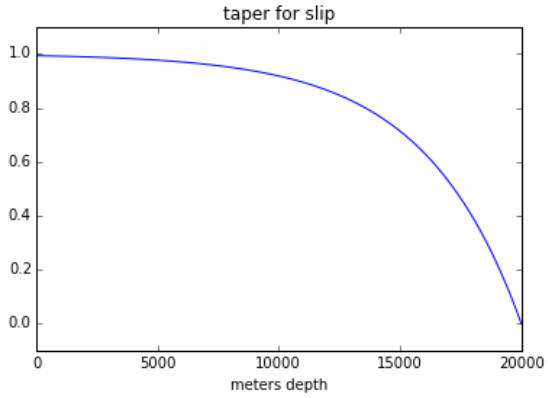

Figure 2.10: The taper (2.4.10) applied to the slip distribution.

Again we drop Mode 0 from the sum, since this mode is roughly constant over the fault.

To create slip realizations, we use (2.5.1) and then apply a tapering only at the down-dip edge, given by

$$\tau(d) = 1 - \exp(-20(d - d_{max})/d_{max}) \tag{2.4.10}$$

with $d_{max} = 20$ km. This taper function is plotted in Figure 2.10. We then scale the slip so that the resulting seismic moment gives $M_w = 8.8$. Figure 2.11 shows 5 typical realizations, comparing the slip generated by a 60-term K-L expansion with the slip generated when the series is truncated after 7 terms. The resulting seafloor deformation in each case is also shown, along with the potential energy and the subsidence/uplift ΔB_{shore} at one point on the coast, the location of Crescent City, CA. Note that in each case the 7-term series gives a smoother version of the slip obtained with 60 terms, and the seafloor deformations are more similar than the slip patterns, as expected from the fact that the Okada model yields a solution to a PDE that is smoothing [27, 82].

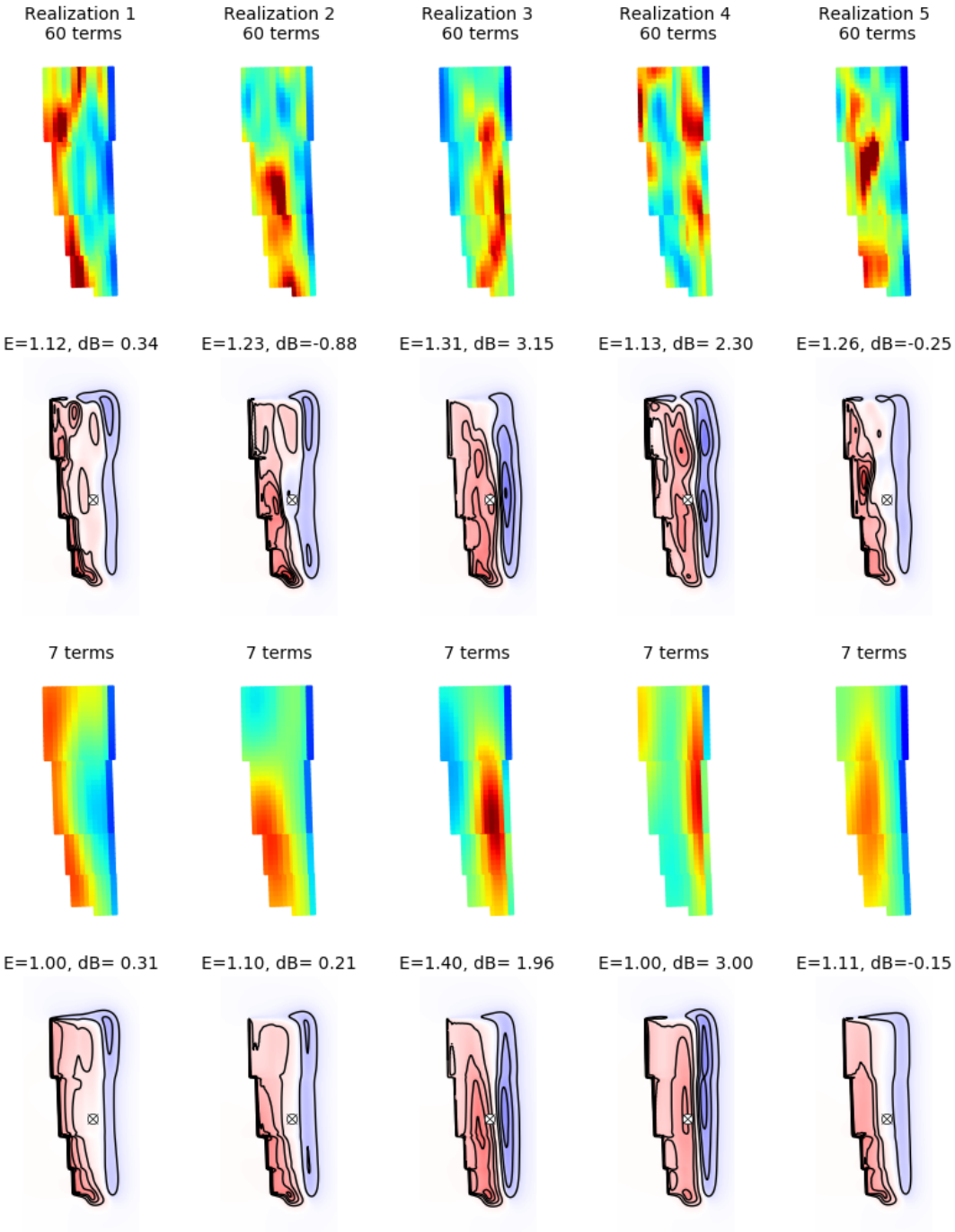

Figure 2.11: The top row shows 5 sample realizations of slip on the southern CSZ fault, as computed with a 60-term K-L expansion. The second row shows the resulting seafloor deformation, with an indication of the potential energy and the vertical displacement at Crescent City, CA, which is indicated by the X in the figures. The third row shows the same 5 realizations but with the K-L series truncated to 7 terms, and the bottom row shows the resulting seafloor deformations.

2.4.4 Discussion

We have presented an approach to defining a probability distribution for earthquake slip patterns on a specified fault geometry that has been subdivided into an arbitrary number of rectangular subfaults, with a specified mean and covariance matrix. Slip realizations can be generated that have either have a joint normal distribution or a joint lognormal distribution. Once the parameters have been chosen that define the distribution, it is very easy to generate an arbitrary number of sample realizations from the distribution, simply by drawing the coefficients z_k of the K-L series from independent normal distributions.

We have also illustrated that with a realistic choice of correlation length, the K-L series can be truncated to a relatively small number of terms. For tsunami modeling applications, the Okada model is applied to each slip pattern to generate the seafloor deformation and it was shown that this is a smoothing operation that can further reduce the number of terms needed, and hence the dimension of the stochastic space that must be explored in doing PTHA analysis.

To use this approach for practical PTHA analysis, two major challenges must be addressed. The first is to tackle the *epistemic uncertainty* associated with the lack of knowledge about possible future earthquakes. We would like to choose the parameters defining the probability distribution in a suitable way for real fault geometries so that it accurately represents the space of possible future earthquakes. This can be very challenging due to the lack of knowledge and sparsity of data from past events, but the existing expertise and geophysical constraints should at least be incorporated in the choice of these parameters. The ability to generate many realizations and examine statistics of quantities such as those used in this work may help in this. As one example, the parameters chosen here for the CSZ example tend to give uplift rather than subsidence at Crescent City [66]. If this is viewed as inconsistent with the geological evidence from past events, this could be adjusted, for example by tapering the slip more on the down-dip side. Moving more of the slip up-dip will cause more subsidence at the shore. It would also be possible to explore ways in which the

epistemic uncertainty associated with the lack of knowledge about the true probability distribution affect the resulting hazard maps generated by a PTHA analysis, for example by doing the analysis with different parameter choices, and hence different probability distributions, to see how robust the PTHA analysis is to changes in assumptions.

The second major challenge is to deal with the *aleatoric uncertainty* that is still present even if the parameters defining the probability distribution were known to be correct. We are still faced with a high-dimensional space to sample in order to perform PTHA analysis. For example, if we wish to compute a hazard map similar to that shown in Figure 2.7 for the probability that the maximum flow-depth ζ (2.3.2) at some particular point will exceed various depths, then for each exceedence value ζ_e we need to calculate (2.3.3). To compute this value, we can use a simple Monte-Carlo method in which the the set of scenarios $\mathcal{S}$ in (2.3.5) are each of equal probability, (2.3.3) is approximated by

$$P[\zeta > \zeta_e] \approx \frac{1}{N_{\text{mc}}} \sum_{j=1}^{N_{\text{mc}}} I[\zeta(\mathbf{z}^{[j]}); \zeta_e]. \tag{2.4.11}$$

with N_{mc} samples and $\mathbf{z}^{[j]}$ now represents the jth sample, drawn from the joint normal distribution with density $\rho(\mathbf{z})$. Note that in writing $\zeta(\mathbf{z})$ we have implicitly transformed the variables $\mathbf{s}$ to $\mathbf{z}$ by the K-L expansion This is feasible with a cheap-to-compute surrogate or proxy, but would not be possible if a full tsunami model is used to compute ζ, which may take hours of computing time for each sample. An alternative would be to compute the integral with a quadrature algorithm based on sampling on a grid in $\mathbf{z}$-space, but this is infeasible for high dimensions m. For example, if $m = 10$ then a tensor-product grid with only 4 points in each direction has $4^{10} \approx 10^6$ points.

Many other techniques have been developed in recent years to estimate such integrals in high dimensional spaces, including for example Latin hypercube sampling (e.g., [83])), sparse grids (e.g., [81]), and quasi-random grids (e.g., [31]) that have many fewer points than uniform tensor-product grids. There are also many Monte-Carlo sampling methods that can obtain accurate results with many fewer samples than the naive sum of (2.4.11),

including multi-level or multi-fidelity methods (e.g., [26, 41, 87]) that combine results from many simulations that are cheap to compute with a relatively few simulations with the full model on a fine grid. Cheaper approximations might be obtained by using some of the proxy quantities above, by computing with a full tsunami model but on coarse grids, or by developing surrogate models or statistical emulators based on relatively few samples (e.g. [9, 10, 67, 100]).

2.5 Multi-resolution strategy

In this section we discuss a multi-resolution strategy. In this strategy coarse-grid runs are performed to obtain $\eta_{\mathbf{s}}$ for most events in the set of earthquake scenarios $(\mathbf{s}, p) \in \mathcal{S}$ in (2.3.7), rather than running fine-grid runs for all scenarios. This results in a significant saving of computational cost. Then, using these coarse-grid runs and a few fine-grid runs, we construct an approximate hazard map.

This multi-resolution strategy is a special case of a more general methodology called *source filtering*, where one selects among a given set earthquake sources $\mathcal{S}$ a smaller representative subset that can be used effectively for approximating the resulting hazard map. For instance, sources can be filtered through clustering techniques on inexpensive proxy variables, or on the seafloor deformations they cause through the Okada model. Moreover, other approximation strategies using post-processing of coarse-grid runs can improve the accuracy of the approximations. This section is based on a study supported by Federal Emergency Management Agency (FEMA) Region IX performed jointly with L. M. Adams, R. J. LeVeque and F. I. Gonzaléz [3], where such additional source filtering methodologies have been explored.

This section is organized as follows. First, a technique for generating extreme earthquake scenarios $\mathcal{S}$ to be used for our numerical experiments are discussed in Section 2.5.1, and the multi-resolution strategy is discussed in Section 2.5.2.

2.5.1 Generation of extreme earthquake scenarios

We use a Karhunen-Loève (K-L) expansion to generate a random slip pattern on a specified fault geometry, sampled from random fields with a specified covariance structure. The general approach we use is described in more detail in Section 2.4. For the fault geometry, we used a subset of the Cascadia Subduction Zone, taking only the southern-most 8 fault segments as illustrated above in Figure 2.9. The same geometry was used in Section 2.4 and is discussed further there. The slip $\mathbf{s} \in \mathbb{R}^{540}$ on each subfault was assumed to follow a joint lognormal

distribution. To define the 540×540 correlation matrix, we first computed the pairwise distance between subfault i and subfault j. The correlation C_{ij} is computed as in (2.4.9), again with correlation lengths 40% of the fault length and width, respectively.

Since the lognormal variables are not linear, we perform the K-L expansion on the transformed Gaussian variables, $\log \mathbf{s}$, with corresponding correlation C^g_{ij}. The slip $\mathbf{s}$ is now written as,

$$\mathbf{s} = \exp\left[\boldsymbol{\mu} + \sum_{k=1}^{539} z_k \sqrt{\lambda_k}\mathbf{v}_k\right] \quad \text{where } z_k \sim \mathcal{N}(0,1), \tag{2.5.1}$$

and where λ_j and $\mathbf{v}_j$ are eigenpairs of the covariance matrix for the Gaussian variables. In this work we will truncate the K-L expansion after 20-terms, that is, we use the approximation

$$\mathbf{s} \approx \exp\left[\boldsymbol{\mu} + \sum_{k=1}^{20} z_k \sqrt{\lambda_k}\mathbf{v}_k\right] \quad \text{where } z_k \sim \mathcal{N}(0,1). \tag{2.5.2}$$

In Section 2.4.3, we illustrated that 7-term expansion is sufficient, so the 20-term expansion used here will generate events whose seafloor deformations will accurately represent the uncertainty in slip for PTHA. As before, a slip realization is generated by drawing samples from the normal distribution $\mathcal{N}(0,1)$, then computing $\mathbf{s}$ in the series above. Once a slip is generated from the series, we apply a tapering $\tau(d)$ at the down-dip edge, given by setting $d_{max} = 20$ km in (2.4.10). The plot of this taper is given in Figure 2.10. We then scale the slip to obtain the desired seismic moment, for example $M_w = 8.6$.

We generated 100 events of magnitude $M_w = 8.6$ using the K-L expansion with a correlation length of 130 km in the strike direction and 40 km in the dip direction. In order to generate extreme events, these were sampled by first choosing the z_k coefficients each from a Gaussian with standard deviation 1. We draw 20,000 quasi Monte Carlo samples from a Halton sequence, to ensure that the samples are spread out and cover the probability space well. These samples are shown in Figure 2.12. Then we dilate them by multiplying by a factor (here 4σ), with the result that the majority of the events are far out in the tail of the distribution. After exponentiating the Gaussian K-L expansion, this dilation appears in the

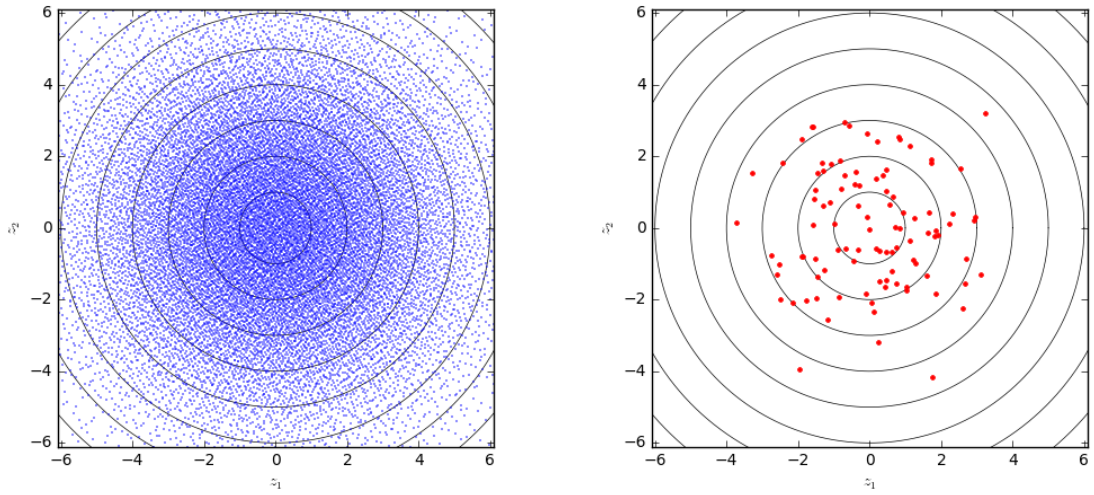

Figure 2.12: The 20,000 quasi Monte Carlo samples, projected on to z_1-z_2 plane (left), and 100 cluster centers obtained through k-means clustering (right).

expression (2.5.1) as

$$\mathbf{s}/\exp(\boldsymbol{\mu}) \sim \exp\left[\sum_{k=1}^{20}(4\sigma)z_k\sqrt{\lambda_k}\mathbf{v}_k\right] = \left(\exp\left[\sum_{k=1}^{20} z_k\sqrt{\lambda_k}\mathbf{v}_k\right]\right)^4. \tag{2.5.3}$$

Taking the power of 4 of the slips above has the effect of generating events that tend to have highly concentrated regions of high slip. That is, the resulting tsunami varies dramatically depending on where the high slip region is located, in part because there can be large values of uplift or subsidence along the coast.

Each of these sample points correspond to a potential earthquake scenario, therefore we have so far obtained a catalogue of 20,000 potential earthquake scenarios (corresponding annual probabilities have not yet been assigned.) Since we are constrained by the computational budget, we must reduce the number of scenarios to $N = 100$. To this end, we apply the k-means clustering to the large number of scenarios, and obtain 100 clusters C_k. There

are various clustering algorithms that could accomplish this, for example, the well-known Lloyd's algorithm [68]. The cluster centers will each represent an earthquake scenario which will be used for the tsunami simulation. An example of cluster centers is shown in Figure 2.12.

The remaining task is to assign the annual probability to each of the earthquake scenarios. Relative probabilities were assigned to these 100 events by assuming that each event is representative of some region of parameter space (a high-dimensional space whose dimension is equal to the number of terms used in the K-L expansion, 20 for this work). A Voronoi tessellation of this 20-dimensional space was constructed in performing the k-means clustering above, subdividing the space into disjoint regions R_k, each corresponding to the cluster C_k. This region was formed for each event determined by coefficients $\mathbf{z}^{[j]} \in \mathbb{R}^{20}$ and consists of all points $\mathbf{z}$ that are closer to $\mathbf{z}^{[k]}$ than to any other $\mathbf{z}^{[j]}$. Then $N_{\text{mc}} = 1,000,000$ Monte Carlo samples of the $\mathbf{z}$ values were generated from the joint normal distribution and the fraction of these points falling in each region R_k was used for the probability p_k. These samples will be indexed by $\mathbf{z}_\ell$. Let us denote the indicator function that returns 1 if a sample $\mathbf{z}$ lies in the region by $I_k : \mathbb{R}^{20} \to \{0, 1\}$ for sample $\mathbf{z}$ by $I_k(\mathbf{z})$ and the k-th cluster by R_k,

$$I_k(\mathbf{z}) = \begin{cases} 1 & \text{if } \mathbf{z} \in R_k, \\ 0 & \text{otherwise.} \end{cases} \tag{2.5.4}$$

Then the probability weights are computed by

$$p_k = \frac{1}{N_{\text{mc}}} \sum_{\ell=1}^{N_{\text{mc}}} I_k(\mathbf{z}_\ell) \tag{2.5.5}$$

Since the higher-order terms in the K-L expansion do not significantly affect the tsunami resulting from the earthquake, we computed the probability weights based on the 7-dimensional $\mathcal{N}(0, I)$, for computational efficiency.

From this set of 100 realizations we created 300 more by scaling them up to a larger

magnitude by multiplying the slip on each subfault (and hence the seafloor deformation) by approximately a factor of 2, which increases the moment magnitude by 0.2. In this manner we generated 100 realizations at each magnitude M_w 8.6, 8.8, 9.0, and 9.2.

The probabilities computed for each of he original 100 realizations using the Voronoi diagram approach described above was then apportioned between the 4 different magnitude events using weights of $(0.3,\ 0.3,\ 0.3,\ 0.1)$, with equal weighting given to the smaller three magnitudes and less weight to the extreme M_w 9.2 events.

These realizations are not meant to be a realistic representation of actual observable events associated with the CSZ. In particular, this approach gives realizations with highly concentrated slip patches where the slip exceeded 180 m for some of the M_w 9.2 events which is not realistic, although it generates an interesting test data corresponding to rare and extreme events. Additionally, further work is required to verify that the probability assigned here using Voronoi diagrams yield a good approximation to an optimal estimate, which is beyond the scope of this thesis.

2.5.2 Multi-resolution strategy using SVD

In this section we outline a strategy that exploits the high similarity between the singular modes of the fine-grid runs and those of the coarse-grid runs. The strategy is inspired by existing multi-level or multi-fidelity strategies; see recent works [34, 79], as well as [41, 87]. This strategy is carried out separately for each of the four magnitudes represented in the study, recalling there are 100 fine-grid runs and 100 coarse-grid runs for each magnitude.

Let us denote by B_0 the bathymetry before any deformation occurs, that is, before the Okada model is applied, and by $h_{\max}$ the maximum of water height h over all time.

For each of the rectangular fgmax data, let us flatten the 88×78 grid data of $h_{\max} + B_0$ values into column vectors, then stack them into a 6864×100 matrix. Let us denote by A_f such a matrix formed for the 100 fine-grid runs and by A_c a corresponding matrix for the

100 coarse-grid runs. We compute the SVD of these two matrices,

$$A_c = U_c \Sigma_c V_c^T = U_c Y_c \quad \text{where } Y_c = \Sigma_c V_c^T, \tag{2.5.6}$$

$$A_f = U_f \Sigma_f V_f^T = U_f Y_f \quad \text{where } Y_f = \Sigma_f V_f^T. \tag{2.5.7}$$

This strategy relies on the fact that first columns of U_c and those of U_f are similar, as well as the first rows of Y_c and those of Y_f. The similarity between U_c and U_f, for example, are shown in Figure 2.13.

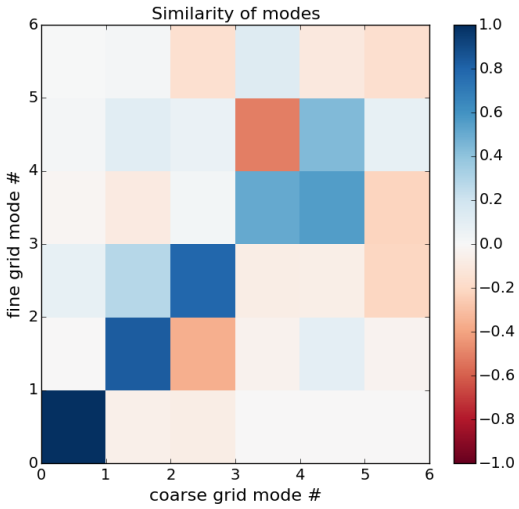

Figure 2.13: High similarity between coarse and fine singular modes, for the first 6 modes taken from magnitude 9.0 events. Similarity is measured by the cosine of the angle between the two unit vectors.

The values on the diagonal of $U_c^T U_f$ have absolute values close to 1, indicating a strong similarity. The similarity begins to break down after the 4th to 5th mode, so we use the first 4 modes to construct an approximation to A_f.

In our study, computing A_f would require conducting all the fine-grid runs. Therefore, in order to compute an approximation to A_f, which we denote by $\hat{A}_f$, we attempt to construct

an approximation to the first four columns of U_f and the first four rows of Y_f using minimal amount of information from the fine-grid runs. Since the coarse and fine runs are similar for the first four modes, we can conclude that the coarse runs that have large components (values in the first 4 rows of Y_c) in the first four left singular directions can be used to build an approximation to the first 4 columns of U_f which we call $\hat{U}_f$. Since we are given Y_c, we sort the columns (indexed by i) of its first four rows 4 ways for $j = 1, 2, 3, 4$, largest to smallest by the value

$$\frac{|Y_{c,ji}|}{\left[\sum_{\substack{k\neq j \\ k\in\{1,2,3,4\}}} |Y_{c,ki}|^2\right]^{1/2}}. \tag{2.5.8}$$

In this study, the largest 2 columns for each of $j = 1, 2, 3, 4$ were chosen, so that we arrive at 8 column indices of the coarse grid runs. These indices will be the fine-grid runs we will perform, and we will denote the set of these indices by I.

Taking the fine-grid rectangular fgmax $h_{\max} + B_0$ values from the runs indexed by I, we again flatten the grid data and form a 6864×8 matrix, which we denote by B_f. The SVD of B_f is

$$B_f = \bar{U}_f \bar{\Sigma}_f \bar{V}_f^T = \bar{U}_f \bar{Y}_f \quad \text{where } \bar{Y}_f = \bar{\Sigma}_f \bar{V}_f^T. \tag{2.5.9}$$

We can now use the first four columns of $\bar{U}_f$ as our value for $\hat{U}_f$.

Now, we turn to the approximation of the first four rows of Y_f. While the first four rows of Y_c and Y_f are similar, we can not simply equate them. We could use the known information in the first four rows of $\bar{Y}_f$ (called $\tilde{Y}_f$) to find a linear map T that can transform the first four rows of Y_c (called $\bar{Y}_c$) into its fine-grid counterpart ($\tilde{Y}_f$) in the least-squares sense, by solving the minimization problem

$$\text{minimize} \|T\bar{Y}_{c,i} - \tilde{Y}_{f,i}\|^2 \quad \text{for } i \in I. \tag{2.5.10}$$

The components of the computed T shown in Figure 2.14 reveal that it is roughly a diagonal map, indicating that there is also a strong similarity between the columns of $\bar{Y}_c$ and $\tilde{Y}_f$ represented in I.

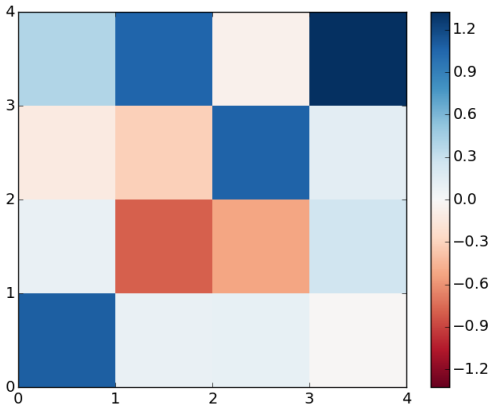

Figure 2.14: Components of T in the minimization problem of (2.5.10).

Finally, we choose $\hat{A}_f = \hat{U}_f T \bar{Y}_c$ to be the approximation to A_f. This approximation is done for each of the magnitudes (8.6, 8.8, 9.0, and 9.2). Since we used 8 fine-grid runs for each magnitude, we have used 32 fine-grid runs total. This results in a significant computational savings, by reducing the number of fine-grid runs required by 92%.

2.5.3 Discussion

While this multi-resolution strategy yields reasonable results, a visual inspection of the inundation patterns suggest that the method does not exploit an intuitively obvious pattern present in the data. Consider the three inundation patterns shown in Figure 2.15. The three patterns follow an obvious dynamic: the inundation patterns closely follow the contour lines of the topography. They resemble a solution to a transport problem with proper influx boundary conditions and variable coefficients. If one views the time-variation as a nonlinear

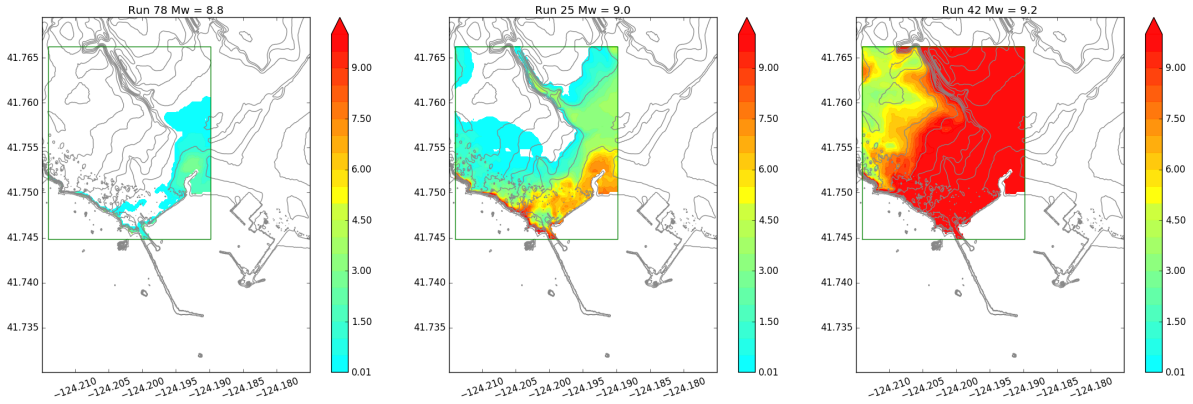

Figure 2.15: High-rank behavior inundation patters visible from set of scenarios $\mathcal{S}$. Run number 78 with $M_w = 8.8$ (left), Run number 25 with $M_w = 9.0$ (middle), and Run number 42 with $M_w = 9.2$ (right).

parameter, the three flooding patterns in Figure 2.15 look like a solution to a transport problem shown at different times.

Therefore, the low-dimensional behavior of inundation patterns can be better represented in terms of the level-sets of the topography, and transport between these level-sets. In contrast, the three responses are clearly linearly independent, which can be easily deduced from the fact that the support of the reponses are substantially different. This observation can be summarized by the statement that the responses ζ exhibit *high-rank* behavior with respect to its argument $\mathbf{s}$. Naturally, this high-rank behavior here cannot be captured well by the SVD, since such a behavior cannot be represented by a small number of basis functions. This is commonly observed from responses from hyperbolic problems (see Section 3.1 for a simple 1D example). Therefore, this method can potentially improved if one finds a different tool than the SVD or other linear decompositions that takes into account this high-rank behavior.

Chapter 3

The Transport Reversal Algorithm

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 [14, 24, 33, 52, 53, 91, 92, 102, 112] have been successfully applied to various partial differential equations (PDEs), including those arising in fluid dynamics [15, 30, 58, 85, 93, 115]. However, these projection-based methods can be ineffective when applied to 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 [21] where a fail-safe h-adaptive algorithm was introduced.

In this chapter, we introduce the transport reversal algorithm. It is an iterative algorithm which extracts traveling structures from solutions to hyperbolic PDEs in a greedy manner. This chapter is based on a joint preprint with S. Moe and R.J. LeVeque [96].

3.1 Motivating example

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

$$u_t + cu_x = 0 \qquad \text{in } \Omega, \tag{3.1.1}$$

along with the periodic boundary condition and the initial condition

$$u(0,t) = u(1,t) \quad \text{for } t \in [0,T], \qquad u(x,0) = u_0(x) \equiv \delta(x). \tag{3.1.2}$$

We assume $c = 1$ here. Let us seek a solution using the finite volume method (FVM) with upwind flux [64]. We set the grid points $x_j = jh$ for $j = 0, 1, ..., N$ 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

$$u_j^n \approx \frac{1}{h} \int_{x_{j-1/2}}^{x_{j+1/2}} u(x, t_n)\, \mathrm{d}x, \tag{3.1.3}$$

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,

$$\mathbf{u}^n = \mathbf{K}^n \mathbf{u}^0, \quad \text{where} \quad \mathbf{K} \equiv \begin{bmatrix} 0 & 0 & \cdots & 0 & 1 \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix} \quad \text{and} \quad \mathbf{u}^0 = \frac{1}{h}\mathbf{e}_1. \tag{3.1.4}$$

This solution has no error apart from the discretization of the initial data, and thereafter reproduces the cell averages (3.1.3) exactly.

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

$$\mathbf{A} \equiv \begin{bmatrix} \mathbf{u}^0 & \mathbf{u}^1 & \cdots & \mathbf{u}^{N-1} \end{bmatrix} = \frac{1}{h}\mathbf{I}, \quad \text{where } \mathbf{I} \text{ is the identity in } \mathbb{R}^{N \times N}. \tag{3.1.5}$$

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

A after some R terms. Usually, we choose the smallest R such that, for a given tolerance $\varepsilon \ll 1$, the remainder satisfies

$$\sum_{j=R+1}^{N} \sigma_j^2 \Bigg/ \sum_{j=1}^{N} \sigma_j^2 < \varepsilon, \quad \text{in this case } 1 - R/N < \varepsilon. \tag{3.1.6}$$

The LHS decreases linearly in R, so R must be large even for a moderately small ε to satisfy (3.1.6). 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 **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 [97, 98] 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* [57] 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

$$\min_a \int_0^1 |u(x-a,t) - u_0(x)|^2 \, \mathrm{d}x. \tag{3.1.7}$$

This minimization resembles the orthogonal Procrustes problem [46], 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 $u(t)$ on M, the 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 $u(t)$ [98]. 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 [98] 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 (3.1.7), 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 3.2.

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 3.5. Then an extension to the case where the speed c in (3.1.1) varies with respect to the spatial variable is introduced in Section 3.8. Nothing prevents these extensions from being used in conjunction with the iterative transport reversal algorithm introduced in the preceding section. In Section 3.9, we present some geometric interpretations.

Transport reversal shares features with the shifted proper orthogonal decomposition (sPOD) introduced in [95]. It can be related to the dynamic mode decomposition (DMD) [99, 109] in the sense that the periodic shift operator in (3.1.4) 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 towards 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 [4, 5, 6, 70, 86]. The algorithms in this chapter allows one to apply these methods in conjunction with *displacement interpolation* (see, e.g., [113]) 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 (gPC) [89].

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 [51]. In exploiting this remarkable property, one obtains a multidimensional extension of the large time-step method [60, 61, 62], and therefore the multidimensional analogue of the transport reversal algorithm. The scope of this chapter does not permit a detailed account of this important extension. A thorough treatment will appear elsewhere, based on the one-dimensional algorithm presented in this chapter.

3.2 Transport reversal

In this section, we discretize and generalize the problem (3.1.7) and then introduce the transport reversal algorithm. To motivate the discussion, let us revisit the problem (3.1.1). Recall the finite volume solution $\mathbf{u}^n$, the matrix of shifts with periodic boundary conditions $\mathbf{K}$ (3.1.4), and the snapshot matrix $\mathbf{A}$ (3.1.5). 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 $\mathring{\mathbf{A}}$,

$$\mathring{\mathbf{A}} \equiv \begin{bmatrix} \mathbf{I} & \mathbf{K}^{-1} & \mathbf{K}^{-2} & \cdots & \mathbf{K}^{-(N-1)} \end{bmatrix} \odot \begin{bmatrix} \mathbf{u}^0 & \mathbf{u}^1 & \mathbf{u}^2 & \cdots & \mathbf{u}^{N-1} \end{bmatrix} \tag{3.2.1}$$

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

$$\mathring{\mathbf{A}} = \begin{bmatrix} \mathbf{u}^0 & \cdots & \mathbf{u}^0 \end{bmatrix} = \begin{bmatrix} \frac{1}{h} & \frac{1}{h} & \cdots & \frac{1}{h} \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix} = \begin{bmatrix} N & N & \cdots & N \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}, \tag{3.2.2}$$

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

$$\mathring{\mathbf{A}} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^* = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \begin{bmatrix} N\sqrt{N} \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{N}} & \frac{1}{\sqrt{N}} & \cdots & \frac{1}{\sqrt{N}} \end{bmatrix}. \tag{3.2.3}$$

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 (3.1.7), and we formulate its generalization as follows.

Definition 3.2.1 (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 $\mathbf{A}$, then define the integers $\nu_j \in \mathbb{Z}_N$ to be the minimizers

$$\nu_j = \text{argmin}_{\omega \in \mathbb{Z}_N} \|\mathbf{a}_j - \mathbf{K}^\omega \mathbf{b}\|_2^2 \quad \text{for } j = 1, 2, ..., M. \tag{3.2.4}$$

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 (3.2.4) as

$$\boldsymbol{\nu} = \mathcal{C}(\mathbf{A}; \mathbf{b}). \tag{3.2.5}$$

In (3.2.4) we are merely shifting the entries of the pivot $\mathbf{b}$ to match $\mathbf{a}_j$ as much as possible. Here we introduce some more notations regarding the computation $\mathcal{C}$.

Notation 3.2.2. 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 (3.2.4) 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 $\ell : \{1, ..., M\} \to \{1, ..., M\}$, 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 columns 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 3.2.3 (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

$$\mathcal{T}(\mathbf{A};\boldsymbol{\nu}) \equiv \begin{bmatrix} \mathbf{K}^{\nu_1} & \mathbf{K}^{\nu_2} & \cdots & \mathbf{K}^{\nu_M} \end{bmatrix} \odot \begin{bmatrix} \mathbf{a}_1 & \mathbf{a}_2 & \cdots & \mathbf{a}_M \end{bmatrix}. \tag{3.2.6}$$

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

$$\mathcal{T}(\mathbf{b};\boldsymbol{\nu}) \equiv \begin{bmatrix} \mathbf{K}^{\nu_1}\mathbf{b} & \mathbf{K}^{\nu_2}\mathbf{b} & \cdots & \mathbf{K}^{\nu_M}\mathbf{b} \end{bmatrix}. \tag{3.2.7}$$

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 $\boldsymbol{\nu}$,

$$\mathcal{T}(\mathcal{T}(\mathbf{A};-\boldsymbol{\nu});\boldsymbol{\nu}) = \mathbf{A} \quad \text{ for } \mathbf{A} \in \mathbb{R}^{N\times M}. \tag{3.2.8}$$

The key observation in the example above (3.2.3) 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 [95]. The effectiveness of this approach, along with extensions will be discussed further in Section 3.4.

Unfortunately, template fitting has several important drawbacks. We will demonstrate them through typical examples of hyperbolic PDEs. Suppose the given matrix $\mathbf{A}$ is a snapshot matrix from the following four hyperbolic problems. Diagrams visually illustrating the solution behavior are shown in Figure 3.1, with respective enumeration.

(P1) Advection equation with source term,

$$u_t + u_x = -\gamma u \quad \text{ in } (0,1), \quad \text{ with } \gamma > 0, \tag{3.2.9}$$

where $u(x,0)$ is a non-negative 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 (3.2.4).

(P2) Advection equation

$$u_t + u_x = 0 \quad \text{in } (0,1), \tag{3.2.10}$$

with absorbing boundary conditions, $u_x = -u_t$ at the right boundary $x = 1$, and $u(x,0)$ a density pulse. In (3.2.4) periodic shift $\mathbf{K}$ assumes periodic boundary conditions, so there is little hope of capturing this absorption.

(P3) Acoustic equations in a homogeneous medium,

$$\begin{bmatrix} p \\ u \end{bmatrix}_t + \begin{bmatrix} 0 & K_0 \\ 1/\rho_0 & 0 \end{bmatrix} \begin{bmatrix} p \\ u \end{bmatrix}_x = 0 \quad \text{in } (0,1) \tag{3.2.11}$$

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 (3.2.4) cannot be used to represent the two different speeds.

(P4) Burgers' equations

$$u_t + uu_x = 0 \quad \text{in } (0,1), \tag{3.2.12}$$

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

In this section, we address these issues by the generalization of the operators $\mathcal{C}$ and $\mathcal{T}$, in which we add new features to template fitting procedure (3.2.4). Each of these features are introduced one by one in Sections 3.2.1, 3.2.2, 3.2.3 and 3.2.4. Algorithm 3.2.4 describes the final iterative algorithm.

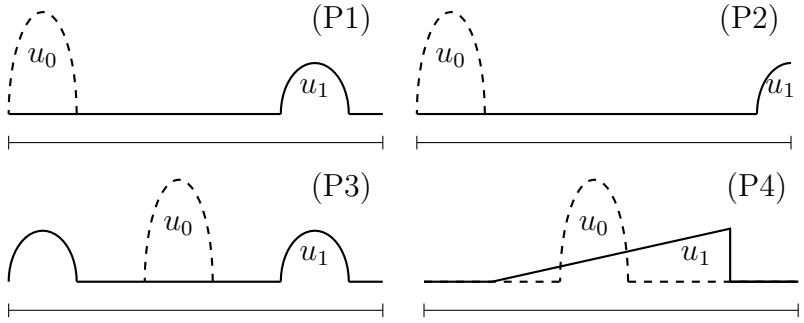

Figure 3.1: Illustration of solution behavior for the four hyperbolic problems (P1), (P2), (P3) and (P4). u_0 denotes the initial profile, drawn in dashed lines, and u_1 denotes the solution at some future time, in solid lines.

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 3.1 (P1). The minimization problem in (3.2.4) does not take the scaling into account. We introduce a scaling by projecting the j-th column onto the pivot. The projection $\mathcal{P}$ is defined as

$$\mathcal{P}(\mathbf{a}_j;\mathbf{b}) \equiv \begin{cases} \dfrac{\mathbf{b}\mathbf{b}^T}{\|\mathbf{b}\|^2}\mathbf{a}_j & \text{if } \|\mathbf{b}\| > 0, \\ \mathbf{0} & \text{otherwise, if } \|\mathbf{b}\| = 0. \end{cases} \tag{3.2.13}$$

Now, we replace the functional in the minimization problem (3.2.4) by measuring the difference between the j-th 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

$$\nu_j = \text{argmin}_{\omega\in\mathbb{Z}_N} \|\mathbf{a}_j - \mathcal{P}(\mathbf{a}_j;\mathbf{K}^\omega\mathbf{b})\|_2^2 . \tag{3.2.14}$$

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

$$\boldsymbol{\nu} = \mathcal{C}(\mathbf{A};\mathbf{b},\mathcal{P}), \tag{3.2.15}$$

by supplying the projection map $\mathcal{P}$. The scaling (3.2.13) must also be stored, and we organize it in the vector $\mathbf{h}$,

$$h_j = \mathcal{P}(\mathbf{a}_j; \mathbf{K}^{\nu_j}\mathbf{b}) \quad \text{and} \quad \mathbf{h} = \begin{bmatrix} h_1 & \cdots & h_M \end{bmatrix}. \tag{3.2.16}$$

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 $\mathbf{h}$,

$$\mathcal{T}(\mathbf{A}; \boldsymbol{\nu}, \mathbf{h}) \equiv \begin{bmatrix} h_1\mathbf{K}^{\nu_1} & \cdots & h_M\mathbf{K}^{\nu_M} \end{bmatrix} \odot \begin{bmatrix} \mathbf{a}_1 & \cdots & \mathbf{a}_M \end{bmatrix}. \tag{3.2.17}$$

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

$$\mathcal{T}(\mathbf{b}; \boldsymbol{\nu}, \mathbf{h}) \equiv \begin{bmatrix} h_1\mathbf{K}^{\nu_1}\mathbf{b} & h_2\mathbf{K}^{\nu_2}\mathbf{b} & \cdots & h_M\mathbf{K}^{\nu_M}\mathbf{b} \end{bmatrix}. \tag{3.2.18}$$

3.2.2 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 (3.2.14), 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 (3.2.4), and the existence of the exact inverse in (3.2.8) 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 towards 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 i-th component ρ_i of $\boldsymbol{\rho}$ is defined as

$$\rho_i = (\mathcal{S}(\mathbf{a}_j; \mathbf{b}))_i \equiv \begin{cases} 1 & \text{if } \operatorname{sign}(a_{ij} - b_i) \cdot \operatorname{sign}(a_{ij}) \geq 0 \\ & \qquad \text{and } |a_{ij} - b_i| \leq |a_{ij}|\,, \\ 0 & \text{otherwise.} \end{cases} \tag{3.2.19}$$

The intention is to use $\rho_i b_i$ to approximate a_i. The first condition $\operatorname{sign}(a_{ij} - b_i) \cdot \operatorname{sign}(a_{ij}) \geq 0$ 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 (3.2.19) in place of $\mathbf{b}$. To simplify the notation, we will use the shorthand

$$\mathcal{S}(\mathbf{a}_j; \mathbf{b}, \mathcal{P}) \equiv \mathcal{S}(\mathbf{a}_j; \mathcal{P}(\mathbf{a}_j; \mathbf{b})). \tag{3.2.20}$$

Now the minimization (3.2.14) is further updated: we shift $\mathbf{b}$ and scale it using the projection $\mathcal{P}$, and we cut-off using $\mathcal{S}$, then we compare with $\mathbf{a}_j$. The new minimization problem becomes,

$$\nu_j = \operatorname{argmin}_{\omega \in \mathbb{Z}_N} \left\| \mathbf{a}_j - \mathcal{S}(\mathbf{a}_j; \mathbf{K}^\omega \mathbf{b}, \mathcal{P}) \odot \mathcal{P}(\mathbf{a}_j; \mathbf{K}^\omega \mathbf{b}) \right\|_2^2 . \tag{3.2.21}$$

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

$$\boldsymbol{\nu} = \mathcal{C}(\mathbf{A}; \mathbf{b}, \mathcal{P}, \mathcal{S}). \tag{3.2.22}$$

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

$$\boldsymbol{\rho}_j = \mathcal{S}(\mathbf{a}_j; \mathbf{K}^{\nu_j} \mathbf{b}, \mathcal{P}). \tag{3.2.23}$$

We store these as columns of the matrix

$$\mathbf{P} = \begin{bmatrix} \boldsymbol{\rho}_1 & \cdots & \boldsymbol{\rho}_M \end{bmatrix}, \tag{3.2.24}$$

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

$$\begin{aligned} \mathcal{T}(\mathbf{A}; \boldsymbol{\nu}, \mathbf{h}, \mathbf{P}) &\equiv \begin{bmatrix} h_1 \boldsymbol{\rho}_1 \odot \mathbf{K}^{\nu_1} & \cdots & h_M \boldsymbol{\rho}_M \odot \mathbf{K}^{\nu_M} \end{bmatrix} \odot \begin{bmatrix} \mathbf{a}_1 & \cdots & \mathbf{a}_M \end{bmatrix} \\ &= \begin{bmatrix} h_1 \boldsymbol{\rho}_1 \odot \mathbf{K}^{\nu_1} \mathbf{a}_1 & \cdots & h_M \boldsymbol{\rho}_M \odot \mathbf{K}^{\nu_M} \mathbf{a}_M \end{bmatrix}. \end{aligned} \tag{3.2.25}$$

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.

3.2.3 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 $\mathbf{P}_1$,

$$\boldsymbol{\nu}_1 \equiv \mathcal{C}(\mathbf{R}_1; \mathbf{b}_1, \mathcal{P}, \mathcal{S}), \quad \mathbf{h}_1 \equiv \mathcal{P}(\mathbf{R}_1; \mathbf{b}_1, \boldsymbol{\nu}_1) \quad \text{and} \quad \mathbf{P}_1 \equiv \mathcal{S}(\mathbf{R}_1; \mathbf{b}_1, \boldsymbol{\nu}, \mathcal{P}). \tag{3.2.26}$$

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

$$\mathbf{R}_2 \equiv \mathbf{R}_1 - \mathcal{T}(\mathbf{b}_1; \boldsymbol{\nu}_1, \mathbf{h}_1, \mathbf{P}_1). \tag{3.2.27}$$

This forms one iteration. We remark that the conditions in (3.2.19) 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 k-th iteration by computing the ratio $\|\mathbf{R}_k\|/\|\mathbf{R}_{k-1}\|$. One may set a threshold τ_1 so that

$$\text{if } \frac{\|\mathbf{R}_k\|_F}{\|\mathbf{R}_{k-1}\|_F} > \tau_1, \text{ then update the pivot } \mathbf{b}_\ell \text{ to } \mathbf{b}_{\ell+1}. \tag{3.2.28}$$

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 ℓ-th 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 $\mathbf{V}$,

$$\mathbf{V} \equiv \begin{bmatrix} \boldsymbol{\nu}_1 & \cdots & \boldsymbol{\nu}_K \end{bmatrix}. \tag{3.2.29}$$

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

collect them in $\mathbf{Q}$,

$$\mathbf{Q} \equiv \begin{bmatrix} \mathbf{P}_1 & \cdots & \mathbf{P}_K \end{bmatrix}, \tag{3.2.30}$$

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

$$\mathbf{H} \equiv \begin{bmatrix} \mathbf{h}_1 & \cdots & \mathbf{h}_K \end{bmatrix}. \tag{3.2.31}$$

3.2.4 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 (3.2.21) when $j > 1$ to try to keep $|\nu_{(j-1)k} - \nu_{jk}|$ small:

$$\nu_{jk} = \operatorname{argmin}_{\omega \in \mathbb{Z}_N} \left\| \mathbf{a}_j - \mathcal{S}(\mathbf{a}_j; \mathbf{K}^{\omega}\mathbf{b}, \mathcal{P}) \odot \mathcal{P}(\mathbf{a}_j; \mathbf{K}^{\omega}\mathbf{b}) \right\|_2^2 + \lambda \left| \omega - \nu_{(j-1)k} \right|^2, \tag{3.2.32}$$

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 (3.2.32) 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., [113]), 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, 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 (3.2.32).

A simplified pseudo-code of the transport reversal is given in Algorithm 3.2.4.

Algorithm 3.2.4 (Transport reversal algorithm)**.**

procedure TR($\mathbf{A}, K, \tau_0, \tau_1$)
▷ input matrix $\mathbf{A}$, max. no. of iterations K, and tolerances τ_0, τ_1
  $\ell \leftarrow 1$ ▷ pivot number
  $k \leftarrow 0$ ▷ iteration count
  $r_{\text{old}} \leftarrow \|\mathbf{A}\|_F$
  $\mathbf{R} \leftarrow \mathbf{A}$ ▷ initialize residual $\mathbf{R}$
  $\mathbf{b}_\ell \leftarrow \mathbf{R}(:, \ell)$ ▷ choose first column of $\mathbf{R}$ as pivot
  while ($r_{\text{old}} > \tau_0$ and $k \leq K$) **do**
    $k \leftarrow k + 1$
    $(\boldsymbol{\nu}_k, \mathbf{h}_k, \mathbf{P}_k) \leftarrow (\mathcal{C}(\mathbf{R}; \mathbf{b}_\ell, \mathcal{P}, \mathcal{S}), \mathcal{P}(\mathbf{R}; \mathbf{b}_\ell, \boldsymbol{\nu}_k), \mathcal{S}(\mathbf{R}; \mathbf{b}_\ell, \boldsymbol{\nu}_k, \mathcal{P}))$
▷ computation is done concurrently
    $\mathbf{R} \leftarrow \mathbf{R} - \mathcal{T}(\mathbf{b}_\ell; \boldsymbol{\nu}_k, \mathbf{h}_k, \mathbf{P}_k)$
    $r_{\text{new}} \leftarrow \|\mathbf{R}\|_F$
    if $r_{\text{new}}/r_{\text{old}} > \tau_1$ **then**
      $\ell \leftarrow \ell + 1$ ▷ pivoting
      $\mathbf{b}_\ell \leftarrow \mathbf{R}(:, \ell)$ ▷ update pivot to be the ℓ-th column of the new $\mathbf{R}$
    end if
    $r_{\text{old}} \leftarrow r_{\text{new}}$
  end while
  $\mathbf{B} \leftarrow [\mathbf{b}_1, \cdots, \mathbf{b}_\ell]$

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**

3.3 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 (3.2.21): we set λ in (3.2.32) as $2.5/(CN)$ where

$$\begin{aligned} C \equiv \max_{\omega \in \mathbb{Z}_N} \|\mathbf{a}_j - \mathcal{S}(\mathbf{a}_j; \mathbf{K}^\omega \mathbf{b}, \mathcal{P}) \odot \mathcal{P}(\mathbf{a}_j; \mathbf{K}^\omega \mathbf{b})\|_2^2 \\ - \min_{\gamma \in \mathbb{Z}_N} \|\mathbf{a}_j - \mathcal{S}(\mathbf{a}_j; \mathbf{K}^\gamma \mathbf{b}, \mathcal{P}) \odot \mathcal{P}(\mathbf{a}_j; \mathbf{K}^\gamma \mathbf{b})\|_2^2 . \end{aligned} \tag{3.3.1}$$

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

3.3.1 Acoustic equation

The snapshot of the p variable for the acoustic equation is given in Figure 3.2. 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

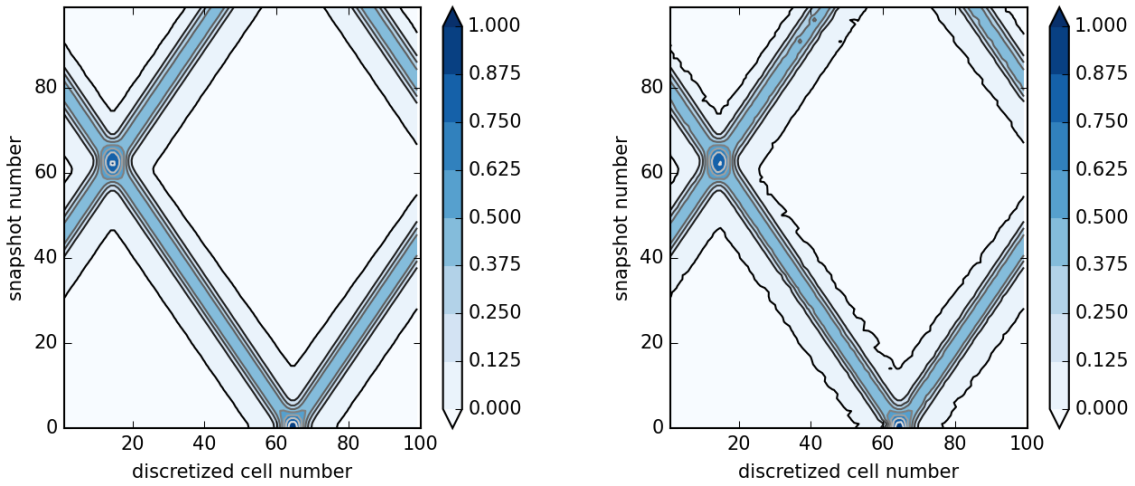

Figure 3.2: Snapshot matrix of the p variable in the acoustic equation (P3) (left) and its approximation via the transport reversal algorithm (right). The L^2-norm of the difference is 2.1841×10^{-3}.

iteration are shown in Figure 3.3. 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 of between the two profiles when they pass each other near snapshot number 60. Notice how in the computed shift numbers for $k = 1$ shown in Figure 3.3 the profile is transported first to the right, 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 (3.2.32) so that the second derivative of the shift numbers are kept small.

3.3.2 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 3.4. The entire snapshot matrix and its approximate reconstruction are shown in Figure 3.5. The total number of iterations was $K = 30$ and pivoting occured once at iteration 19.

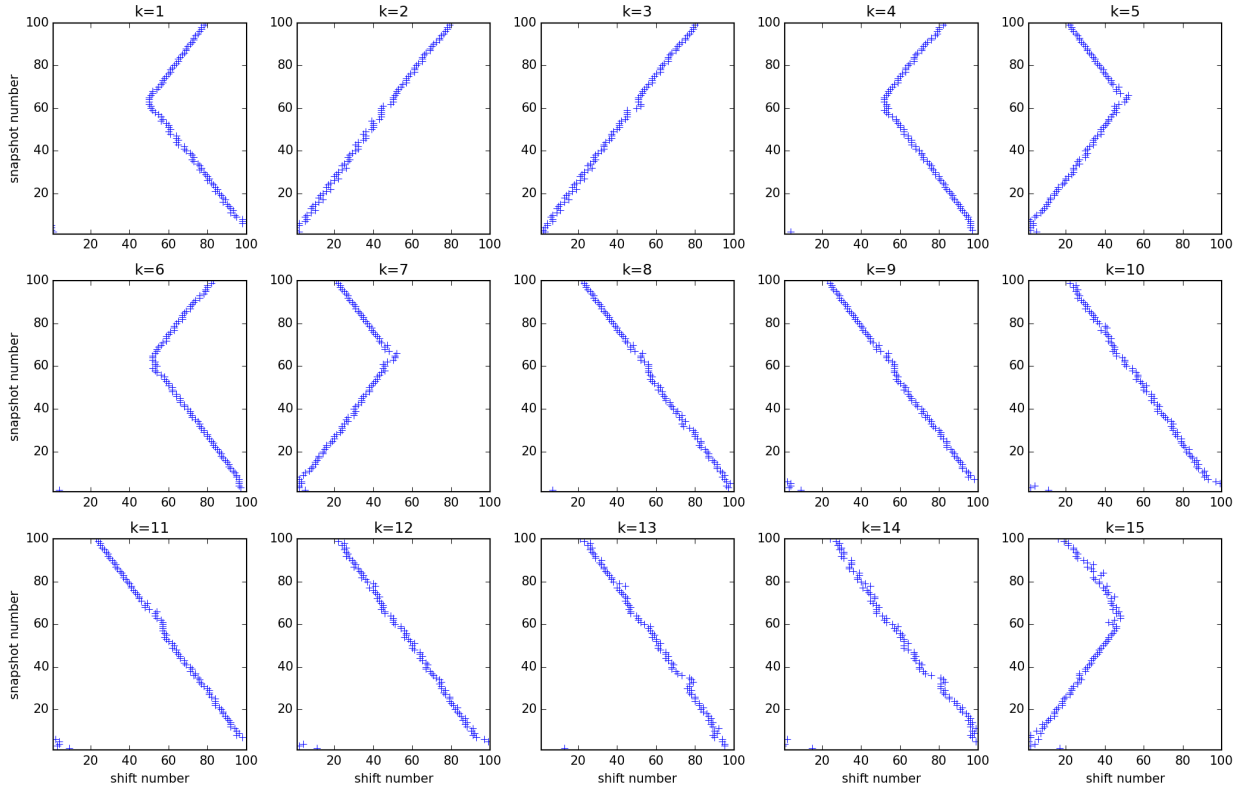

Figure 3.3: Shift numbers $\boldsymbol{\nu}_k$ (3.2.29) for each iteration k, for the acoustic equations example. Single pivot (the initial condition) was used.

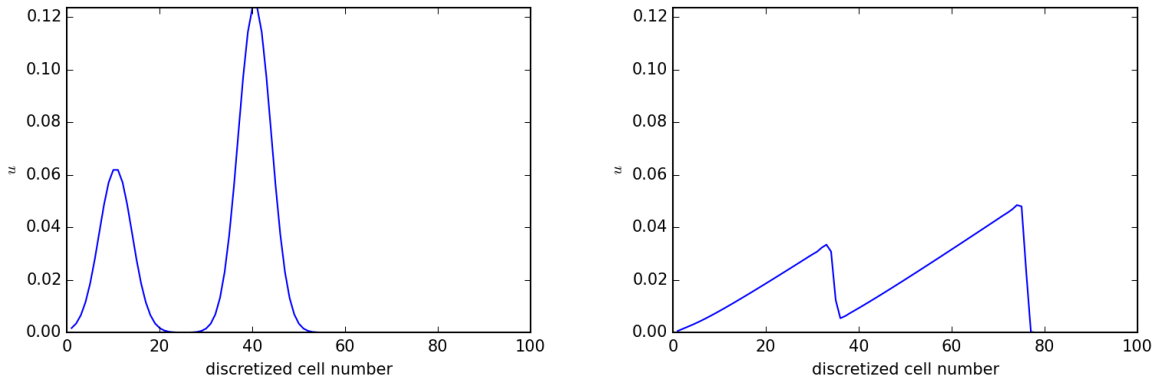

Figure 3.4: Initial condition of the Burgers' equation (P4) (left) and its final snapshot (right).

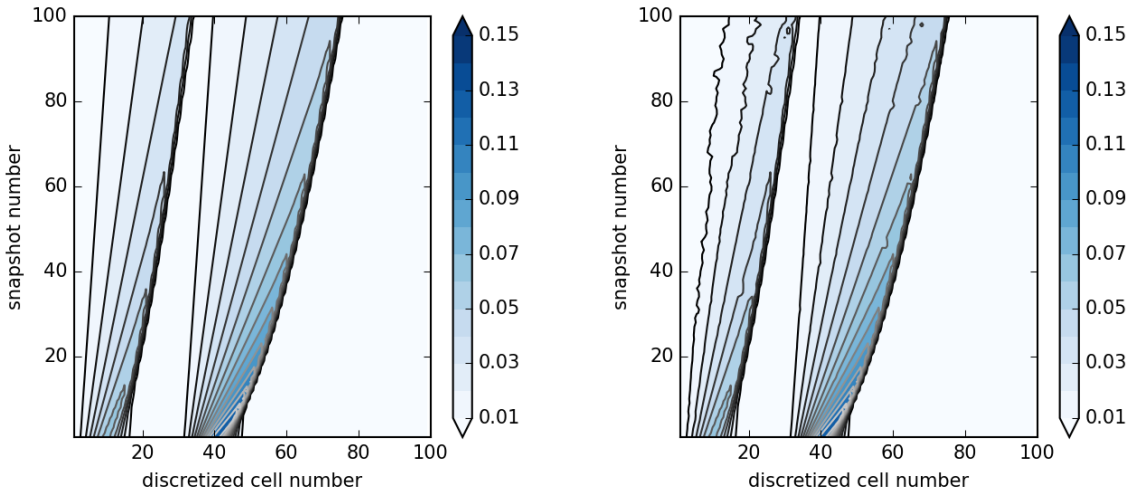

Figure 3.5: Snapshot matrix of the Burgers' equation (P4) (left) and its approximation via the transport reversal algorithm (right). L^2-norm of the difference is 9.6333×10^{-5}.

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 3.6. 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 3.6 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 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

$$h_{jk}\boldsymbol{\rho}_{jk} \odot \mathbf{K}^{\nu_{jk}}\mathbf{a}_1 \quad \text{for } k = 1, 2 \text{ and } j = 15, 50. \tag{3.3.2}$$

These contributions are shown in Figure 3.7. 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

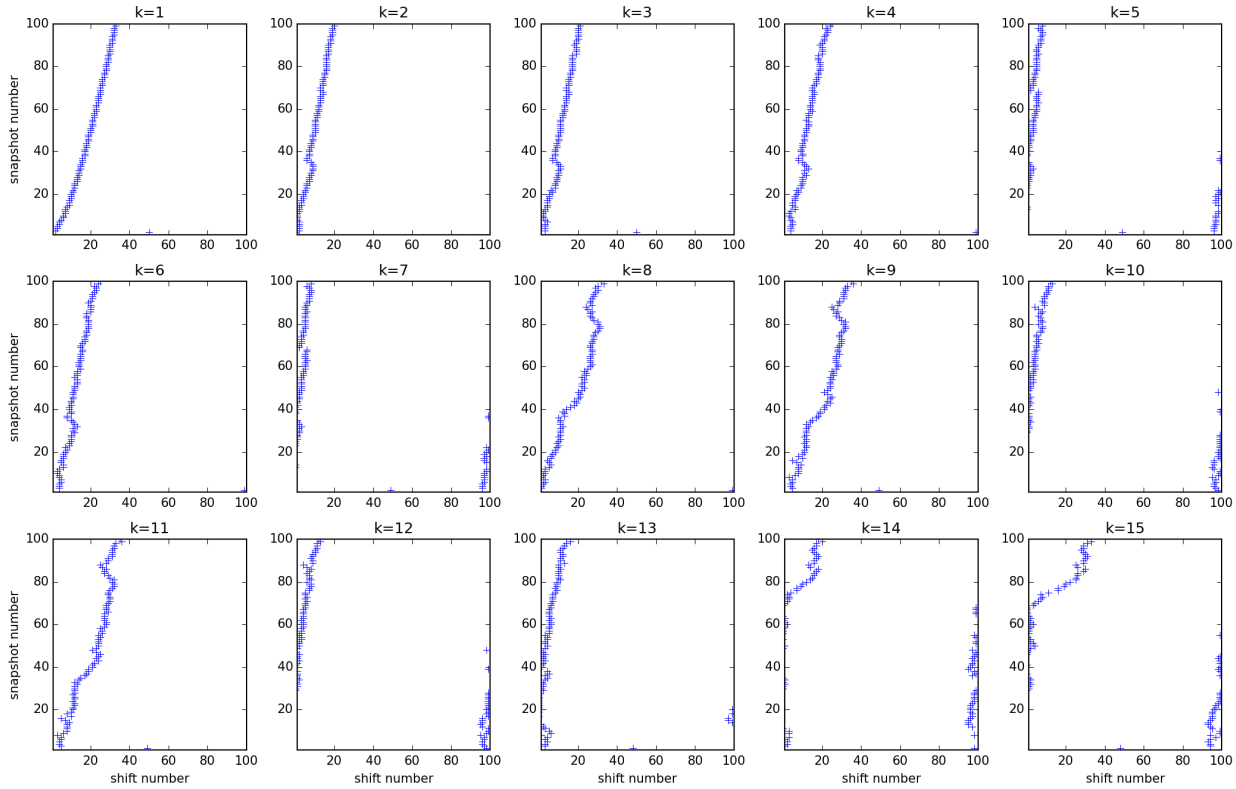

Figure 3.6: Shift numbers $\boldsymbol{\nu}_k$ (3.2.29) for iterations $k = 1$ to 15, for the Burgers' equations example. The first pivot (the initial condition) was used for all iterations shown here.

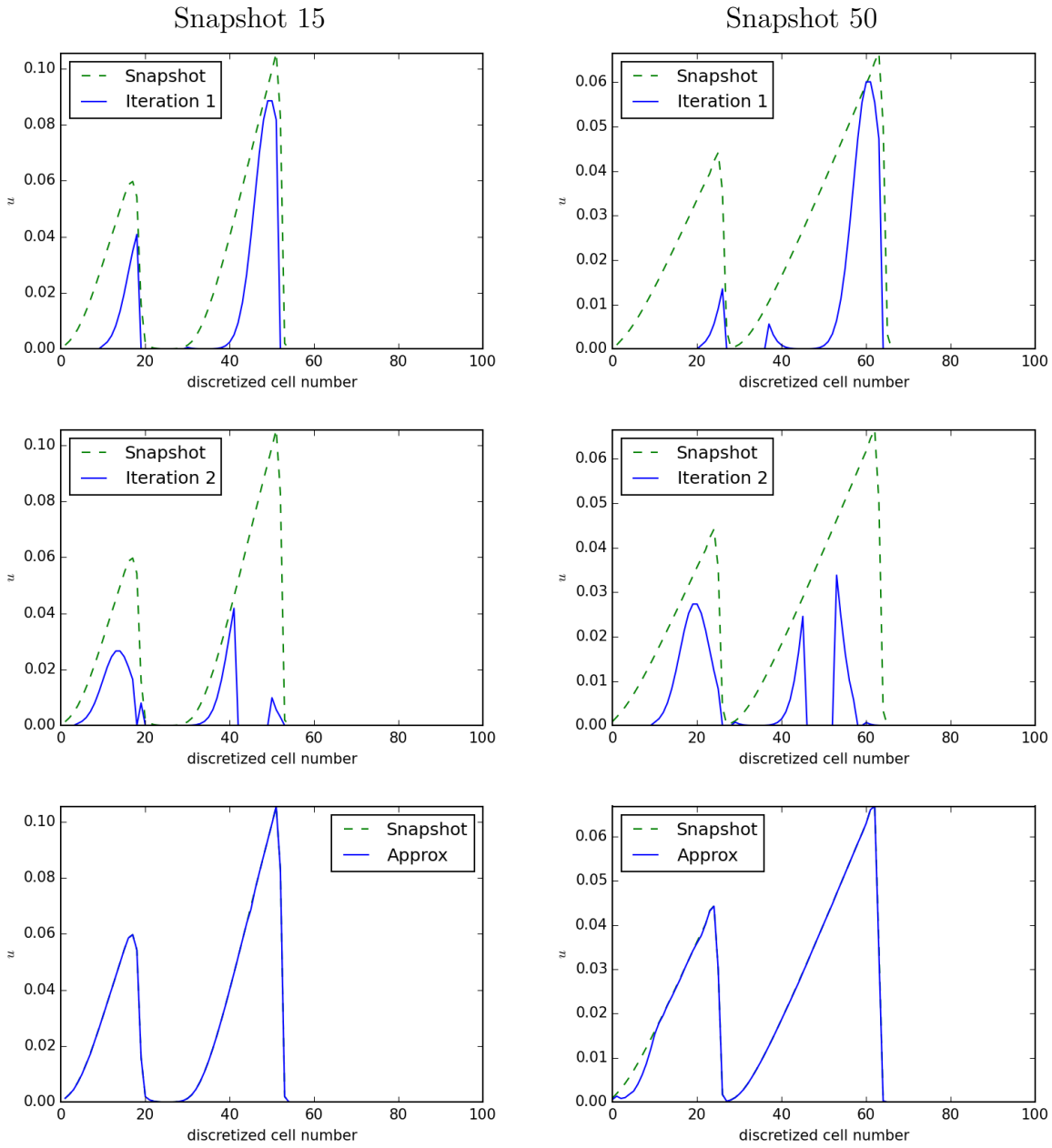

Figure 3.7: Contributions from first two iterations of the transport reversal (rows 1 and 2) for snapshots 15 and 50 (columns 1 and 2) given by the expression (3.3.2). Final approximation at iteration 30 is given in the last row. The contribution from first iteration alone attempts to capture the taller shock to the right (row 1), and that of the second iteration captures the shorter one to the left (row 2).

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.

3.4 Extensions of the shift operator

In this section, we consider extensions of the matrix $\mathbf{K}$ (3.1.4) above. The matrix $\mathbf{K}^\omega$ ($\omega \in \mathbb{Z}_N$) is a basic component of the minimization problem (3.2.4), and two extensions of $\mathbf{K}^\omega$ will be given here. First, we start with an extension using a linear interpolation between $\mathbf{K}^\omega$ and $\mathbf{K}^{\omega+1}$ by an analogy to the upwind flux. This yields a continuous operator $\mathcal{K}(\tilde{\omega})$ over the real numbers ($\tilde{\omega} \in \mathbb{R}$), rather than over integers. Since this operator now causes some numerical diffusion due to its approximation, a reconstruction procedure is introduced to sharpen the profile. The second extension allows the advection velocity in the advection equation (3.1.1) to depend on the spatial variable. This extension is $\mathcal{K}_c(\tilde{\omega})$ with prescribed velocity field c. We also discuss the pivoting procedure that becomes necessary for linear systems.

We remark that the extensions that appear in this section can be used in the greedy algorithm introduced in the previous section. Such use would only require that one change the operator $\mathbf{K}$ above with $\mathcal{K}$ or $\mathcal{K}_c$.

3.5 Extension of K 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 (3.1.1) is given by

$$u_j^{n+1} = u_j^n - \frac{\Delta t}{\Delta x}\left(f_{j+1/2}^n - f_{j-1/2}^n\right). \tag{3.5.1}$$

The upwind flux is defined by $f^n_{j-1/2} \equiv cu^n_{j-1}$, and letting $\nu \equiv c\Delta t/\Delta x$ be the *shift number,* the time-step (3.5.1) can be is expressed as a linear interpolation between u^n_{j-1} and u^n_j,

$$u^{n+1}_j = \left(\frac{c\Delta t}{\Delta x}\right) u^n_{j-1} + \left(1 - \frac{c\Delta t}{\Delta x}\right) u^n_j = \nu u^n_{j-1} + (1-\nu)\, u^n_j. \tag{3.5.2}$$

We write the update in (3.5.1) as a matrix multiplication.

Definition 3.5.1. Let $\mathbf{K}$ be the permutation matrix in (3.1.4). 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 3.5.1. $\mathbf{K}(\nu)$ *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 $0 \le \nu, \omega \le 1$ *and* $\nu + \omega \le 1$, $\mathbf{K}(\nu)\mathbf{K}(\omega) = \mathbf{K}(\nu+\omega) + \mathcal{O}\left(1/N^2\right)$, *where the constant for* $\mathcal{O}\left(1/N^2\right)$ *is a shifted discrete Laplacian (see 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. (a) Follows from the definition since $\mathbf{I}, \mathbf{K}$ commute,

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

(b) Again from the definition,

$$\begin{aligned}\mathbf{K}(\nu)^T\mathbf{K}(\omega) &= \left((1-\nu)\mathbf{I} + \nu\mathbf{K}^T\right)((1-\omega)\mathbf{I} + \omega\mathbf{K}) \\ &= ((1-\omega)\mathbf{I} + \omega\mathbf{K})\left((1-\nu)\mathbf{I} + \nu\mathbf{K}^T\right) = \mathbf{K}(\omega)\mathbf{K}(\nu)^T.\end{aligned}$$

(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, $\mathbf{I}-2\mathbf{K}+\mathbf{K}^2 = \mathbf{K}\left(\mathbf{K}^T+\mathbf{K}-2\mathbf{I}\right) = \frac{1}{N^2}\mathbf{K}\mathbf{L}_h$, 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 $\begin{bmatrix}1 & \cdots & 1\end{bmatrix}$.

□

The notation $\mathcal{O}(1/N^2)$ here and throughout the chapter 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 (3.5.1) 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 3.5.2. *$\mathbf{K}(\nu)^T\mathbf{K}(\nu) = \mathbf{K}(\nu)\mathbf{K}(\nu)^T$ 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*

$$\left\|\mathbf{K}(\nu)^T\mathbf{K}(\nu) - \mathbf{I}\right\|_2 \leq 4\nu(1-\nu). \tag{3.5.3}$$

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

$$\mathbf{K}(\nu)^T\mathbf{K}(\nu) - \mathbf{I} = \nu(1-\nu)(\mathbf{K}+\mathbf{K}^T-2\mathbf{I}) = \frac{\nu(1-\nu)}{N^2}\mathbf{L}_h, \tag{3.5.4}$$

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

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

□

The inequality (3.5.3) 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 [97, 98], as it indicates that the translational actions discretized in such a way no longer form a group; the inverse (3.5.4) and multiplication (Lemma 3.5.1 (c) and Lemma 3.5.3 (c)) are both only approximate and their residuals indicate the presence of numerical diffusion.

Note that $\mathbf{K}(\nu)$ was defined in Definition 3.5.1 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 [64], 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 3.5.2. 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,

$$\mathcal{K}(\tilde{\nu}) \equiv \begin{cases} \mathbf{K}^s\mathbf{K}(\nu) & \text{if } s \geq 0 \\ \left(\mathbf{K}^T\right)^s \mathbf{K}(\nu) & \text{if } s < 0 \end{cases}. \tag{3.5.5}$$

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

$$\mathcal{K}(s,\omega) \equiv \begin{cases} \mathbf{K}^s \mathbf{K}(\omega) & \text{if } s \geq 0 \\ \left(\mathbf{K}^T\right)^s \mathbf{K}(\omega) & \text{if } s < 0 \end{cases}. \tag{3.5.6}$$

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 $-\tilde{\nu} = -(s+1) + (1-\nu)$,

$$\mathcal{K}(\tilde{\nu})^T = \left(\mathbf{K}^T\right)^{s+1} ((1-\nu)\mathbf{I} + \nu\mathbf{K}) = \left(\mathbf{K}^{s+1}\right)^T \mathbf{K}(1-\nu) = \mathcal{K}(-\tilde{\nu}). \tag{3.5.7}$$

All of Lemma 3.5.1 follows through easily.

Lemma 3.5.3. *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*

(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 $\mathcal{K}(\tilde{\nu})\mathcal{K}(\tilde{\omega}) = \mathcal{K}(\tilde{\nu} + \tilde{\omega}) + \mathcal{O}\left(1/N^2\right)$, where the constant for $\mathcal{O}\left(1/N^2\right)$ 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 3.5.1. □

This is a generalization of the upwind method beyond the constraint of the CFL condition, and can be viewed as a special case of large time-step (LTS) method [60], which can also be extended to nonlinear systems [61, 62].

3.6 Transport reversal in $\mathbb{R}$

Using the definitions above, we generalize the minimization problem (3.2.4) to be applied to a snapshot matrix $\mathbf{A}$. We will use the notations $\mathbf{D} \equiv \mathbf{K} - \mathbf{I}$ and $\mathbf{L} \equiv \mathbf{K} + \mathbf{K}^T - 2\mathbf{I}$.

Lemma 3.6.1. *Suppose* $\mathbf{a}, \mathbf{b} \in \mathbb{R}^N$ *are non-constant.*

If we let $\nu = \operatorname{argmin}_{\omega \in \mathbb{R}} \left\| \mathbf{b} - \mathbf{K}(\omega)^T \mathbf{a} \right\|_2^2$, *then*

$$\nu = \frac{1}{2}\left(1 - 2\frac{\mathbf{a}^T \mathbf{D}\mathbf{b}}{\mathbf{a}^T \mathbf{L}\mathbf{a}}\right). \tag{3.6.1}$$

Proof. Let,

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

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 RHS does not vanish. □

Recall that the CFL condition required that $\nu \in [0, 1]$, and note that ν given by (3.6.1) is not guaranteed to lie in this stability region. The minimization (3.2.4) is now extended to the case when $\mathbf{K}$ is replaced by the matrix $\mathcal{K}(\tilde{\nu})$,

$$\min_{\tilde{\omega} \in \mathbb{R}} \left\| \mathbf{a} - \mathcal{K}(\tilde{\omega})^T \mathbf{b} \right\|_2^2. \tag{3.6.2}$$

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 (3.5.4),

$$\min_{\tilde{\omega}\in\mathbb{R}} \left(\left\| \mathbf{b} - \mathcal{K}(\tilde{\omega})^T \mathbf{a} \right\|_2^2 + \mathcal{O}(1/N^2) \right). \tag{3.6.3}$$

The problem (3.6.2) is only of one variable $\tilde{\omega}$ lying in an interval $[0, N]$, although this can be viewed as a non-convex minimization problem in $\mathbb{R}^N$ as we will see in Section 3.9. With the partitioning $\{[m, m+1] : m = 0, \cdots, N-1\}$ of $[0, N]$, a recursive relation can be found for the formula (3.6.1) in terms of j, restricting the variable $\tilde{\omega}$ to a set of $2N-1$ positive reals.

Lemma 3.6.2. *Suppose* $\mathbf{a}, \mathbf{b} \in \mathbb{R}^N$ *and let* $\nu_s^{\pm}$ *be defined as*

$$\nu_s^+ \equiv \mathrm{argmin}_{\omega\in\mathbb{R}} \left\| \mathbf{b} - \mathcal{K}(s,\omega)^T \mathbf{a} \right\|_2^2, \quad \nu_s^- \equiv \mathrm{argmin}_{\omega\in\mathbb{R}} \left\| \mathbf{b} - \mathcal{K}(-s,\omega)^T \mathbf{a} \right\|_2^2. \tag{3.6.4}$$

Then we have the relations

$$\nu_{s+1}^+ = \nu_s^+ + \frac{(\mathbf{K}^s\mathbf{b})^T \mathbf{L}\mathbf{a}}{\mathbf{a}^T\mathbf{L}\mathbf{a}}, \quad \nu_{s+1}^- = \nu_s^- + \frac{\left(\left(\mathbf{K}^T\right)^s \mathbf{b}\right)^T \mathbf{L}\mathbf{a}}{\mathbf{a}^T\mathbf{L}\mathbf{a}}. \tag{3.6.5}$$

Proof. Consider the $(+)$ case, the $(-)$ case follows similarly.

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

Using this in formula (3.6.1),

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

Then we have

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

□

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 (3.6.2) only requires comparison of at most $2N-1$ discrete values,

$$\tilde{\nu} = \text{argmin}_{\tilde{\omega} \in W} \left\| \mathbf{b} - \mathcal{K}(\tilde{\omega})^T \mathbf{a} \right\|_2^2 \text{ where } W \equiv \{0, \hat{\nu}_0^+, 1, \hat{\nu}_1^+, \cdots, N-1, \hat{\nu}_{N-1}^+\}, \tag{3.6.6}$$

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

In many examples the data $\mathbf{a}$ and $\mathbf{b}$ have localized features. This fact can be incorporated into our computation of (3.6.2) 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 3.9.1.)

Definition 3.6.1 (Transport reversal in $\mathbb{R}$)**.** Given a matrix $\mathbf{A} \in \mathbb{R}^{N \times M}$, let $\mathbf{a}_j$ denote the j-th column of $\mathbf{A}$, and let $\mathbf{b} \in \mathbb{R}^N$ be a given *pivot*. Let

$$\tilde{\nu}_j \equiv \text{argmin}_{\tilde{\omega} \geq 0} \left\| \mathbf{a}_j - \mathcal{K}(\tilde{\omega})^T \mathbf{b} \right\|_2^2, \quad \text{for } j = 1, \cdots, M. \tag{3.6.7}$$

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

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

$$\mathcal{T}(\mathbf{A}; \tilde{\boldsymbol{\nu}}) \equiv \begin{bmatrix} \mathcal{K}(\tilde{\nu}_1) & \mathcal{K}(\tilde{\nu}_2) & \cdots & \mathcal{K}(\tilde{\nu}_M) \end{bmatrix} \odot \begin{bmatrix} \mathbf{a}_1 & \mathbf{a}_2 & \cdots & \mathbf{a}_M \end{bmatrix}. \tag{3.6.8}$$

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 Proposition 3 in [7] and is stated as follows.

Proposition 3.6.3. *If φ is an eigenvector of $\mathring{\mathbf{A}}^T\mathring{\mathbf{A}}$ with eigenvalue λ, then φ is also an eigenvector of $\left(\mathcal{K}(\tilde{\nu})\mathring{\mathbf{A}}\right)^T\left(\mathcal{K}(\tilde{\nu})\mathring{\mathbf{A}}\right)$ to the same λ for every $\tilde{\nu}\in\mathbb{R}$, up to $\mathcal{O}(1/N^2)$.*

Proof. Follows immediately from (3.5.4) in Lemma 3.5.2. □

3.7 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 (3.5.4) in the form of a discrete Laplacian $\mathbf{L}_h$. This can be utilized to improve the accuracy by applying a post processing 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

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

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 $\mathbf{a}$,

$$(\mathbf{I}+\alpha\mathbf{L}_h)\,\mathbf{u} = \mathbf{b}, \quad \text{satisfying} \quad u_1 = a_1 \quad \text{and} \quad u_N = a_N. \tag{3.7.1}$$

This inversion acts to remove the diffusive error caused by grid interpolation (3.5.2). 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 (3.7.1) 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,

$$\left[(\mathbf{I}+\alpha_1\mathbf{L}_h)^{-1}\mathcal{K}(\tilde{\nu}_1) \quad \cdots \quad (\mathbf{I}+\alpha_{M-1}\mathbf{L}_h)^{-1}\mathcal{K}(\tilde{\nu}_{M-1})\right] \odot \left[\tilde{\mathbf{a}}_1 \quad \cdots \quad \tilde{\mathbf{a}}_M\right]. \tag{3.7.2}$$

As mentioned in remarks following Lemma 3.5.2, 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 3.8.3.

3.8 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 (3.1.7) when the advection speed $c > 0$ in (3.1.1) 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 (3.6.7) has largely been a discretization of the continuous minimization problem (3.1.7) with some numerical error (3.5.3). 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 $\Delta x_j = x_{j+1} - x_j$,

$$|\overline{\omega} - \omega_j| < \delta \ll 1 \quad \text{where } \omega_j = \frac{c(x_{j+1/2})}{\Delta x_j} \quad \text{for } j = 0, \cdots, N. \tag{3.8.1}$$

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 (3.8.1) is trivial. Let us give a simple example of c and $\{x_j\}_{j=0}^N$ that satisfies this property. Consider the advection equation (3.1.1) 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 (3.8.1) holds with $\bar{\omega} = 2N/3$ and $\delta = 0$.

However, the property (3.8.1) 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 3.8.1, even if c does not satisfy the property (3.8.1).

3.8.1 Variable speed reversal

We proceed to generalize the transport reversal procedure by employing the large time-step (LTS) method [60, 61, 62]. This is also reminiscent of Lagrangian methods such as the particle-in-cell method [29, 50] or its variant the material point method [105, 118]. 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

$$\begin{cases} \dot{x_j} = c(x_j), \\ x_j(0) = x_j^0, \end{cases} \quad \text{for } j = 0, 1, \cdots, N, \tag{3.8.2}$$

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

$$x_j(t^L) = x_j(0) + \int_0^{t^L} c\left(x_j(t)\right)\, \mathrm{d}t. \tag{3.8.3}$$

Recall $c(x)$ was assumed to be piecewise constant, so we let $c_j \equiv c(x_{j+1/2})$ and we compute (3.8.3) 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 i-th cell $\mathcal{C}_i$ during the time interval $[t_\ell, t_{\ell-1}]$, $\Delta t_{ij}^\ell \equiv -\left|\{t \in [t_{\ell-1}, t_\ell] : x_j(t) \in \mathcal{C}_i\}\right|$. 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

$$x_j^{\ell+1} = x_j^\ell + \sum_{i=1}^N c_i \Delta t_{ij}^\ell \qquad \mod 1. \tag{3.8.4}$$

Once time stepping has reached the final time t^L, we can update the cell average by computing the total flux for each cell,

$$u_j^{n+1} = u_j^n + \sum_{i=1}^N \Delta_i^n c_i \Delta t_{ij}^\ell. \tag{3.8.5}$$

The procedure is sketched in Figure 3.8.

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 (3.8.2), so that $x_j(t+T) = x_j(t)$. Recall

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 $\mathcal{C}_2, \mathcal{C}_3, \mathcal{C}_4$.)

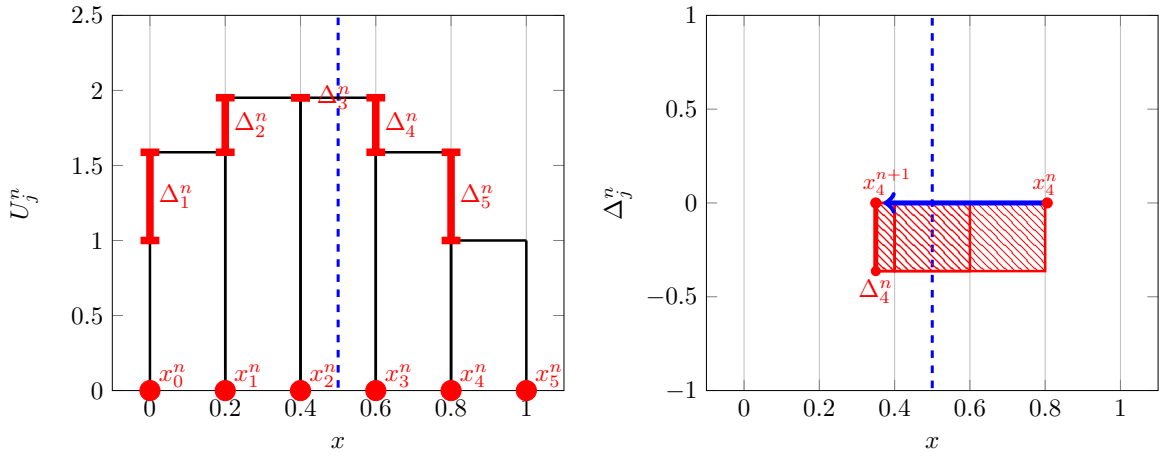

Figure 3.8: An illustration of the variable speed advection with periodic boundary conditions. The dotted blue line at $x = 0.5$ denotes the interface where c changes. After the fluxes for each cell is computed, we update u_j^n by computing the total change of volume as in (3.8.5), resulting in u_j^{n+1}.

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

$$\bar{c} \equiv \frac{|\Omega|}{T} = \frac{1}{|\Omega|} \int_{\Omega} c(x)\, \mathrm{d}x, \tag{3.8.6}$$

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 (3.6.2) by a negation of the argument using (3.5.7), we have

$$\min_{\tilde{\omega} \in \mathbb{R}} \|\mathbf{a}_j - \mathcal{K}_c(-\tilde{\omega})\mathbf{b}\|_2^2 . \tag{3.8.7}$$

Now we define the reversal for the variable speed case.

Definition 3.8.1 (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 j-th column of $\mathbf{A}$. Let $\mathbf{b} \in \mathbb{R}^N$ be a given *pivot*. Then compute

$$\tilde{\nu}_j = \mathrm{argmin}_{\tilde{\omega} \in \mathbb{R}} \|\mathbf{a}_j - \mathcal{K}_c(-\tilde{\omega})\mathbf{b}\|_2^2, \quad \text{for } j = 1, \cdots, M. \tag{3.8.8}$$

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

$$\mathcal{T}_c(\mathbf{A}; \boldsymbol{\nu}) \equiv \begin{bmatrix} \mathcal{K}_c(\tilde{\nu}_1) & \cdots & \mathcal{K}_c(\tilde{\nu}_M) \end{bmatrix} \odot \begin{bmatrix} \mathbf{a}_1 & \cdots & \mathbf{a}_M \end{bmatrix}. \tag{3.8.9}$$

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

$$\mathcal{T}_c(\mathbf{A}; -\tilde{\boldsymbol{\nu}}) \equiv \begin{bmatrix} \mathcal{K}_c(-\tilde{\nu}_1) & \cdots & \mathcal{K}_c(-\tilde{\nu}_M) \end{bmatrix} \odot \begin{bmatrix} \mathbf{a}_1 & \cdots & \mathbf{a}_M \end{bmatrix}. \tag{3.8.10}$$

The orthogonality condition in Proposition 3.6.3 still holds with small error. Also, no simple relation such as (3.6.5) are found, and the sharpening procedure (3.7.2) cannot be easily applied. This is due to the loss of convexity to be discussed in Section 3.9.

3.8.2 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),

$$\begin{bmatrix} p \\ u \end{bmatrix}_t + \begin{bmatrix} 0 & K \\ 1/\rho & 0 \end{bmatrix} \begin{bmatrix} p \\ u \end{bmatrix}_x = 0. \tag{3.8.11}$$

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

$$\lambda^1 = u - c, \quad \mathbf{r}_1 = \begin{bmatrix} -\rho c \\ 1 \end{bmatrix} \quad \text{and} \quad \lambda^2 = u + c, \quad \mathbf{r}_2 = \begin{bmatrix} \rho c \\ 1 \end{bmatrix}. \tag{3.8.12}$$

where $c \equiv \sqrt{K/\rho}$. We can rewrite the equation (3.8.11) 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 (3.8.12). 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, 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 3.12 below.

Therefore we need to dynamically change the pivot vector $\mathbf{b}$ in (3.8.7) appropriately to other columns as the wave profile evolves. Let us recall the *pivot map* $\ell : \{1, \cdots, M\} \to \{1, \cdots, M\}$ 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 3.8.2 (Variable speed transport reversal with pivoting)**.** Let the matrix $\mathbf{A} \in \mathbb{R}^{N\times M}$ be given, a pivot map $\ell : \{1, ..., M\} \to \{1, ..., M\}$, and a variable speed $c : [0, N] \to \mathbb{R}$. Then let

$$\tilde{\nu}_j \equiv \text{argmin}_{\tilde{\omega}\in\mathbb{R}} \left\|\mathbf{a}_j - \mathcal{K}_c(-\tilde{\omega})\mathbf{a}_{\ell(j)}\right\|_2^2, \quad \text{for } j = 1, \cdots, M. \tag{3.8.13}$$

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

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 3.8.3 for numerical results using this particular pivoting criterion.

3.8.3 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 3.2, and consider relatively simple examples to focus on the effect of the extensions of the shift operator.

Acoustic equation in homogeneous media

We apply the reversal (3.6.8) to the constant speed acoustic equation (3.8.11) 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 (3.8.12) we transform the state variables p and u, to the characteristic

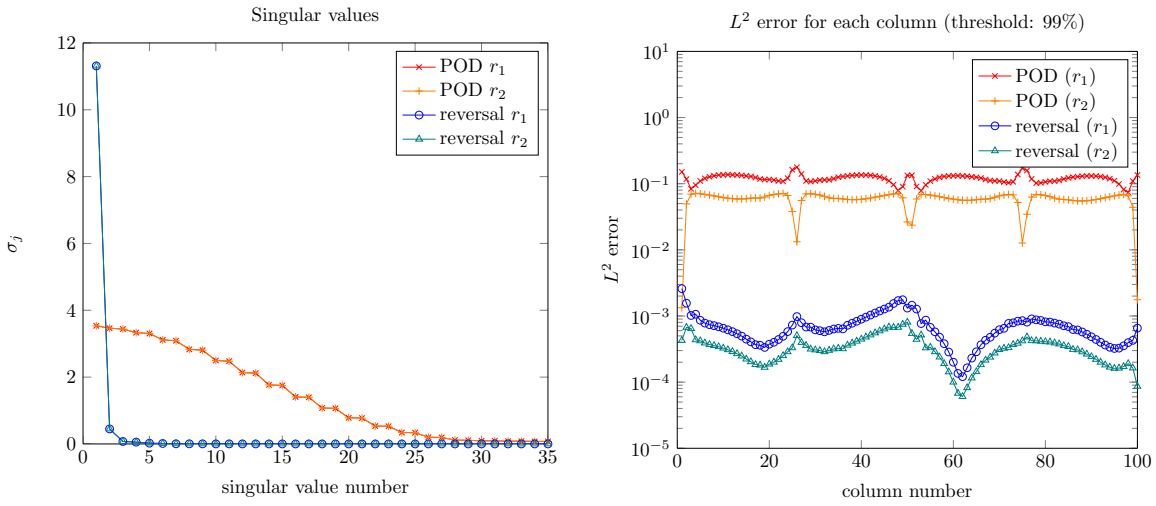

Figure 3.9: Left: Fast decay of singular values of snapshots in the variables r_1 and r_2 for the acoustic equation with homogeneous media. The largest 35 singular values are shown. Three singular values represent 99% of the threshold for both variables. Right: L^2 error for each column of the reconstruction for eigenvector variables $r_1$and r_2.

variables r_1 and r_2. Then we apply the reversal procedure (3.6.8) 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 3.9. The threshold of 99% is achieved with only 3 basis vectors. The reconstruction is plotted against the snapshot itself in Figure 3.10, and they are nearly identical. The POD reconstruction is also plotted. The L^2-errors from the two reconstructions are compared in Figure 3.9, where the reversal consistently outperforms the naïve POD.

Acoustic equation with heterogeneous media

Now we consider the acoustic equation (3.8.11) with heterogeneous media, with two different materials. An interface will be located at $x = 0.5$. Letting ℓ designate *left* part of the domain $(0, 0.5)$ and r the *right* part of the 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 $\rho_r = 4$, $K_r = 1$.

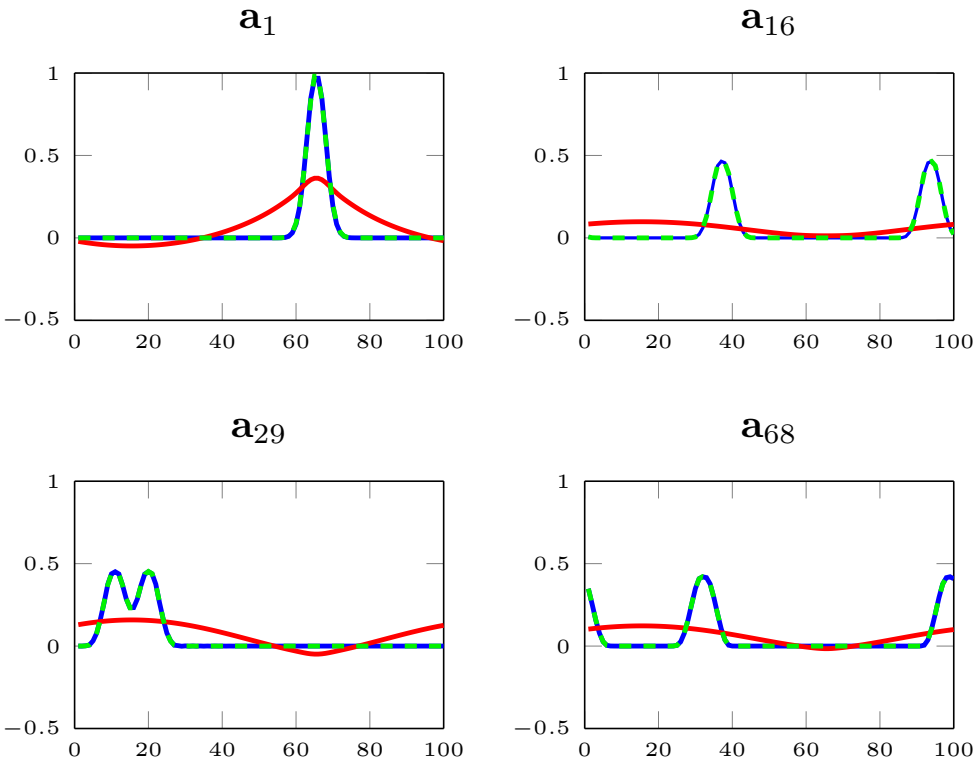

Figure 3.10: Reconstruction of the solution to acoustic equation with homogeneous media, for state variable p. The snapshot is given in dashed green, the reversal reconstruction in blue, and POD reconstruction in red. Three reduced basis vectors were used for both reversal and POD.

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 condition p_0 and u_0 are both Gaussian humps of identical shape traveling towards 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 (3.8.13) 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

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

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 3.11. 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 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 3.12. 7 and 5 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.

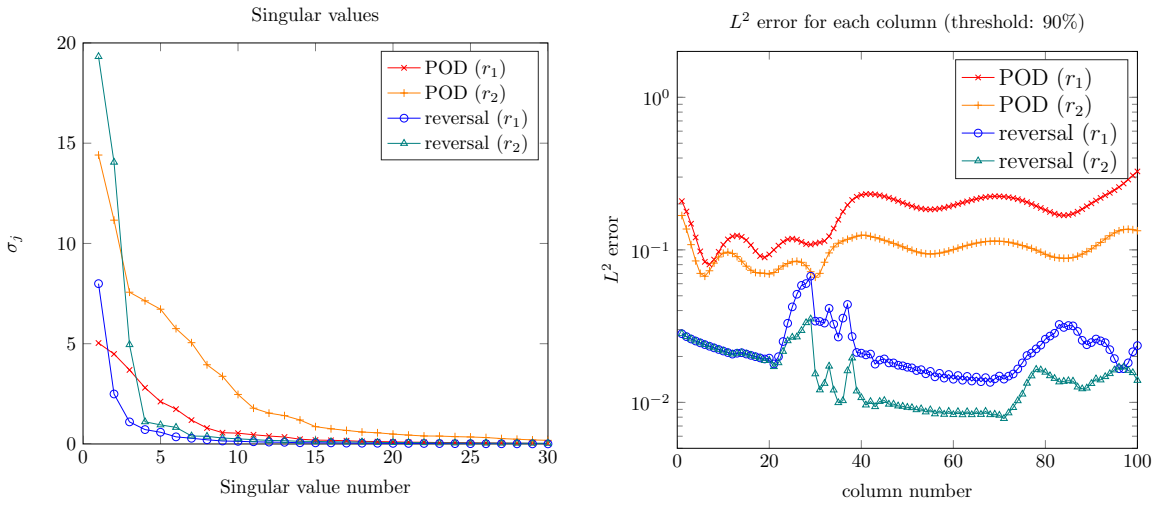

Figure 3.11: Faster decay of singular values of r_1 and r_2 for the acoustic equation (3.1.1) with heterogeneous media (left). Largest 30 singular values are shown. 5 and 7 singular values represent 90% threshold for these variables, respectively. L^2-error for each column of reconstruction for characteristic variables r_1 and r_2. The error is concentrated near the snapshots (20-40) in which the wave profile is undergoing rapid change near the interface.

3.9 Geometric interpretation

In this section, we present some geometric interpretations of the reversal procedure introduced in the Sections 3.2 and 3.8 that arise naturally. Recall the matrices $\mathbf{K}(\nu)$ (Def. 3.5.1), $\mathcal{K}(\tilde{\nu}) = \mathcal{K}(s,\nu)$ (3.5.5) and variable speed operator $\mathcal{K}_c(\tilde{\nu})$ (3.8.7). Let us define,

$$\mathcal{I}_s(\mathbf{a}) \equiv \{\mathcal{K}(s,\nu)\mathbf{a} : \nu \in [0,1]\} \text{ and } \mathcal{M}(\mathbf{a}) \equiv \bigcup_{s\in\mathbb{Z}} \mathcal{I}_s = \{\mathcal{K}(\tilde{\nu})\mathbf{a} : \tilde{\nu} \in \mathbb{R}\}. \tag{3.9.1}$$

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 $\mathbb{R}^N$ (3.9.1), although $\mathcal{M}(\mathbf{a})$ is not convex in $\mathbb{R}^N$ in general.

In the minimization problem (3.6.2) 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

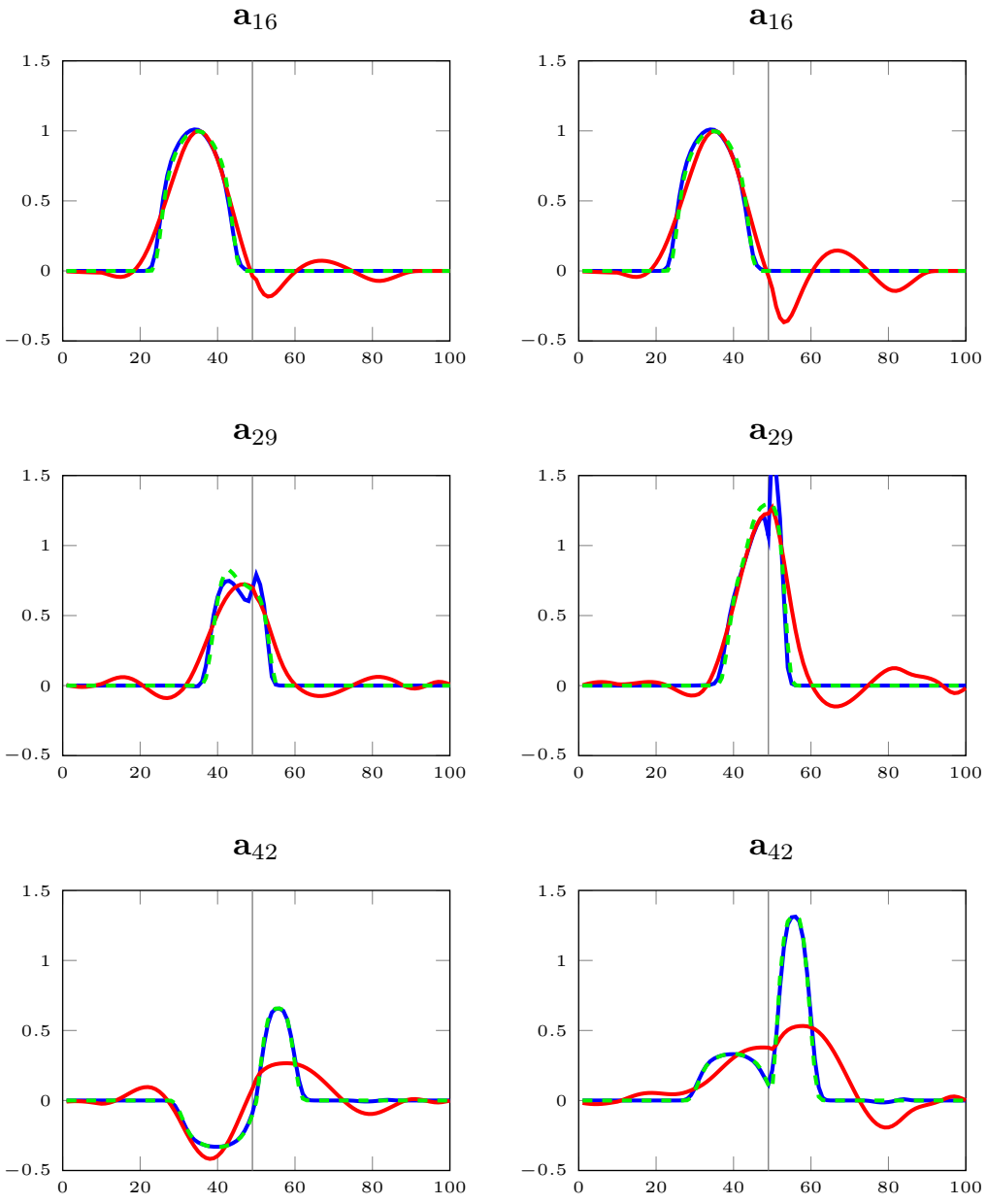

Figure 3.12: Reconstruction of the solution to acoustic equation for state variable p (left column) and u (right column). The snapshot is given in dashed green, reversal reconstruction in blue, and POD reconstruction in red. The interface is displayed as a vertical gray line. Four reduced vectors were used for each reconstruction.

$\|\mathbf{a}\|_2$ but $\sum_{j=1}^{N}(\mathbf{a})_j$ due to mass conservation in Lemma 3.5.1 (d). This is also reflected in the $\mathcal{O}(1/N^2)$ numerical diffusion term in $\mathcal{K}(\tilde{\nu})^T\mathcal{K}(\tilde{\nu}) - \mathbf{I}$ (3.5.3).

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 $-\mathbf{a}^T\left(\mathbf{K}+\mathbf{K}^T-2\mathbf{I}\right)\mathbf{a}/\left\|\mathbf{D}\mathbf{a}\right\|_2^2$.

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 (3.6.2). 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 respected 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 on the $\mathcal{M}(\mathbf{a})$ becomes more constrained 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 (3.6.7) can always be found. That is, suppose $\mathbf{a}, \mathbf{b} \in \mathbb{R}^N$, then there always exists a shift number $\tilde{\nu}$ minimizing $\left\|\mathbf{b} - \mathcal{K}(\tilde{\nu})^T\mathbf{a}\right\|_2$. $\|\mathbf{b}-\mathbf{c}\|_2^2$ 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 (3.2.32) 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 [98], the minimization (3.6.2) 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 (3.6.2), 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 3.9.1. *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*

$$L = \frac{N}{\gcd\left[\operatorname{supp}\mathcal{F}(\mathbf{a}) \setminus \{0\}\right]} \tag{3.9.2}$$

Proof. 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, 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 $\left(\mathbf{K}^T\right)^{L-\alpha}\mathbf{K}(\alpha)^T\mathbf{a} = \mathbf{a}$. 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 non-constant $\mathbf{a}$ since $-\mathbf{L}$ is positive semi-definite with nullspace equal to that of constant vectors. Hence α cannot be in $(0, 1)$ it so must be zero.

Now since L is an integer modulo N let us assume $L > 0$ without loss of generality, 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 the equation (3.9.2). □

Given this dimension L in (3.9.2), we can reduce the discrete set W in (3.6.5) 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 [98].

In the variable speed case, the polygon $\mathcal{M}$ no longer retains its regularity. Let us define a variable speed counter part to (3.9.1),

$$\mathcal{I}_{c,s}(\mathbf{a}) \equiv \{\mathcal{K}_c(\tilde{\nu})\mathbf{a} : \nu \in [s, s+1]\} \text{ and } \mathcal{M}_c(\mathbf{a}) \equiv \bigcup_{s\in\mathbb{Z}} \mathcal{I}_s = \{\mathcal{K}_c(\tilde{\nu})\mathbf{a} : \nu \in \mathbb{R}\}. \tag{3.9.3}$$

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 (3.8.7) more challenging and causes the sharpening procedure (3.7.2) to run into difficulties, outside simple special cases for which one may impose more boundary conditions (3.7.1) near the interface.

3.10 Summary and discussion

This chapter introduced a greedy algorithm that extracts the transport structure from the snapshot matrix by building on the template fitting strategy. Extensions of the algorithm through 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 post-processing 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.

Chapter 4

Dimensional splitting with the Radon transform

Dimensional splitting provides the simplest approach to obtaining a multi-dimensional method from a one-dimensional method [64, 74, 104]. Although extremely powerful, existing splitting methods do not preserve a special feature that is easily obtained for 1D methods. For 1D hypebolic partial differential equations (PDEs) of the type

$$q_t + Aq_x = 0 \tag{4.0.1}$$

where A is a constant diagonalizable matrix with real and distinct eigenvalues, one can devise large time-step (LTS) methods that allow the solution to be solved up to any time without incurring excessive numerical diffusion [60, 61, 62]. Previous splitting methods do not lead to such LTS methods in multi-dimensions.

In this chapter, we introduce a dimensional splitting method that allows multi-dimensional linear constant coefficient hyperbolic problems to be solved up to desired time. The method relies on the intertwining property of Radon transforms [51, 80], thereby transforming a multi-dimensional problem into a family of one-dimensional ones. Simply by applying an 1D LTS method on each of these one-dimensional problems, one obtains a multi-dimensional LTS method. While this intertwining property is well-known and appears in standard references, it has not been used for constructing multi-dimensional numerical methods, to the best of our knowledge.

The method also has implications for the problem of imposing absorbing boundary conditions, a problem that has received sustained interest over many decades [11, 12, 22, 32]. By using the Radon transform, the splitting decomposes multi-dimensional waves into planar

ones, thereby allowing a separate treatment of each incident planar wave near the boundary. This yields the desired absorbing boundary conditions in odd dimensions, and in even dimensions one obtains an approximation up to $\mathcal{O}(1/t)$ that does not cause spurious reflections.

Another useful application of this dimensional splitting is in displacement interpolation, a concept that arises naturally in optimal transport [113]. Our interest in displacement interpolation is motivated by model reduction. To construct reduced order models for typical hyperbolic problems, one cannot rely solely on linear subspaces [1, 21], and it is necessary to interpolate over the Lagragian action [95, 96, 97, 98]. In a single spatial dimension this can be done in a relatively straightforward manner, owing to the LTS methods available for 1D [96]. The multi-dimensional LTS method is useful also for the multi-dimensional extension of displacement interpolation, and this in turn will yield a straightforward way for low-dimensional information to be extracted for multi-dimensional hyperbolic problems.

For the dimensional splitting to be computationally successful, one requires an algorithm for computing the Radon transform and its inverse efficiently. Throughout this chapter we use the approximate discrete Radon transform (ADRT), also called simply the discrete Radon transform (DRT), devised in [18, 45]. Here we will refer to ADRT as DRT. It is a fast algorithm with the computational cost of $\mathcal{O}(N^2 \log N)$ for an $N \times N$ image or grid,[1] and the efficiency is obtained through a geometric recursion of so-called *digital lines*. The inversion algorithm using the full multi-grid method appeared in [88], but here we adopt a simpler approach by making use of the conjugate gradient algorithm [47] for the inversion. We conjecture the worst-case cost of the inversion to be $\mathcal{O}(N^{5/2} \log N)$.

This chapter is organized as follows. In Section 4.1, we give a review of the intertwining property of the Radon transform and introduce the dimensional splitting method. In Section 4.2, discuss its applications in absorbing boundary and in displacement interpolation. In Section 4.3, we give a brief introduction to the DRT algorithm and discuss its inversion. In this chapter we will fully implement only constant coefficient linear problems in spatial

[1]The term *grid* (*cell*) is a more appropriate term for our PDE applications, but DRT originally comes from imaging literature so we will sometimes also use the term *image* (*pixel*), interchangeably.

dimension two, although we will also discuss how the splitting can be extended to fully nonlinear problems and to higher spatial dimensions. Further investigations into these and other related topics will be mentioned in Section 4.5.

The Radon transform was introduced by Johann Radon [90] and has been a major subject of study, perhaps primarily due to its use in medical imaging but also as a general mathematical and computational tool.

4.1 Dimensional splitting using the Radon transform

In this section, we briefly review the intertwining property of the Radon transform [51, 80] then show that it can be used as a dimensional splitting tool that extends the large time-step (LTS) operator to multiple spatial dimensions. It preserves the ability to take large time-steps without loss of accuracy in the constant coefficient case. Moreover, this splitting can be used for fully nonlinear problems as well, in a similar manner to the other splitting methods, with the usual CFL condition for the time-step.

4.1.1 Intertwining property of Radon transforms

The *Radon transform* $\hat{\varphi} : S^{n-1} \times \mathbb{R} \to \mathbb{R}$ of the function $\varphi : \mathbb{R}^n \to \mathbb{R}$ is defined as

$$\hat{\varphi}(\boldsymbol{\omega}, s) = \mathcal{R}\varphi(\boldsymbol{\omega}, s) = \int_{\langle \mathbf{x}, \boldsymbol{\omega} \rangle = s} \varphi(\mathbf{x}) \, \mathrm{d}m(\mathbf{x}), \tag{4.1.1}$$

in which $\mathrm{d}m$ is the Euclidean measure over the hyperplane. For any fixed pair $(\boldsymbol{\omega}, s) \in S^{n-1} \times \mathbb{R}$, the set $\{\mathbf{x} \in \mathbb{R}^n : \langle \mathbf{x}, \boldsymbol{\omega} \rangle = s\}$ defines a hyperplane, so the transform is simply an integration of the function over this hyperplane. In effect, $\hat{\varphi}$ decomposes φ into planar waves in the direction of $\boldsymbol{\omega}$.

The *back-projection* is defined as the dual of $\mathcal{R}$ with respect to the obvious inner product over $S^{n-1} \times \mathbb{R}$. For $\psi : S^{n-1} \times \mathbb{R} \to \mathbb{R}$ the back-projection $\check{\psi}$ is

$$\check{\psi}(\mathbf{x}) = \mathcal{R}^{\#}\psi(\mathbf{x}) = \int_{S^{n-1}} \psi(\boldsymbol{\omega}, \langle \boldsymbol{\omega}, \mathbf{x} \rangle) \, \mathrm{d}S(\boldsymbol{\omega}), \tag{4.1.2}$$

where $\mathrm{d}S$ is the measure on S^{n-1} and $\langle \cdot, \cdot \rangle$ denotes the usual inner product in $\mathbb{R}^n$.

The Radon transform has a remarkable property, that it intertwines a partial derivative with a univariate derivative. The i-th partial derivative $\partial/\partial x_i$ of φ is now transformed to the derivative of $\hat{\varphi}$ with respect to s multiplied by ω_i,

$$\left(\frac{\partial}{\partial x_i}\varphi(\mathbf{x})\right)^{\wedge} = \omega_i \frac{\partial}{\partial s}\hat{\varphi}(\boldsymbol{\omega}, s). \tag{4.1.3}$$

This is the key property that allows us to transform a multi-dimensional hyperbolic problem into a collection of one-dimensional problems. For example, let us apply the Radon transform to the transport equation in $\mathbb{R}^2$, in which the scalar state variable $q : \mathbb{R}^+ \times \mathbb{R}^2 \to \mathbb{R}$ satisfies,

$$q_t + \boldsymbol{\theta} \cdot \nabla q = 0 \quad \text{where} \quad \boldsymbol{\theta} \in S^1. \tag{4.1.4}$$

The transformation produces a family of 1D advection equations

$$\hat{q}_t + (\boldsymbol{\theta} \cdot \boldsymbol{\omega})\hat{q}_s = 0, \tag{4.1.5}$$

whose coefficient varies for each $\boldsymbol{\omega}$. Similarly, consider the acoustic equations for $p, u, v : \mathbb{R}^+ \times \mathbb{R}^2 \to \mathbb{R}$, where the state variable p denotes pressure, u the velocity in x-direction, v the velocity in y direction,

$$\begin{bmatrix} p \\ u \\ v \end{bmatrix}_t + \begin{bmatrix} 0 & K_0 & 0 \\ 1/\rho_0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} p \\ u \\ v \end{bmatrix}_x + \begin{bmatrix} 0 & 0 & K_0 \\ 0 & 0 & 0 \\ 1/\rho_0 & 0 & 0 \end{bmatrix} \begin{bmatrix} p \\ u \\ v \end{bmatrix}_y = 0. \tag{4.1.6}$$

After the transform, we obtain

$$\begin{bmatrix} \hat{p} \\ \hat{u} \\ \hat{v} \end{bmatrix}_t + \begin{bmatrix} 0 & \omega_1 K_0 & \omega_2 K_0 \\ \omega_1/\rho_0 & 0 & 0 \\ \omega_2/\rho_0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \hat{p} \\ \hat{u} \\ \hat{v} \end{bmatrix}_s = 0. \tag{4.1.7}$$

This PDE has one spatial dimension in variable s. Letting $\mu = \omega_1 u + \omega_2 v$ and $\nu = -\omega_2 u + \omega_1 v$, (4.1.7) can be rewritten as three equations for new states $\hat{p}, \hat{\mu}$ and $\hat{\nu}$. If one omits the trivial equation $\nu_t = 0$, the equation (4.1.7) is reduced to the 1D acoustic equations,

$$\begin{bmatrix} \hat{p} \\ \hat{\mu} \end{bmatrix}_t + \begin{bmatrix} 0 & K_0 \\ 1/\rho_0 & 0 \end{bmatrix} \begin{bmatrix} \hat{p} \\ \hat{\mu} \end{bmatrix}_s = 0. \tag{4.1.8}$$

In this case, the equation depends on $\boldsymbol{\omega}$ through the variable μ. However, the equation itself is invariant over all $\boldsymbol{\omega}$, owing to the fact that the problem (4.1.7) is isotropic. Moreover, note that this is exactly the same equation obtained in the physical space if you consider the case of a plane wave where the data varies only in the direction ω so that derivatives in the orthogonal direction vanish.

The Radon transform therefore transforms n-dimensional hyperbolic problems such as (4.1.4) and (4.1.7) into their 1-dimensional counterparts (4.1.5) and (4.1.8), respectively.

4.1.2 Multi-dimensional extension of large time-step (LTS) methods

Previous dimensional splitting methods [64, 74, 104] such as Strang splitting do not allow a natural extension of large time-step (LTS) methods [60, 61, 62] to multiple spatial dimensions. In order to take large time-steps for constant coefficient multi-dimensional hyperbolic problems, one can use the Fourier transform, for example. Upon taking the Fourier transform, one is left with a set of ordinary differential equations (ODEs) different from the original problem [17, 108]. On the other hand, using the Radon transform, one obtains a dimensional splitting that reduces the multi-dimensional problem into a family of one-dimensional counterparts of similar (if not identical) form. This allows 1D LTS methods to be applied for each of these problems, and the multi-dimensional solution is obtained by computing the inverse of the Radon transform. Moreover, the Radon transform provides an intuitive geometrical interpretation as a decomposition into planar waves and thus yields other useful applications. These applications will be illustrated in Section 4.2.

This multi-dimensional extension of the LTS method for the constant coefficient case is very straightfoward. Taking the Radon tranform of the problem as above, one obtains a set of 1D problems such as (4.1.5) or (4.1.8). Then one applies the 1D LTS solution operator $\mathcal{K}$ to evolve the initial data $\hat{u}_0(\boldsymbol{\omega}, s)$ for each $\boldsymbol{\omega}$ up to desired final time T. The operator $\mathcal{K}$ may depend on the direction $\boldsymbol{\omega}$, so we denote the dependence as a parameter by writing $\mathcal{K} = \mathcal{K}(T; \boldsymbol{\omega})$. This yields the Radon transform of the solution at time T,

$$\hat{q}(T, \boldsymbol{\omega}, s) = \mathcal{K}(T; \boldsymbol{\omega}) \hat{q}_0(\boldsymbol{\omega}, s). \tag{4.1.9}$$

Then, to compute the solution q we can apply the inversion formula

$$c_n q(T, \mathbf{x}) = \begin{cases} \mathcal{R}^{\#} \dfrac{d^{n-1}}{ds^{n-1}} \hat{q}(T, \boldsymbol{\omega}, s) & \text{if } n \text{ is odd}, \\ \mathcal{R}^{\#} H_s \dfrac{d^{n-1}}{ds^{n-1}} \hat{q}(T, \boldsymbol{\omega}, s) & \text{if } n \text{ even}. \end{cases} \tag{4.1.10}$$

where the constant $c_n = (4\pi)^{(n-1)/2}\Gamma(n/2)/\Gamma(1/2)$ and H_s denotes the Hilbert transform. Much is known about the inversion; see standard texts such as [51, 80] for more details.

This splitting can also be related to the Strang splitting, if one views it as a decomposition of the multi-dimensional problem into planar wave propagation. In Strang splitting one constructs the planar waves emanating in varying directions by dividing a single time-step into multiple successive planar wave propagations. The Radon transform decomposes the multi-dimensional directions by explicitly discretizing the sphere S^{n-1}.

Let us consider a concrete example, the 2D acoustic equation (4.1.7). Let us set $K_0 = \rho_0 = 1$, so that we have the sound speed $c = 1$, and impose the initial conditions

$$q_0(x, y) = \begin{bmatrix} p_0(x, y) \\ 0 \\ 0 \end{bmatrix}, \quad p_0(x, y) = \begin{cases} \cos(\pi(x^2 + y^2)/2) & \text{if } x^2 + y^2 < 1, \\ 0 & \text{otherwise.} \end{cases} \tag{4.1.11}$$

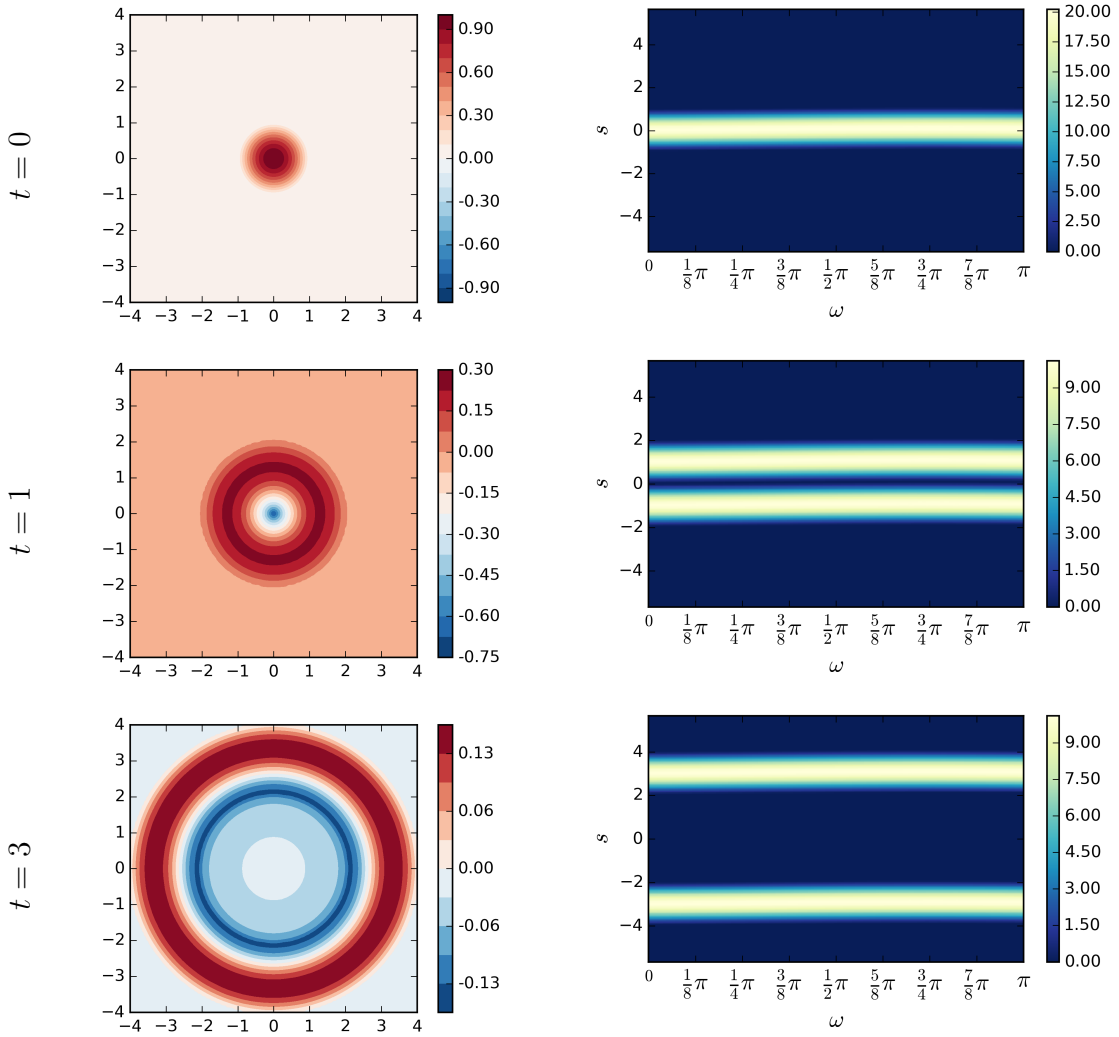

Figure 4.1: The solution to the acoustic equation using the Radon transform in the square domain $[-4, 4] \times [-4, 4]$. The pressure p is shown on the left column and its Radon transform $\hat{p}$ is shown on the right column, at times $t = 0$ (first row), $t = 1$ (second row), and $t = 3$ (third row).

The initial pressure profile is a cosine hump supported in a disk of radius 1 centered at the origin, and the initial velocity profile is identically zero. We will also set absorbing boundary conditions in the manner to be described in Section 4.2.1.

On the transformed side (4.1.8), the evolution for any fixed direction $\boldsymbol{\omega} \in S^1$ is given by the d'Alembert solution (4.1.12),

$$\hat{q}(t, \boldsymbol{\omega}, s) = \frac{1}{2} \left(\mathbf{r}_1 \hat{q}_0(\boldsymbol{\omega}, s - t) + \mathbf{r}_2 \hat{q}_0(\boldsymbol{\omega}, s + t) \right) \tag{4.1.12}$$

where

$$\mathbf{r}_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad \text{and} \quad \mathbf{r}_2 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}. \tag{4.1.13}$$

This reduces to simple shifts at corresponding speeds, which can be computed easily up to any time t. This is precisely the 1D LTS solution for the constant coefficient case.

The solution to the acoustic equation computed on the domain $[-4, 4] \times [-4, 4]$ is shown in the left column of Figure 4.1. The Radon transform of the pressure term $\hat{p}$ is plotted in the right column of the same figure. Note that this problem is radially symmetric about the origin. A consequence of this is that the Radon transform is invariant with respect to the variable $\boldsymbol{\omega}$, hence the Radon transform of the solutions at different times all appear as horizontal stripes. (There is a small amount of shift, following from the fact that for an image of even size N, the origin is chosen as the $(N/2, N/2)$-pixel, slightly off center.)

A key observation is that the evolution of the solution in the transformed variables is a sum of two shifting horizontal stripes, although the wave profile in the spatial domain propagates radially. For each fixed angle $\boldsymbol{\omega}$, one only need solve the d'Alembert solution (4.1.12), which is easy to solve to any time t by shifting the initial profile twice each according to two opposite speeds, and summing them. Intuitively, the shifts correspond to the propagation of decomposed planar waves for any fixed normal directions in S^1.

The actual computational did not make use of the continuous Radon transform (4.1.1), but rather a completely discrete approximation called the discrete Radon transform (DRT),

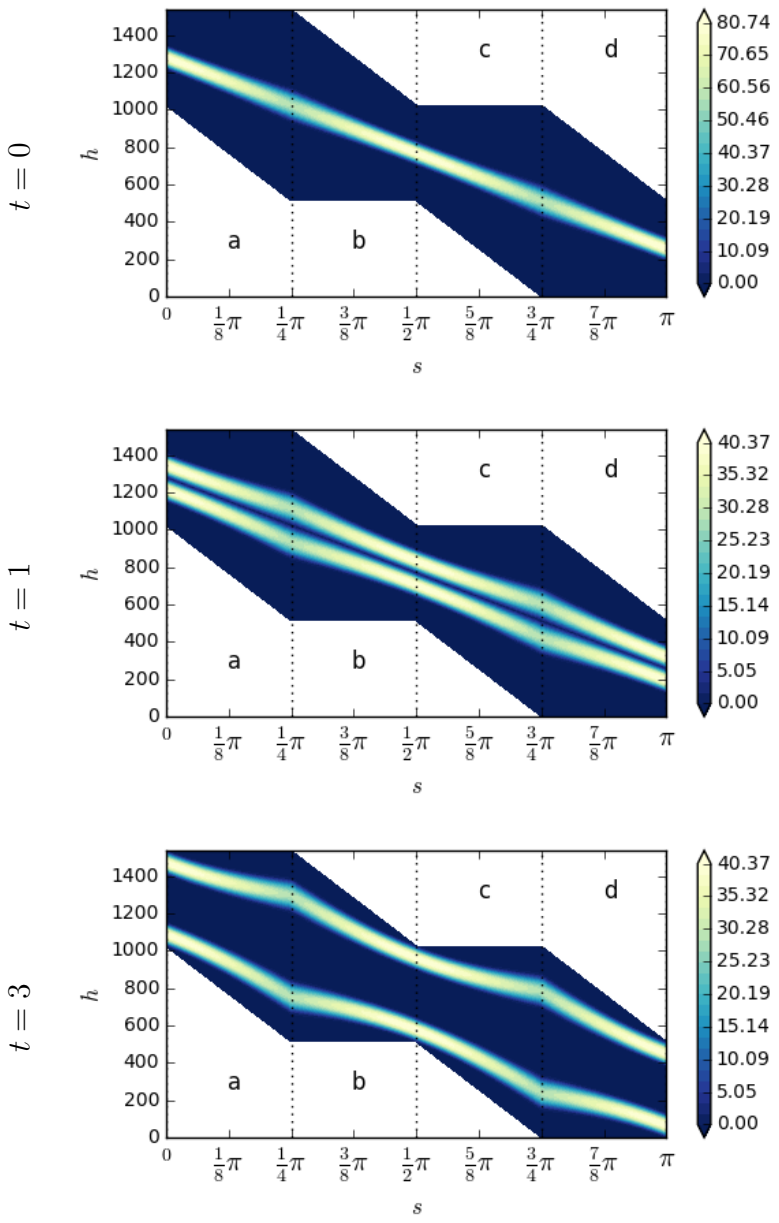

Figure 4.2: The discrete Radon transform (DRT) of the solution to the acoustic equation (4.1.7), for times $t = 0, 1$ and 3. The parameters h and s which appear on the axes designate *d-lines* (see Figure 4.11) and indices $\{a, b, c, d\}$ denote quadrants (see Figure 4.12). Details appear in Section 4.3.

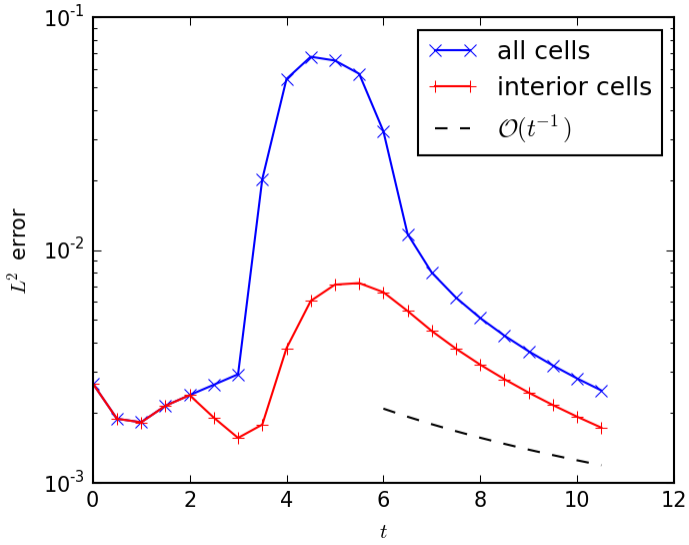

Figure 4.3: L^2 difference between the splitting solution and the reference solution over time, for the acoustic equation (4.1.7) with the initial condition (4.1.11). The difference over all cells are shown together with the difference over interior cells inside $[-3,3] \times [-3,3]$. The slope of $\mathcal{O}(1/t)$ is also shown for comparison.

which is plotted in Figure 4.2. The DRT will be introduced and discussed in further detail in Section 4.3. Here it will suffice to mention that the continuous transform can be obtained by an easy change of variables (4.3.8-4.3.11) which scale the domain and amplitude of the DRT, and that the change of variables do not affect the intertwining property. The 1D LTS method can still be used on the DRT, just as in the case of the continuous transform.

A grid of size 128×128 was used and prologation of $p = 2$ (see Section 4.3.3) was used for the inversion of DRT. This solution is compared with a reference finite volume solution computed on a larger domain $[-8,8]\times[-8,8]$, using the wave propagation algorithm [63] with Lax-Wendroff flux and Van Leer limiter [64, 111], implemented in the CLAWPACK software package [25]. The reference solution was computed on a 1024×1024 finer grid-cells of uniform size, then corresponding cells have been summed and compared with coarser cells of the DRT solution. The ℓ^2-norm of the difference over time is displayed below in Figure 4.3, where

the error is in the order of 10^{-3} up to time $t = 2.5$, before the profile starts approaching the boundary. The subsequent increase in the error is expected from the particular absorbing boundary set up here, as will be discussed in Section 4.2.1.

While this problem was radially symmetric, the splitting is by no means restricted to problems with radial symmetry. Let us modify the initial conditions above so that it is the sum of two cosine humps of different radii and heights,

$$\begin{aligned} q_0(x,y) &= \begin{bmatrix} p_1(x,y) \\ 0 \\ 0 \end{bmatrix}, \\ p_1(x,y) &= p_0(x+1, y+1.5) + 1.5\, p_0(1.25(x-0.75), 1.25(x-1.1)). \end{aligned} \tag{4.1.14}$$

The splitting solution and its continuous Radon transform is plotted in Figure 4.4. In the first row of the figure, the initial condition and its Radon transform are shown. The two cosine humps in the initial condition each correspond to a sinusoidal signal on the transformed side. Recall the horizontal line centered at $s = 0$ from the previous example (Figure 4.1). The sinusoidal shift away from $s = 0$ is due to the fact that translation is an anisotropic operation. This can also be deduced from the transformed transport equation (4.1.5) in which the transport speed is $\boldsymbol{\theta} \cdot \boldsymbol{\omega}$, that is, $\cos\phi$ where ϕ is the angle between transport direction $\boldsymbol{\theta}$ and the direction of the transform $\boldsymbol{\omega}$. For example, when the cosine hump at the origin p_0 (4.1.11) is transported away from the origin by $r\boldsymbol{\theta}$, $\hat{p_0}$ is shifted by $\hat{p_0}(\boldsymbol{\omega}, s - r\cos(\phi))$.

In any case, the solution is still given by the d'Alembert solution (4.1.12) and the acoustics equation can be solved exactly same way as before. The DRT used in the actual computations are plotted in Figure 4.5. Each corresponds to a continuous transform in Figure 4.4.

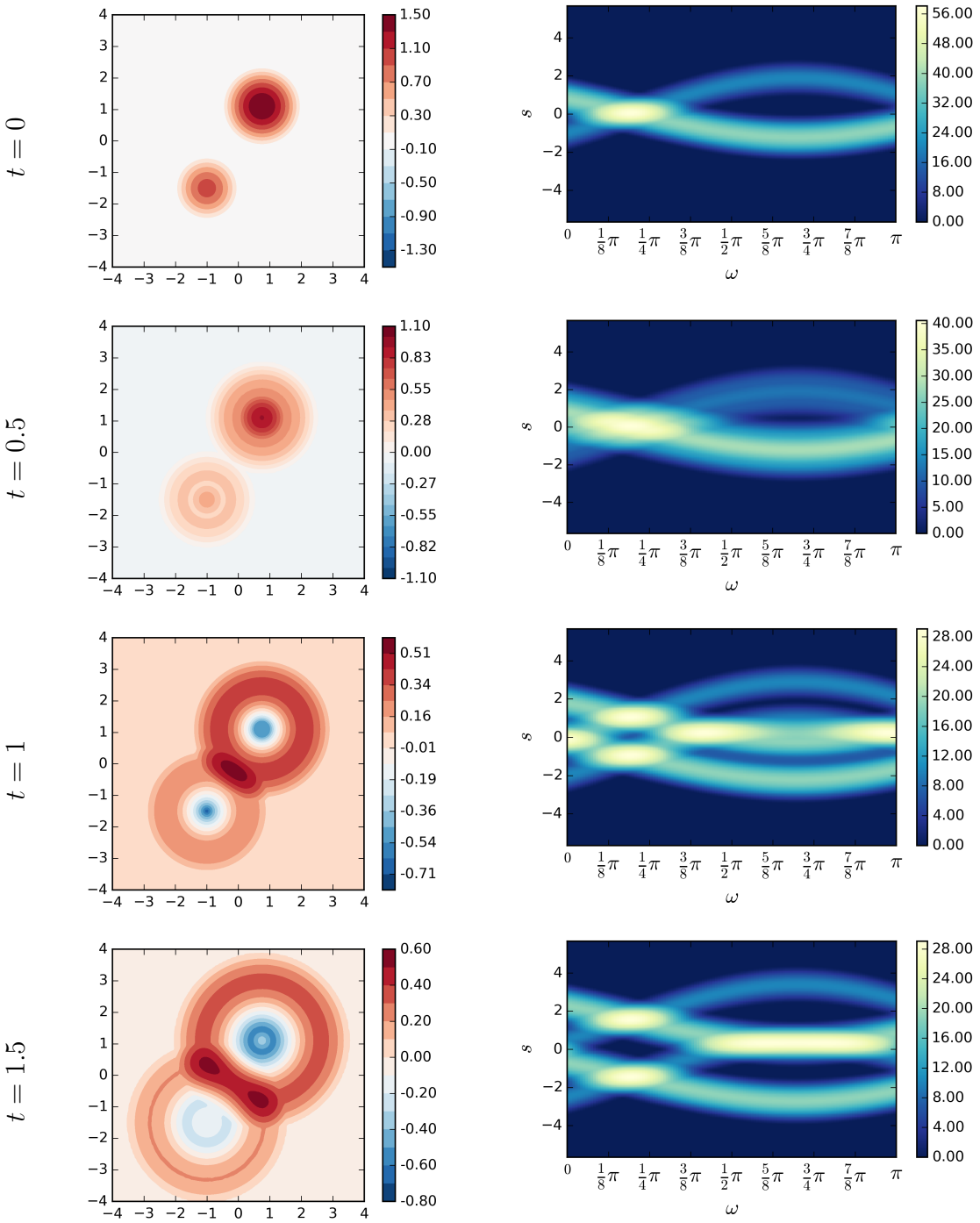

Figure 4.4: The solution to the acoustic equation using the Radon transform in the square domain $[-4, 4] \times [-4, 4]$. The pressure p is shown on the left column and its Radon transform $\hat{p}$ is shown on the right column, at times $t = 0$ (first row), $t = 0.5$ (second row), $t = 1$ (third row), and $t = 1.5$ (fourth row).

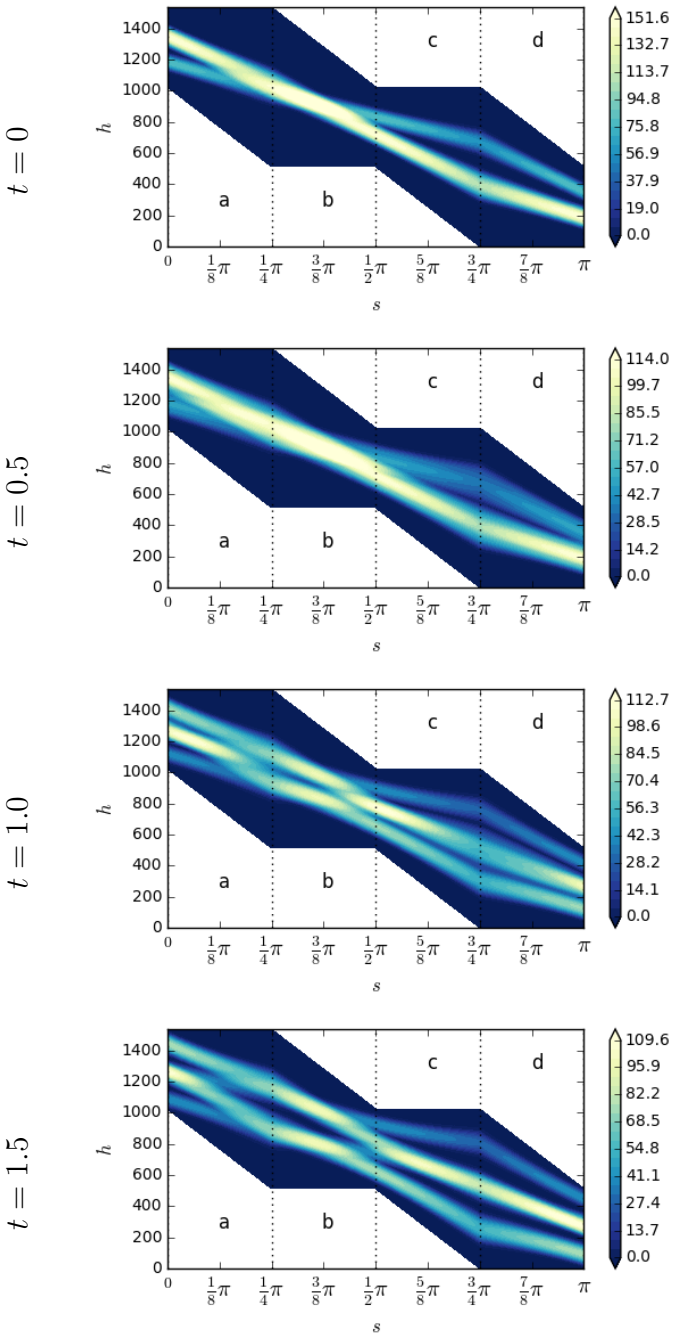

Figure 4.5: The discrete Radon transform (DRT) of the solution to the acoustic equation (4.1.7) with initial conditions (4.1.14), for times $t = 0, 0.5, 1$ and 1.5. The parameters h and s which appear on the axes designate *d-lines* (see Figure 4.11) and indices $\{a, b, c, d\}$ denote quadrants (see Figure 4.12). Details appear in Section 4.3.

4.1.3 Splitting for the nonlinear case

Here we will discuss how the splitting above to can be applied to a fully nonlinear system of hyperbolic equations. For a state vector q, such a PDE is given in the form

$$q_t + f(q)_x + g(q)_y = 0, \tag{4.1.15}$$

where f and g are flux functions that can be nonlinear. Taking the Radon transform as before, we obtain

$$\hat{q}_t + (\omega_1 f(q) + \omega_2 g(q))^\wedge_s = 0. \tag{4.1.16}$$

Let us define the *directional flux function* as

$$h(q) = \omega_1 f(q) + \omega_2 g(q). \tag{4.1.17}$$

Then one obtains the nonlinear 1D equations,

$$\hat{q}_t + h(q)^\wedge_s = 0. \tag{4.1.18}$$

As in the acoustics equation (4.1.8), the dependence on $\boldsymbol{\omega}$ enters through the flux function $h(q)$, while the form of the equation is invariant with respect to $\boldsymbol{\omega}$.

As an example, consider the shallow water equations in 2D, in which $\rho, u, v : \mathbb{R}^+ \times \mathbb{R}^2 \to \mathbb{R}$ denote water height, velocity in the x-direction and velocity in the y-direction, respectively,

$$\begin{bmatrix} \rho \\ \rho u \\ \rho v \end{bmatrix}_t + \begin{bmatrix} \rho u \\ \rho u^2 + \frac{1}{2}\bar{g}\rho^2 \\ \rho uv \end{bmatrix}_x + \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + \frac{1}{2}\bar{g}\rho^2 \end{bmatrix}_y = 0. \tag{4.1.19}$$

Here $\bar{g}$ denotes the gravitational constant. The Radon transform as above yields 1D equation

in the form (4.1.18), in the transformed velocity variables $\mu = \omega_1 u + \omega_2 v$ and $\nu = -\omega_2 u + \omega_1 v$,

$$\begin{bmatrix} \rho \\ \rho\mu \\ \rho\nu \end{bmatrix}_t^{\wedge} + \begin{bmatrix} \rho u \\ \rho\mu^2 + \frac{1}{2}\bar{g}\rho^2 \\ \rho\mu\nu \end{bmatrix}_s^{\wedge} = 0. \tag{4.1.20}$$

Note that the first two equations of (4.1.20) are just the shallow water equation in a single dimension in the normal direction of the hyperplane, whereas the third equation is the conservation of momentum in the transverse direction.

We observe that the transformed equations resemble a finite volume discretization. Let us say that $\xi_{i,j}$ is a discretization of the hyperplane $\{\mathbf{x} \in \mathbb{R}^n : \langle \mathbf{x}, \boldsymbol{\omega}_i \rangle = s_j\}$. The specific discretization for the hyperplanes can take on many different forms, but here we will leave it in a general form. We denote the approximation to $\hat{q}(t_n, \boldsymbol{\omega}_i, s_j)$ at time-step t_n by $\hat{Q}^n_{i,j}$,

$$\hat{Q}^n_{i,j} \approx \int_{\xi_{i,j}} q(t_n, \mathbf{x}) \, \mathrm{d}m(\mathbf{x}). \tag{4.1.21}$$

For each fixed direction $i = i_0$, the collection of hyperplanes $\{\xi_{i_0,j}\}$ form a partition of the domain. We can consider these hyperplanes to be finite volume cells. In the equation (4.1.18) the flux function $h(q)$ assigns the flux between $\xi_{i_0,j}$ and $\xi_{i_0,j+1}$. If the cell boundary between $\xi_{i_0,j}$ and $\xi_{i_0,j+1}$ is denoted by $\xi_{i_0,j+\frac{1}{2}}$, we define the numerical flux $F^n_{i_0,j+\frac{1}{2}}$ to be an approximation to the flux at $\xi_{i_0,j+\frac{1}{2}}$, valid from time-step t_n to t_{n+1}. Then we have the finite volume update

$$\hat{Q}^{n+1}_{i_0,j} = \hat{Q}^n_{i_0,j} - \Delta t (F^n_{i_0,j+\frac{1}{2}} - F^n_{i_0,j-\frac{1}{2}}). \tag{4.1.22}$$

Once these updates are made for all i, the updated $\hat{Q}^{n+1}_{i,j}$ are combined through the inversion formula (4.1.10) to yield the numerical solution at time t_{n+1}.

The dimensional splitting strategy would be to compute the numerical flux $F^n_{i,j+\frac{1}{2}}$ by solving only the 1D Riemann problems in the x and y directions. Since the flux function $h(q)$ is a linear combination of normal fluxes $f(q)$ and $g(q)$ (4.1.17), we can compute h once

we have the approximation for these normal fluxes. In other words, we can solve the 1D Riemann problems for piecewise constant jumps locally in x and y directions, then sum these fluxes across the cell boundary $\xi_{i,j+\frac{1}{2}}$ to obtain the flux between hyperplanes. One thereby decomposes the multi-dimensional Riemann problem into a set of single-dimensional ones, to be combined together by the inversion formula (4.1.10).

Unlike in the constant coefficient case, the flux function $h(q)$ must be updated at every time step, as is usually done for finite volume methods, although one may apply the nonlinear LTS method on the transformed problem regardless. This would be based on the 1D analogues studied in [60, 61, 62]. The fully nonlinear splitting will not be implemented here, but will be investigated in future work.

4.2 Applications of the dimensional splitting

The dimensional splitting described in Section 4.1 above is a decomposition of hyperbolic solutions into evolution of planar waves. This decomposition can be useful in diverse settings. Here we discuss two applications: the absorbing boundary conditions and the displacement interpolation.

4.2.1 Absorbing boundary conditions

It is well-known that imposing absorbing boundary conditions to emulate infinite domains in multi-dimensional wave propagation is a challenging problem [11, 12, 22, 32]. On the other hand, the 1D extrapolation boundary condition is much more tractible [64]. A major advantage of this splitting method is that the 1D extrapolation boundary conditions can be used on the transformed side at the computational boundary to avoid any reflections. This yields *exactly* the desired absorbing boundary conditions in the odd-dimensional case. Therefore, the dimensional splitting introduced in the previous section can be used directly to impose absorbing boundary conditions in 3D. On the contrary, there is an error caused by such an extrapolation in the even-dimensional case. This is due to the Huygens' principle, evident in the presence of the Hilbert transform in the inversion formula (4.1.10). In this section,

we discuss the type of error caused by imposing such extrapolation boundary conditions via the Radon transform in even dimensions.

In the true infinite domain, the non-zero values in transformed variables beyond the computational boundary of $\mathcal{S}^{n-1} \times \mathbb{R}$ affect the solution within the computational domain in the original variables $\mathbb{R}^n$. For example, the vertical translation of horizontal strips in Figure 4.1 should continue beyond the finite computational boundary, and by imposing a 1D extrapolation boundary condtion we would be neglecting this infinite propagation. To make this more precise, denote the computational (finite) transformed domain by $\Omega = \{(\boldsymbol{\omega}, s) \in S^{n-1} \times (-b, b)\}$ for some $b > 0$. Let χ_Ω be the characteristic function of the finite and $\chi_{\mathbb{R}^n \setminus \Omega} = 1 - \chi_\Omega$. For n even, the exact solution q can be written as,

$$q(T, \mathbf{x}) = \frac{1}{c_n} \mathcal{R}^\# H_s \frac{d^{n-1}}{ds^{n-1}} \hat{q}(T, \boldsymbol{\omega}, s) \tag{4.2.1}$$

$$= \frac{1}{c_n} \mathcal{R}^\# H_s \chi_\Omega \frac{d^{n-1}}{ds^{n-1}} \hat{q}(T, \boldsymbol{\omega}, s) + \frac{1}{c_n} \mathcal{R}^\# H_s \chi_{\mathbb{R}^n \setminus \Omega} \frac{d^{n-1}}{ds^{n-1}} \hat{q}(T, \boldsymbol{\omega}, s). \tag{4.2.2}$$

Recall that $\mathcal{R}^\#$ is the back-projection (4.1.2). The first term in (4.2.2) is the approximate solution one would obtain if extrapolation boundary is set up at the boundaries $s = \pm b$. Let us call this approximate solution $q_h(\mathbf{x})$. Then the error is

$$q(T, \mathbf{x}) - q_h(T, \mathbf{x}) = \frac{1}{c_n} \mathcal{R}^\# \left(\text{p.v} \int_{(-\infty, -b)] \cup [b, \infty)} \frac{1}{r - s} \frac{\partial^{n-1}}{\partial r^{n-1}} \hat{q}(T, \boldsymbol{\omega}, r) \mathrm{d}r \right), \tag{4.2.3}$$

where p.v denotes the principal value integral. Note that in hyperbolic problems in free space, wave profiles will radiate outwards, that is, the support of $\hat{q}$ will be transported towards $r = \pm\infty$. This causes the RHS above to decay with time. Furthermore, the principal integral is a smooth function of s as long as $\partial^{n-1}\hat{q}/\partial r^{n-1}$ is integrable. Since $\mathcal{R}^\#$ is an integral, we expect the error to be smoother than $q(\mathbf{x})$.

Let us revisit the acoustic equations example (4.1.7) from Section 4.1.2, with initial conditions (4.1.11). Since $\hat{q}_0(x, y)$ is supported in $\{(\boldsymbol{\omega}, s) \in S^{n-1} \times \mathbb{R} : |s| \leq 1\}$, we have a

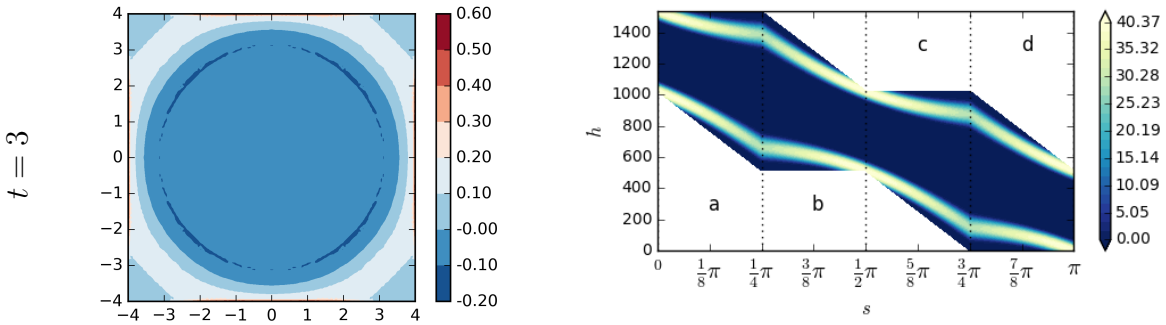

Figure 4.6: The splitting solution to the acoustic equation (4.1.7) in the square domain $[-4,4]\times[-4,4]$. The pressure p and its DRT $\mathcal{R}_N p$ at time $t=3$.

simple estimate for the case when t is sufficiently large so that $(-1+t,1+t)\subset(b,\infty)$,

$$
\begin{aligned}
|q(\mathbf{x})-q_h(\mathbf{x})| &= \frac{1}{2c_n}\mathcal{R}^{\#}\left|\int_{-\infty}^{-b}\frac{\mathbf{r}_1}{r-s}\frac{d}{dr}\,\hat{q}_0(r+t)\mathrm{d}r\right. \\
&\qquad\left.+\int_{b}^{\infty}\frac{\mathbf{r}_2}{r-s}\frac{d}{dr}\,\hat{q}_0(r-t)\mathrm{d}r\right| \qquad (4.2.4)\\
&\le \frac{2\,|S^{n-1}|}{c_n}\frac{\sup\|p_0'\|_1}{|t+1|},
\end{aligned}
$$

where $|S^{n-1}|$ is the surface area of an n-sphere. Therefore we see that the effect of the extrapolation boundary decays relatively slowly, at the rate of $\mathcal{O}(1/t)$.

These observations help us understand the L^2 error plot in Figure 4.3. Recall that the reference solution was computed in a larger domain of size $[-8,8]\times[-8,8]$ instead of imposing absorbing boundary conditions (see description in Section 4.1.2). The error from the truncation (4.2.4) begins to appear around time $t=2.5$ and peaks around time $t=5$, then decays to zero with time. The solution at time $t=3$, as it has begun to interact with the boundary, is shown in Figure 4.6. Note that there are no reflections from this boundary condition. On the other hand, a thin layer appears at the computational boundary. The layer is clearly non-physical, but is localized and has limited affect on the solution further in the interior. The DRT of the solution is also shown to its right, and we can see that the two

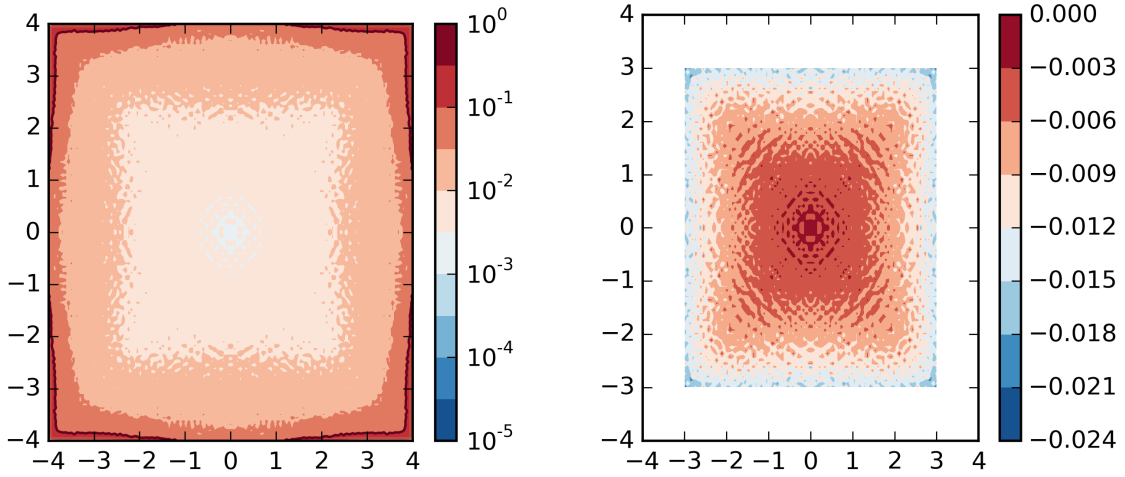

Figure 4.7: Difference to the reference solution at time $t = 5$. The difference for all cells in log-scale (left) and the difference for the interior cells in $[-3, 3] \times [-3, 3]$ (right).

pulses from the d'Alembert solution are hitting the 1D extrapolation boundaries (the top and bottom boundaries of the polygonal region in dark blue). The pulses first arrive at the DRT boundary at the angles $0, \pi/2$ and π, and this agrees with the solution plot to the left.

Comparing this solution to the reference solution, one discovers that the bulk of this error is concentrated near the thin layer which appears near the boundary. In Figure 4.7, we have plotted the difference between our solution and the reference solution on the computational domain $[-4, 4] \times [-4, 4]$ to the left. When we restrict the contour plot to the interior portion of the domain $[-3, 3] \times [-3, 3]$ as is shown to the right, we see that the error is significantly smaller as we move away from the boundary. This is to be expected from the decay of the principal value integral in (4.2.3): the further away the interior point is from the boundary, the smaller the effect of the trunctation by χ_Ω. If we denote the the distance by d, then the decay will be $\mathcal{O}(1/(d + t))$. As t increases and the waves leave the domain, the error also decays, at a slightly faster rate than the rate $\mathcal{O}(1/t)$ estimated by (4.2.4).

4.2.2 Displacement interpolation

In projection-based model reduction, the solution to a parametrized PDE is projected into a low-dimensional subspace, yielding a fast solver with significantly lower computational cost without compromising accuracy. To discover this low-dimensional subspace, the popular approach is to use proper orthogonal decomposition (POD) [15]. For hyperbolic PDEs, however, the solutions do not lie in a low-rank linear subspace, even for the simplest problems [1, 21, 95, 96, 98]. For instance, the d'Alembert solution (4.1.12) is a linearly independent function of s for each $t > 0$. It is easy to see that a linear projection of this solution to a low-dimensional basis would not yield a good approximation of the solution. Naturally, methods to remove translational symmetry [95, 96, 98] are being actively explored. This is also intimately related to the concept of *displacement interpolation*, a term we borrow from the optimal transport literature [113], in which one aims to minimize the Wasserstein distance, although we will not make the connection more explicit here.

Let us first illustrate how displacement interpolation arises naturally, with a simple 1D example. Suppose ϕ_0 is a hat function, given by

$$\phi_0(x) = \begin{cases} \frac{x}{h} + 1 & \text{if } -h < x < 0, \\ -\frac{x}{h} + 1 & \text{if } 0 \leq x < h, \\ 0 & \text{otherwise,} \end{cases} \tag{4.2.5}$$

for some $h > 0$. Let ϕ_1 and ϕ_2 be translation and scaling of ϕ_0,

$$\phi_1(x) = \phi_0(x) \quad \text{and} \quad \phi_2(x) = \frac{1}{4}\phi_0(x-2). \tag{4.2.6}$$

For $h = 0.1$, the two functions are shown in the first row of Figure 4.8. The linear interpo-

lation ψ of the two functions with weights $(1-\tau)$ and τ is given by

$$\begin{aligned}\psi(x) &= (1-\tau)\phi_1(x) + \tau\phi_2(x) \\ &= (1-\tau)\phi_0(x) + \frac{\tau}{4}\phi_0(x-2)\end{aligned} \tag{4.2.7}$$

whereas a displacement interpolation between the two functions under a simple transport map (1D translation) would be given by

$$\begin{aligned}\psi_D(x) &= (1-\tau)\phi_1(x-2\tau) + \frac{\tau}{4}\phi_2(x+2(1-\tau)) \\ &= \left(1-\frac{3}{4}\tau\right)\phi_0(x-2\tau)\end{aligned} \tag{4.2.8}$$

The two interpolants for $\tau = 0.25$ are plotted in the bottom row in Figure 4.8 . Since ϕ_1 and ϕ_2 are both translates of a scalar multiple of ϕ_0, the displacement interpolation reveals the low-rank nature of the two functions, whereas the linear interpolant remains rank two for $\tau \in (0,1)$.

In practice, one must be able to deduce that ϕ_1 and ϕ_2 above lie in the translates of $\text{span}\{\phi_0\}$ without a priori knowledge. To achieve this, one may apply the *template-fitting* procedure [98] which solves the minimization problem

$$\tau_* = \text{argmin}_{\tau\in\mathbb{R}} \left\|\phi_2(x) - \mathcal{K}(\tau)[\phi_1(x)]\right\|_2, \tag{4.2.9}$$

where $\mathcal{K}$ is the translation operator, $\mathcal{K}(\tau)[\phi_1(x)] = \phi_1(x-\tau)$, then perform a singular value decomposition (SVD) on $\{\phi_2, \mathcal{K}(\tau_*)\phi_1\}$ [95, 98]. However, this simple formulation does not take into account multiple traveling speeds or heavily deforming profiles, which limits its applicability. *Transport reversal* was introduced in [96] to overcome these limitations. The algorithm is a greedy iteration over a generalized form of (4.2.9), which decomposes the 1D function $\phi_2(x)$ into multiple traveling structures. To be more precise, given two functions ϕ_1

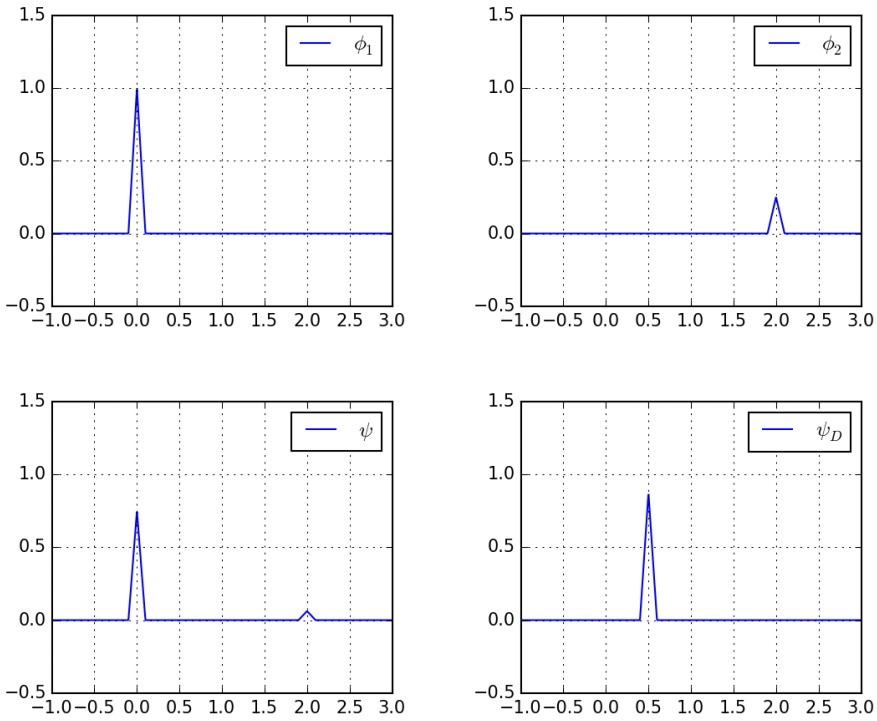

Figure 4.8: Two hat functions ϕ_1 and ϕ_2 (top row) and linear interpolation ψ and displacement interpolation ψ_D between the two functions with respective weights 0.75 and 0.25 (bottom row).

and ϕ_2 as in (4.2.9), transport reversal yields the decomposition

$$\psi_D(x;\tau) = \sum_{k=1}^{K} \eta_j(\tau)\mathcal{K}(\nu_j\tau)[\rho_j(x,\tau)\varphi_1(x)]. \tag{4.2.10}$$

where η_j is a scaling function and ρ_j a cut-off function. For more detailed treatment of this decomposition in 1D, we refer the reader to [96]. Let us suppose we have computed this decomposition. The displacement interpolant ψ_D resulting from this decomposition is set to satisfy,

$$\psi_D(x,0) = \phi_1(x) \quad \text{and} \quad \psi_D(x,1) = \phi_2(x). \tag{4.2.11}$$

As an example, let us assume we are given two functions ϕ_1 and ϕ_2 as shown in Figure 4.9(a) and 4.9(b). These are taken from the 1D slice located at $s = \tan(\frac{5}{8}\pi)(N-1)$ of the DRT from Figure 4.5. The transport reversal would decomposes ϕ_2 into a superposition of two traveling profiles,

$$\eta_1(\tau)\mathcal{K}(\tau)[\rho_1(x)\varphi_1(x)] \quad \text{and} \quad \eta_2(\tau)\mathcal{K}(-\tau)[\rho_2(x)\varphi_1(x)], \tag{4.2.12}$$

each plotted in Figure 4.10. In exact arithmetic, the two iterations of transport reversal would pick-up exactly the d'Alembert solution (although in practice numerical error would require further iterations to pick off the residuals). That is, we would obtain $h_1 = h_2 = 1/2$ and $\rho_1 = \rho_2 = 1$ with $\nu_1 = -\nu_2 = 1$ and $K = 2$ in (4.2.10). Now, the displacement interpolation for $\tau = 1/2$ can be computed, yielding ψ_D shown in Figure 4.9(c). The exact evolution of the two iterates (4.2.12) are shown in Figure 4.10.

Now, this displacement interpolation was done for a single *slice* of the fixed $\boldsymbol{\omega}$ in the transformed variables. Suppose we are given a function φ in 2D. Then by performing the same transport reversal on its Radon transform $\hat{\varphi}$ for all $\boldsymbol{\omega}$ as functions of the variable s, we obtain an extension of the 1D displacement interpolant (4.2.8) to higher spatial dimensions.

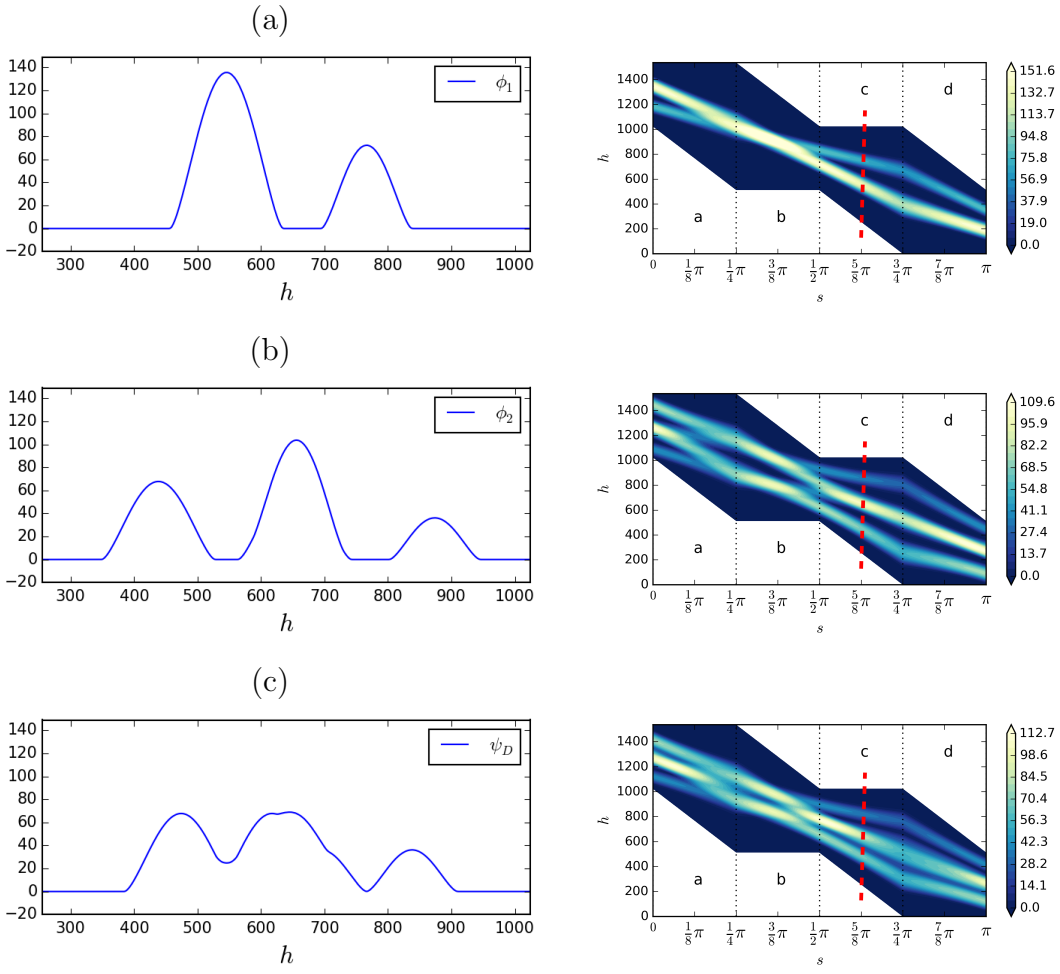

Figure 4.9: Two 1D functions ϕ_1, ϕ_2 and the displacement interpolation ψ_D are shown in the first column. These are exactly the $s = \tan(\frac{5}{8}\pi)(N-1)$ slice of the DRT of the acoustic equation example in Figure 4.5. The slice is indicated by the dashed red vertical line in the plots in the right column.

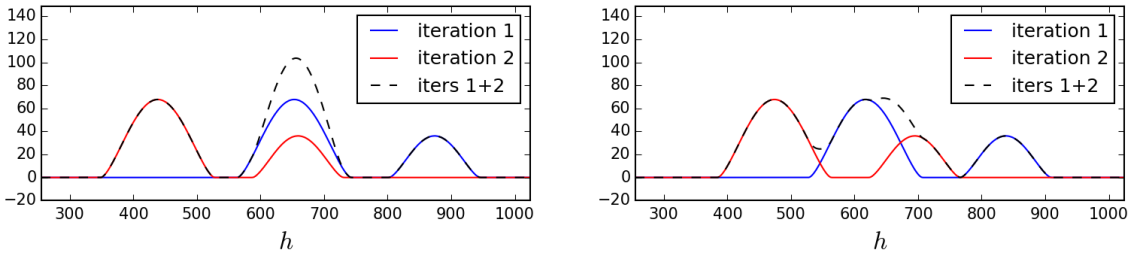

Figure 4.10: The first two contributions (4.2.12) of the transport reversal for ϕ_1 and ϕ_2 shown in Figure 4.9 (left) and the displacement interpolation resulting in $\psi_D(x; 0.5)$ (4.2.10) (right). ψ_D shown in dotted line is also displayed in the bottom of Figure 4.9.

For each fixed angle $\boldsymbol{\omega}_0 \in S^1$ we obtain the transport reversal that decomposes the traveling structures,

$$\hat{\psi}_D(\boldsymbol{\omega}_0, s; \tau) = \sum_{k=1}^{K} \eta_{i,k}(\tau)\mathcal{K}(\nu_{i,k}\tau)[\rho_{i,k}(s, \tau)\hat{\varphi}(\boldsymbol{\omega}_0, s)]. \tag{4.2.13}$$

These can be used for displacement interpolation as above for each $\boldsymbol{\omega}$. The inverse transform can be taken to obtain the displacement interpolant ψ_D. For given a given value τ, we have

$$\psi_D(x, y; \tau) = \mathcal{R}^{-1}\left(\sum_{k=1}^{K} \eta_{i,k}(\tau)\mathcal{K}(\nu_{i,k}\tau)[\rho_{i,k}(s, \tau)\hat{\varphi}(\boldsymbol{\omega}_i, s)]\right). \tag{4.2.14}$$

Let us clarify the implication: for the acoustic equation example with the initial condition (4.1.14), we were given a snapshot of the solution q at time $t_1 = 0.5$ and $t_2 = 1.5$. From the two snapshots, we were be able to accurately approximate the solution for all time, without additional information about the dynamics, without even knowing the PDE. Thus this interpolant can be more useful than linear interpolation: the linear subspace spanned by $\{q(t_1, \mathbf{x}), q(t_2, \mathbf{x})\}$ does not a contain a good approximation for representing the evolving solution.

This ability to compute the displacement interpolation by exploiting the simple dynamic on the transformed side will be useful in the future development of transport reversal as a model reduction tool in multi-dimensional settings.

4.3 Discrete Radon transform (DRT)

There are many different discretizations of the Radon transform and its inverse [8, 16, 19, 56, 73], arguably the most well-known being the filtered backprojection (FBP) algorithm [80]. However, its reliance on Fourier transforms and spherical harmonics lead to some filtering of high-frequency content, causing Gibbs phenomenon near the sharp edges in the solution. This is not suitable for use in hyperbolic PDEs, which are known to develop shocks discontinuities.

Instead, we consider the use of a completely discrete analogue, namely the approximate discrete Radon transform (ADRT), which we refer to simply as the discrete Radon transform (DRT), introduced in [18, 45]. Rather than interpolating pixel values onto straight lines passing near it, DRT sums one entry for each row or column, along so-called *digital lines* or *d-lines*. The *d-lines* are defined recursively, allowing for a fast computation in $\mathcal{O}(N^2 \log N)$ for an image of size $N \times N$. The back-projection is given by reversing the recursion, and is also fast with the same computational cost of $\mathcal{O}(N^2 \log N)$. The precise definitions are given below.

4.3.1 Recursive definition of DRT

The *d-lines* of length N are denoted by $D_N(h, s)$ with two parameters h and s (see Figure 4.11.) h denotes the *height* (x-intercept) and s the *slope* (x-displacement), and the pair corresponds to s and $\boldsymbol{\omega}$ for the continuous transform (4.1.3), respectively. Although the same notation s is used here again after having been used in the continuous setting (4.1.1) we will keep the notation in order to follow the intuitive notation of [88], and mark the continuous variable with a subscript s_c whenever the two are used simultaneously. The definition uses the recursion in which d-lines of length $2n$ are split into left and right d-lines of half its length.

$$\begin{aligned} D_{2n}(h, 2s) &= D_n^L(h, s) \cup D_n^R(h + s, s), \\ D_{2n}(h, 2s + 1) &= D_n^L(h, s) \cup D_n^R(h + s + 1, s). \end{aligned} \tag{4.3.1}$$

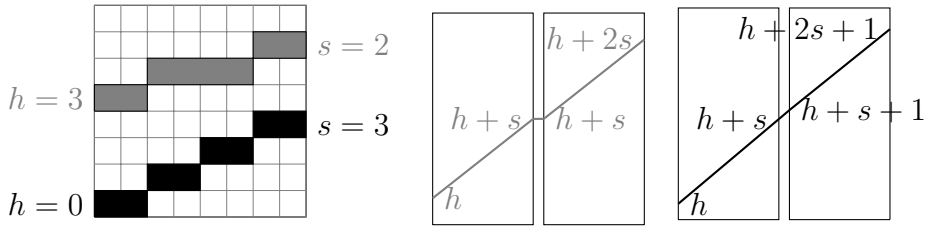

Figure 4.11: Examples of digital lines (d-lines) determined by two parameters h and s (left) and the diagram of the recursion relation (4.3.1) (right). In both figures the case when s is even is in gray, and the case s is odd is in black.

The recursion (4.3.1) defines only a quarter of the possible d-lines, as the slope s will range from 0 to N, corresponding to angles 0 to $\pi/4$ starting from the x-axis in the counter-clockwise direction. This is referred to by saying that the d-lines cover one quadrant, for the full transform one needs to cover the angles from 0 to π. The other d-lines can be computed by transposing and flipping the indices h and s. We will denote the d-lines and DRT corresponding to the angular intervals $[0, \pi/4]$,$[\pi/4, \pi/2]$,$[\pi/2, 3\pi/4]$, and $[3\pi/4, \pi]$ by a,b,c, and d.

The *DRT of an array* $A \in \mathbb{R}^2$ *for the quadrants* a, b, c *and* d are given by the summation of entries of A over the d-lines,

$$(\mathcal{R}_N^a A)_{h,s} = \sum_{(i,j)\in D_N(h,s)} A_{i,j}, \tag{4.3.2}$$

$$\left(\mathcal{R}_N^b A\right)_{h,s} = \sum_{(i,j)\in D_N(h,s)} A_{j,i}, \tag{4.3.3}$$

$$(\mathcal{R}_N^c A)_{h,s} = \sum_{(i,j)\in D_N(h,s)} A_{j,N-i+1}, \tag{4.3.4}$$

$$\left(\mathcal{R}_N^d A\right)_{h,s} = \sum_{(i,j)\in D_N(h,s)} A_{N-i+1,j}. \tag{4.3.5}$$

The full DRT is simply the ordered tuple of all quadrants, and we write

$$\mathcal{R}_N A = (\mathcal{R}_N^a A, \mathcal{R}_N^b A, \mathcal{R}_N^c A, \mathcal{R}_N^d A). \tag{4.3.6}$$

Due to the recursive form of (4.3.1), the transform can be computed in $\mathcal{O}(N^2 \log N)$. The parameters h and s belong to the range $[-s+1, N]$ and $[0, N]$ so $R_N^a A \in \mathbb{R}^{\left(\frac{3}{2}N+\frac{1}{2}\times N\right)}$. Therefore, $\mathcal{R}_N : \mathbb{R}^{N\times N} \to \mathbb{R}^{(6N+2)\times N}$.

There is a simple relationship between the DRT and the continuous Radon transform. First let us say that $s_c \in [-1, 1]$ (perhaps through proper scaling) and parameterize $\boldsymbol{\omega}$ by $\boldsymbol{\omega} = (\cos\theta, \sin\theta)$ where $\theta \in [0, \pi]$. The relation to continuous variables $(s_c, \boldsymbol{\omega})$ is given by

$$s_c = \cos\theta \left(\frac{2h}{N} - 1 + \frac{s}{N-1}\right), \qquad \theta = \arctan\left(\frac{s}{N-1}\right). \tag{4.3.7}$$

Then the explicit relation between the DRT $\mathcal{R}_N$ and the continuous transform $\mathcal{R}$ are given after the density of the lines are also transformed depending on the angle by $\cos\theta$,

$$\mathcal{R}_N^a f(h,s) = \cos\theta\, \mathcal{R}(s_c, \theta), \tag{4.3.8}$$

$$\mathcal{R}_N^b f(h,s) = \cos\theta\, \mathcal{R}(s_c, \pi - \theta), \tag{4.3.9}$$

$$\mathcal{R}_N^c f(h,s) = \cos\theta\, \mathcal{R}(s_c, 3\pi/2 - \theta), \tag{4.3.10}$$

$$\mathcal{R}_N^d f(h,s) = \cos\theta\, \mathcal{R}(s_c, 3\pi/2 + \theta). \tag{4.3.11}$$

The back-projection is the dual of this transform with respect to the usual dot product in $\mathbb{R}^{N^2}$. We will denote the back-projection by B_N or $\mathcal{R}_N^T$. If one explicitly forms the matrix for the linear transforms $\mathcal{R}_N$ and $\mathcal{R}_N^T$ they are indeed transposes of each other. $\mathcal{R}_N^T$ is the discrete analogue of $\mathcal{R}^\#$ in (4.1.2), a summation of all values assigned to d-lines passing through a point.

$\mathcal{R}_N^T$ is computed by reversing the sweep (4.3.1) above and computing a sequence of back-projections of decreasing size. Given a matrix $\hat{A} \in \mathbb{R}^{\left(\frac{3}{2}N+\frac{1}{2}\right)\times N}$, the reverse sweep for one

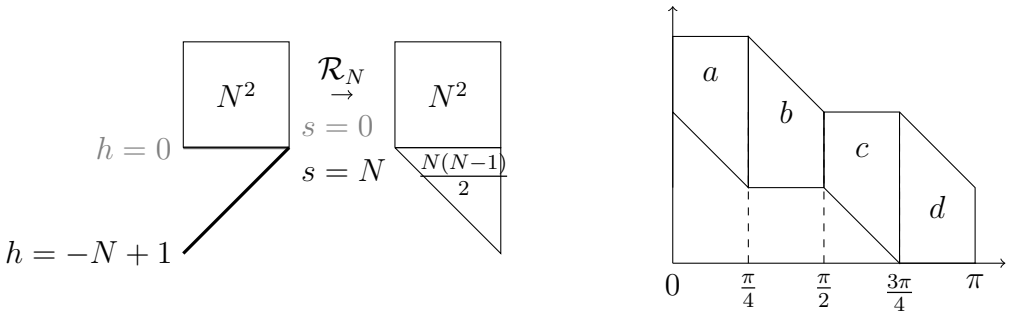

Figure 4.12: The range of a quadrant of a discrete Radon transform (left) and a diagram showing how the boundary of the quadrants $\{a, b, c, d\}$ can be identified (right). Here $\theta = \arctan(s/(N-1))$.

quadrant is given by

$$\begin{aligned} B_n^L(h,s) &= B_{2n}(h,2s) + B_{2n}(h,2s+1), \\ B_n^R(h+s,s) &= B_{2n}(h,2s) + B_{2n}(h-1,2s), \end{aligned} \tag{4.3.12}$$

where the initial array $B_N(h,s) = \hat{A}_{h,s}$, and n is set to $N/2$. This summation is repeated for the two half-images B_n^L and B_n^R on the LHS, until n reaches 1. Again, this summation is only for one quadrant, and we denote the end result as $\left(B_N^a \hat{A}\right)_{i,j}$.

The full back-projection is given by

$$\left(\mathcal{B}_N \hat{A}\right)(i,j) = \frac{1}{4N^2}\left(B_N^a \hat{A} + B_N^b \hat{A} + B_N^c \hat{A} + B_N^d \hat{A}\right) \tag{4.3.13}$$

Its inversion algorithm using a full multi-grid method was demonstrated in [88] along with convergence analysis. In this chapter, we use the conjugate gradient (CG) algorithm as will be discussed below in Section 4.3.3.

We end this section with the remark that the recursion (4.3.1) need not be in two and can be in any prime number, much like the fast Fourier transform [28].

4.3.2 DRT in dimension three

Just as the continuous Radon transform was defined in (4.1.1) for arbitrary number of dimensions n, the DRT can also be generalized to higher dimensions [72]. Here we treat the 3D case as an example. The recursive definitions (4.3.1) for the *d-planes* parametrized by three parameters (h, s_1, s_2) can be derived for each hexadecant in a straightforward manner, as follows

$$\begin{aligned} D_{2n}(h, 2s_1, 2s_2) &= D_n^{LL}(h, s_1, s_2) \cup D_n^{RL}(h + s_1, s_1, s_2) \\ &\quad \cup D_n^{LR}(h + s_2, s_1, s_2) \cup D_n^{RR}(h + s_1 + s_2, s_1, s_2), \end{aligned} \tag{4.3.14}$$

$$\begin{aligned} D_{2n}(h, 2s_1 + 1, 2s_2) &= D_n^{LL}(h, s_1, s_2) \cup D_n^{RL}(h + s_1 + 1, s_1, s_2) \\ &\quad \cup D_n^{LR}(h + s_2, s_1, s_2) \cup D_n^{RR}(h + s_1 + s_2 + 1, s_1, s_2), \end{aligned} \tag{4.3.15}$$

$$\begin{aligned} D_{2n}(h, 2s_1, 2s_2 + 1) &= D_n^{LL}(h, s_1, s_2) \cup D_n^{RL}(h + s_1, s_1, s_2) \\ &\quad \cup D_n^{LR}(h + s_2 + 1, s_1, s_2) \cup D_n^{RR}(h + s_1 + s_2 + 1, s_1, s_2), \end{aligned} \tag{4.3.16}$$

$$\begin{aligned} D_{2n}(h, 2s_1 + 1, 2s_2 + 1) &= D_n^{LL}(h, s_1, s_2) \cup D_n^{RL}(h + s_1 + 1, s_1, s_2) \\ &\quad \cup D_n^{LR}(h + s_2 + 1, s_1, s_2) \cup D_n^{RR}(h + s_1 + s_2 + 1, s_1, s_2). \end{aligned} \tag{4.3.17}$$

The DRT over one hexadecant (a quarter of a quadrant) is defined as the sum over these d-planes as in (4.3.2-4.3.5), and now the full transform in 3D is given by applying these to each of the hexadecant

$$\mathcal{H} = \begin{Bmatrix} aa, & ab, & ac, & ad, \\ ba, & bb, & bc, & bd, \\ ca, & cb, & cc, & cd, \\ da, & db, & dc, & dd \end{Bmatrix}. \tag{4.3.18}$$

Hence, via transposing and flipping the indices as necessary, the full DRT is the ordered tuple

$$\mathcal{R}_N A = \begin{pmatrix} \mathcal{R}_N^{aa} A, & \mathcal{R}_N^{ab} A, & \mathcal{R}_N^{ac} A, & \mathcal{R}_N^{ad} A, \\ \mathcal{R}_N^{ba} A, & \mathcal{R}_N^{bb} A, & \mathcal{R}_N^{bc} A, & \mathcal{R}_N^{bd} A, \\ \mathcal{R}_N^{ca} A, & \mathcal{R}_N^{cb} A, & \mathcal{R}_N^{cc} A, & \mathcal{R}_N^{cd} A, \\ \mathcal{R}_N^{da} A, & \mathcal{R}_N^{db} A, & \mathcal{R}_N^{dc} A, & \mathcal{R}_N^{dd} A \end{pmatrix} \tag{4.3.19}$$

The corresponding back-projection operation for a hexadecant is given by

$$\begin{aligned} B_n^{LL}(h, s_1, s_2) &= B_{2n}(h, 2s_1, 2s_2) + B_{2n}(h, 2s_1 + 1, 2s_2) \\ &\quad + B_{2n}(h, 2s_1, 2s_2 + 1) + B_{2n}(h, 2s_1 + 1, 2s_2 + 1), \end{aligned} \tag{4.3.20}$$

$$\begin{aligned} B_n^{RL}(h + s_1, s_1, s_2) &= B_{2n}(h, 2s_1, 2s_2) + B_{2n}(h - 1, 2s_1 + 1, 2s_2) \\ &\quad + B_{2n}(h, 2s_1, 2s_2 + 1) + B_{2n}(h - 1, 2s_1 + 1, 2s_2 + 1), \end{aligned} \tag{4.3.21}$$

$$\begin{aligned} B_n^{LR}(h + s_2, s_1, s_2) &= B_{2n}(h, 2s_1, 2s_2) + B_{2n}(h, 2s_1 + 1, 2s_2) \\ &\quad + B_{2n}(h - 1, 2s_1, 2s_2 + 1) + B_{2n}(h - 1, 2s_1 + 1, 2s_2 + 1), \end{aligned} \tag{4.3.22}$$

$$\begin{aligned} B_n^{RR}(h + s_1 + s_2, s_1, s_2) &= B_{2n}(h, 2s_1, 2s_2) + B_{2n}(h - 1, 2s_1 + 1, 2s_2) \\ &\quad + B_{2n}(h - 1, 2s_1, 2s_2 + 1) + B_{2n}(h - 1, 2s_1 + 1, 2s_2 + 1). \end{aligned} \tag{4.3.23}$$

The full back-projection is then the average of back-projections B_N over all hexadecants in $\mathcal{H}$,

$$\left(B_N \hat{A}\right)(i, j) = \frac{1}{16N^3} \sum_{k \in \mathcal{H}} B_N^k \hat{A}. \tag{4.3.24}$$

The computational cost of both operations would be $\mathcal{O}(N^3 \log N)$.

4.3.3 Inversion of DRT with Conjugate Gradient Method

In order to use the dimensional splitting method to solve PDEs, a method for computing the inverse of a DRT (4.1.10) is needed. An inversion algorithm using a full multi-grid method appeared in [88]. Here we explore the application of the conjugate gradient method [47] to

the least-squares problem

$$\mathcal{R}_N^T \mathcal{R}_N X = \mathcal{R}_N^T B. \tag{4.3.25}$$

The matrices for the transforms $\mathcal{R}_N^T$ and $\mathcal{R}_N$ are never explicitly formed, as we can use the fast algorithm. We conjecture that the computational cost of a DRT inversion is $\mathcal{O}(N^{5/2} \log N)$ for an $N \times N$ image. Note that this is slightly more costly than $\mathcal{O}(N^2 (\log N)^3)$ that was conjectured for the full multi-grid method [88]. A more careful study of this inversion is of interest on its own right, and will appear elsewhere.

The inversion (4.3.25) is exact only when B lies in the range of $\mathcal{R}_N$. This assumption cannot be satisfied in general once B is evolved with respect to the dynamics of the transformed variables, as in (4.1.9). The changes the transformed side and causes the approximation to $\hat{q}$ to move away from the range of $\mathcal{R}_N$, and becomes a source of error.

A brute force way to avoid this is to simply prolong the original image q before the manipulation, and restrict after the back-projection. This enlarges the range of the transform, and allows one to control the inversion error. Fortunately, oversampling is a feasible option, unlike in the tomography context. Therefore $\mathcal{R}_N$ will be replaced by $\mathcal{R}_{2pN} \mathcal{P}_{2p}$ where $\mathcal{P}_p$ is the 0-th order prologation (where the value of each cell in the original grid is assigned to a $2p \times 2p$ cells in the enlarged grid) and $\mathcal{R}_N^T$ by $\mathcal{S}_{2p} \mathcal{R}_{2pN}^T$ where $\mathcal{S}$ is the restriction operator.

4.4 Inversion of DRT

Let us consider the inversion of the Shepp-Logan phantom often used in tomography. The original image X is shown in the left in Figure 4.13. The forward transform $B = \mathcal{R}_N X$ is plotted in Figure 4.14. The error between the inversion of its forward transform $\tilde{X} = (\mathcal{R}_N^T \mathcal{R}_N)^{-1} \mathcal{R}_N^T B$ and the original image, $X - \tilde{X}$ is shown on the right in Figure 4.13. The error is in the order of 10^{-3} and can be reduced further by controlling the stopping criteria of the CG algorithm. The algorithm converged in 103 iterations and required 186 function and gradient evaluations.

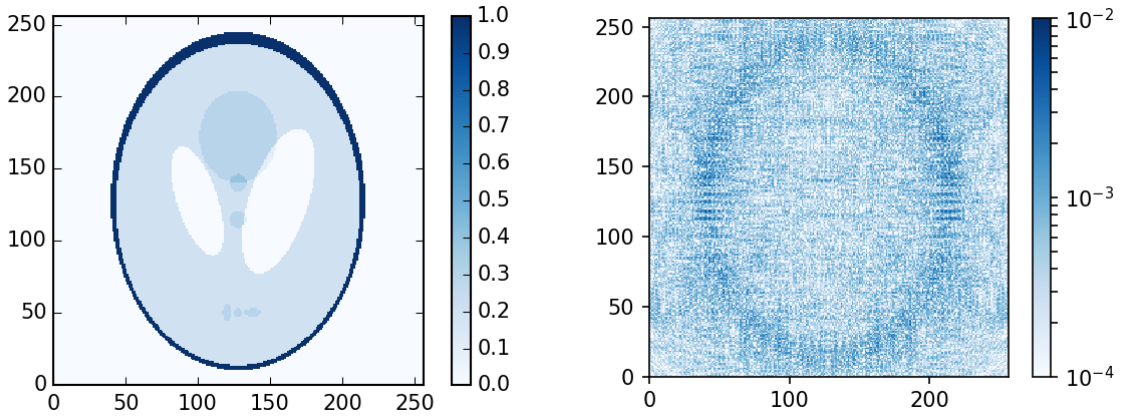

Figure 4.13: The Shepp-Logan phantom of size 256×256 pixels (left) and the DRT inversion error in log-scale (right).

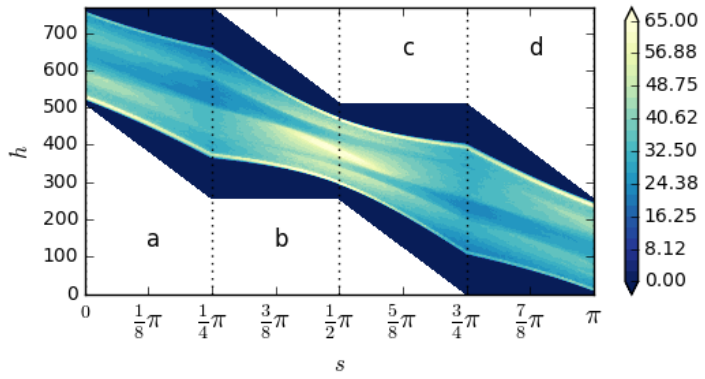

Figure 4.14: DRT of the Shepp-Logan phantom of size 256×256. The original image is shown on the left of Figure 4.13.

Recall that the upper bound on the number of iterations k required for convergence of the CG algorithm to a given tolerance ε is given by

$$k \leq \left\lceil \frac{1}{2}\sqrt{\frac{\lambda_{\max}}{\lambda_{\min}}}\log\frac{2}{\varepsilon} \right\rceil, \tag{4.4.1}$$

where $\lambda_{\min}, \lambda_{\max}$ are the minimum and maximum eigenvalues of $\mathcal{R}_N^T\mathcal{R}_N$, respectively. The eigenvalues can be computed for a slightly modified analogous continuous transform (4.1.1) [80]. Let us define the dual of $\mathcal{R}$ in the weighted L^2 space on the transformed variables $(s, \boldsymbol{\omega})$, which we denote by $\mathcal{R}^*$. We assume a circular domain Ω, so that the codomain is $\{(s, \boldsymbol{\omega}) : s \in [-1, 1], \omega \in S^{n-1}\}$. The weighted inner product is given by

$$\langle \varphi, \psi \rangle = \int_{-1}^{1}\int_{S^{n-1}} \varphi\psi\, w(s)^{1-n}\, \mathrm{d}\boldsymbol{\omega}\, \mathrm{d}s, \quad \text{where } w(s) = (1-s^2)^{1/2}. \tag{4.4.2}$$

Then the singular value decomposition (SVD) of $\mathcal{R}^*\mathcal{R}$ can be computed in terms of orthogonal polynomials and spherical harmonics, and the singular values are given by

$$\sigma_{ml}^2 = \left(\frac{4\pi}{m+1}\right) \quad \text{for } m = 0, 1, ... \text{ and } 0 \leq l \leq m, m+l \text{ even}. \tag{4.4.3}$$

If we truncate the sequence of the squared singular values at the N^2-th term, we have the estimate for the smallest eigenvalue

$$\lambda_{N,\min} \geq \left(\frac{4\pi}{m_0(N^2)+1}\right) \quad \text{where } m_0 = 2\left\lfloor \frac{-1+\sqrt{1+8N^2}}{2} \right\rfloor + 3. \tag{4.4.4}$$

Therefore we obtain the estimate for k,

$$k \leq \left\lceil \frac{1}{2}\sqrt{m_0(N^2)+1}\log\frac{2}{\varepsilon} \right\rceil \leq \left\lceil \frac{1}{\sqrt{2}}\left(\left\lfloor \frac{-1+\sqrt{1+8N^2}}{2} \right\rfloor + 2 \right)^{1/2} \log\frac{2}{\varepsilon} \right\rceil, \tag{4.4.5}$$

therefore we expect the numer of iterations to grow roughly as $\mathcal{O}(\sqrt{N})$.

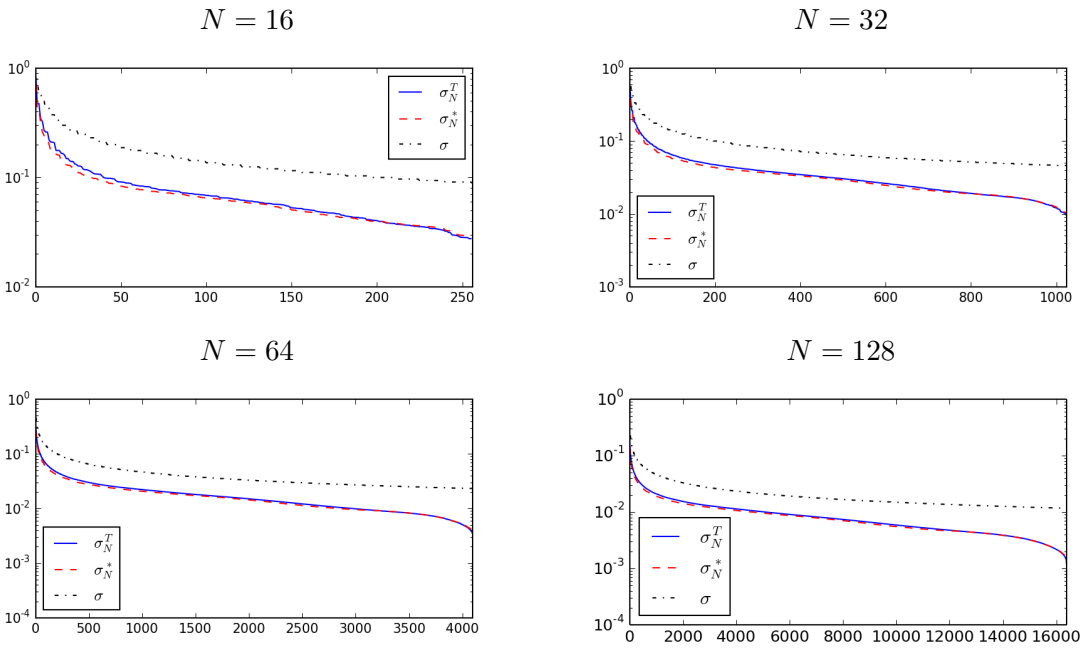

Figure 4.15: Comparison of singular values for varying image sizes. The image sizes are of $N \times N$, and N is set to values $16, 32, 64$ and 128. The singular values σ of the continuous transform $\mathcal{R}^*\mathcal{R}$ is shown in black, σ_N^* of $\mathcal{R}_N^*\mathcal{R}_N$ in red, and σ_N^T of $\mathcal{R}_N^T\mathcal{R}_N$ in blue.

Unfortunately, this estimate is not yet rigorous. First, the exact expression for the singular values assume a disk domain whereas our computational domain is a square. The anisotropy of a square domain hinders a straightforward application of the elegant analysis already done for the disk domain. Second, the DRT approximates the integral along straight lines with d-lines, whose effect is not incorporated yet. Third, the weight function $w(s)$ was neglected in our normal equation (4.3.25). This last point is easily fixed so we will consider the weighted problem below, but we will see that the weighting does not affect the singular values themselves in a significant way.

Let us now compare the eigenvalues computed for the discrete and the continuous case. We compute the singular values for each of $\mathcal{R}^*\mathcal{R}$, $\mathcal{R}_N^*\mathcal{R}$ and $\mathcal{R}^T\mathcal{R}$. The plot of the singular values are shown in Figure 4.15. The singular values σ_N^T and σ_N^* of the DRT consistently decays faster than the continuous transform, and the difference between σ_N^T and σ_N^* is not significant. The faster decay means that (4.4.5) will be optimistic, although the estimate (4.4.1) itself is often pessimistic. In any case, the asymptotic behavior as N increases is similar, so we expect the formula (4.4.5) to be asymptotically correct.

We observe this in Figure 4.16, where we see that the estimate provides a good estimate of the actual number of iterations. The inversion of Shepp-Logan phantoms (Figure 4.13) of increasing image sizes $N \times N$ for $N = 2^5, 2^6, \cdots, 2^{12}$ were computed, and the number of iteration counts were recorded. The estimate (4.4.5) provides a decent guide of the asymptotic behavior, showing the growth of the number of iterations to be $\mathcal{O}(\sqrt{N})$.

This provides confidence in the claim that the computational cost of a DRT inversion is $\mathcal{O}(N^{5/2} \log N)$ for an $N \times N$ image. Note that this is larger than $\mathcal{O}(N^2(\log N)^3)$ that was conjectured by Press for the full multi-grid method [88].

4.5 Summary and discussion

We have introduced a dimensional splitting method using the intertwining property of the Radon transform. Its applications in solving hyperbolic PDEs, imposing absorbing boundary

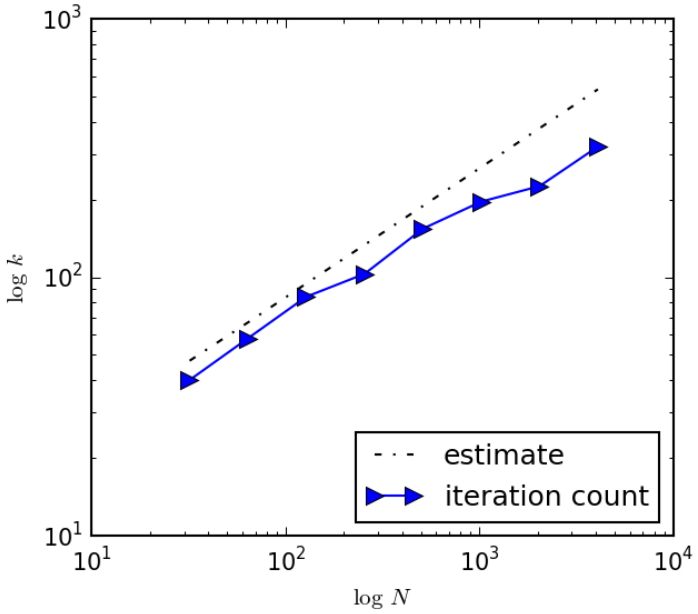

Figure 4.16: Estimate for the number of iterations (4.4.5) vs. realized number of CG iterations for the Shepp-Logan phantom (relative tolerance $\varepsilon = 10^{-4}$).

conditions, and computing displacement interpolations were discussed. For the inversion of DRT the conjugate gradient method was used.

As noted in Section 4.2.1, the dimensional splitting proposed here used with DRT in 3D (Section 4.3.2) allows one to impose absorbing boundary conditions for 3D problems without incurring any error of the type (4.2.3) that appears in 2D. This will be verified in future work. The application of this splitting to fully nonlinear hyperbolic PDEs as discussed in Section 4.1.3 will be studied as well. The utility of the Radon transform for displacement interpolation (Section 4.2.2) will be much more compelling when used in conjunction with the fully multi-dimensional transport reversal [96] as a model reduction tool for general hyperbolic PDEs, and work is underway for such an extension.

The number of CG iterations for the inversion (4.3.25) can be estimated to justify the conjectured $\mathcal{O}(N^{5/2}\log N)$ cost for inversion: this and other inversion results will appear elsewhere. While the prologation used in the inversion (Section 4.3.3) causes expense only of a constant factor, it can be of significant computational cost. Other approaches to reduce the amount of computational effort will be explored. The DRT is essentially a sparse matrix multiplication and is amenable to parallelization, and its performance on graphical processing units (GPUs) will be a future topic of research.

Chapter 5

Conclusion and future work

In this thesis, we considered an uncertainty quantification (UQ) problem arising in tsunami modeling, namely the probabilistic tsunami hazard assessment (PTHA) problem (Section 2.3). A methodology to randomly generate extreme earthquake scenarios and assign probability to each of these scenarios for hazard assessment was introduced (Sections 2.4 and 2.5). We also explored a way to reduce computational costs in creating inundation maps from these scenarios, by exploiting the correlation between coarse-grid and fine-grid runs, in a multi-resolution strategy (Section 2.5). While the multi-resolution strategy yields reasonable results, we have observed an inherent limitation in these methods when the underlying physical phenomena is governed by hyperbolic PDEs. This is due to an inherent limitation in applying singular value decomposition (SVD) to these types of problems.

This difficulty can be overcome with the proposed transport reversal algorithm, which extracts low-dimensional traveling information from snapshot matrices taken from hyperbolic problems (Chapter 3). The algorithm builds on a previous idea of template fitting, and extends it to a more general iterative procedure applicable beyond simple translational symmetries. Motivated by the problem of extending this algorithm to multi-dimensional settings, we introduced a new dimensional splitting method using the Radon transform (Chapter 4). This is a novel dimensional splitting method, that can also be applied to general nonlinear hyperbolic systems, and can be useful in imposing absorbing boundary conditions.

The main effort put forward in the thesis was to lay the groundwork for building a new class of general-purpose reduced-order models (ROMs) for nonlinear hyperbolic systems. There is still much work to be done until such a ROM could be built for practical problems,

such as the PTHA problem.

First, while low-dimensional hyperbolic information could be extracted from transport reversal, it has a more complicated structure than reduced basis often used in previous ROMs. Previously, a reduced basis would simply be a set of small-dimensional linear subspace. Transport reversal yields an output of multiple traveling wave profiles that travel with different speeds and is dynamically masked via the cut-off function. In a parametrized PDE, this output will also vary across different parameters, so further investigations are required to modify these outputs in an appropriate manner.

Second, the transport reversal must be extended to a multi-dimensional setting. Using the dimensional splitting introduced above, one obtains a family of 1D problems. Each family represents a decomposition into planar waves, and many complicated features of wave propagation can be represented by simpler dynamic in the transformed 1D problem. Nonetheless, there now must be adequate communication of information between the different 1D problems over the variable $\boldsymbol{\omega}$ (4.1.1). We believe that this would be possible by devising a generalized variant of the minimization problem (3.2.32).

Third, for the dimensional splitting to be successful, the computational performance of the discrete Radon transform (DRT) and its inversion must be improved. The inversion has much room for improvement, as the naïve prologation technique implemented in Section 4.3.3 can easily become too expensive for tighter threshold of error. The current implementation of the DRT algorithm used in this thesis does not take into consideration machine-dependent processor pipeline or memory hierarchy. The geometric nature of the DRT may yield itself to a faster implementation using domain decomposition.

DRT can be useful in settings outside the construction of ROMs for hyperbolic problems. For instance, it could be helpful for image registration problems when the object tracked is undergoing heavy deformation. The dimensional splitting would turn this into 1D registration problem for each angle, offering an alternative to existing methods. Much like the use of the Fast Fourier transform (FFT) for this class of problems (e.g., see [94]), DRT could prove to be useful in some problem. This topic will also be pursued in future work.

BIBLIOGRAPHY

[1] R. Abgrall, D. Amsallem, and R. Crisovan, *Robust model reduction by L^1-norm minimization and approximation via dictionaries: application to nonlinear hyperbolic problems*, Advanced Modeling and Simulation in Engineering Sciences, 3 (2016), pp. 1–16.

[2] L. Adams, R. LeVeque, and F. González, *The pattern-method for incorporating tidal uncertainty into probabilistic tsunami hazard assessment (PTHA)*, Natural Hazards, 76 (2015), pp. 19–39.

[3] L. M. Adams, R. J. LeVeque, D. Rim, and F. I. Gonzaléz, *Probabilistic Source Selection for the Cascadia Subduction Zone Final Report.* http://depts.washington.edu/ptha/FEMA/FinalReportSourceFilter.pdf, 2017.

[4] D. Amsallem and C. Farhat, *An interpolation method for adapting reduced-order models and application to aeroelasticity*, AIAA Journal, 46 (2008), pp. 1803–1813.

[5] D. Amsallem and C. Farhat, *An online method for interpolating linear parametric reduced-order models*, SIAM Journal on Scientific Computing, 33 (2011), pp. 2169–2198.

[6] D. Amsallem, M. Zahr, Y. Choi, and C. Farhat, *Design optimization using hyper-reduced-order models*, Structural and Multidisciplinary Optimization, 51 (2015), pp. 919–940.

[7] N. Aubry, W.-Y. Lian, and E. Titi, *Preserving symmetries in the proper orthogonal decomposition*, SIAM Journal on Scientific Computing, 14 (1993), pp. 483–505.

[8] A. Averbuch, R. R. Coifman, D. L. Donoho, M. Israeli, Y. Shkolnisky, and I. Sedelnikov,

A framework for discrete integral transformations II: The 2D discrete Radon transform, SIAM Journal on Scientific Computing, 30 (2008), pp. 785–803.

[9] L. S. Bastos and A. O'Hagan, *Diagnostics for Gaussian Process Emulators*, Technometrics, 51 (2009), pp. 425–438.

[10] P. Benner, S. Gugercin, and K. Willcox, *A survey of projection-based model reduction methods for parametric dynamical systems*, SIAM Review, 57 (2015), pp. 483–531.

[11] J.-P. Berenger, *A perfectly matched layer for the absorption of electromagnetic waves*, Journal of Computational Physics, 114 (1994), pp. 185–200.

[12] J.-P. Berenger, *Three-dimensional perfectly matched layer for the absorption of electromagnetic waves*, Journal of Computational Physics, 127 (1996), pp. 363–379.

[13] M. J. Berger and R. J. LeVeque, *Adaptive mesh refinement using wave-propagation algorithms for hyperbolic systems*, SIAM Journal on Numerical Analysis, 35 (1998), pp. 2298–2316.

[14] G. Berkooz, P. Holmes, and J. Lumley, *The proper orthogonal decomposition in the analysis of turbulent flows*, Annual Review of Fluid Mechanics, 25 (1993), pp. 539–575.

[15] G. Berkooz and E. Titi, *Galerkin projections and the proper orthogonal decomposition for equivariant equations*, Physics Letters A, 174 (1993), pp. 94–102.

[16] G. Beylkin, *Discrete Radon transform*, IEEE Transactions on Acoustics, Speech, and Signal Processing, 35 (1987), pp. 162–172.

[17] J. P. Boyd, *Chebyshev & Fourier Spectral Methods*, Springer-Verlag Berlin Heidelberg, 1989.

[18] M. L. Brady, *A fast discrete approximation algorithm for the Radon transform*, SIAM Journal on Computing, 27 (1998), pp. 107–119.

[19] A. Brandt, J. Mann, M. Brodski, and M. Galun, *A fast and accurate multilevel inversion of the Radon transform*, SIAM Journal on Applied Mathematics, 60 (2000), pp. 437–462.

[20] S. C. Brenner and R. Scott, *The Mathematical Theory of Finite Element Methods*, Springer-Verlag New York, 3rd ed., 2008.

[21] K. Carlberg, *Adaptive h-refinement for reduced-order models*, International Journal for Numerical Methods in Engineering, 102 (2015), pp. 1192–1210.

[22] W. C. Chew and W. H. Weedon, *A 3D perfectly matched medium from modified Maxwell's equations with stretched coordinates*, Microwave and Optical Technology Letters, 7 (1994), pp. 599–604.

[23] G. Y. K. Chock, *The ASCE 7 Tsunami Loads and Effects Design Standard*, in Structures Congress 2015, American Society of Civil Engineers, 2015, pp. 1446–1456.

[24] E. Christensen, M. Brons, and J. Sorensen, *Evaluation of proper orthogonal decomposition-based decomposition techniques applied to parameter-dependent nonturbulent flows*, SIAM Journal on Scientific Computing, 21 (2000), pp. 1419–1434.

[25] Clawpack Development Team, *Clawpack software*, 2017, doi:`10.5281/zenodo.262111`, `http://www.clawpack.org`. Version 5.4.0.

[26] K. A. Cliffe, M. B. Giles, R. Scheichl, and A. L. Teckentrup, *Multilevel Monte Carlo methods and applications to elliptic PDEs with random coefficients*, Computing and Visualization in Science, 14 (2011), pp. 3–15.

[27] M. Comninou and J. Dundurs, *The angular dislocation in a half space*, Journal of Elasticity, 5 (1975), pp. 203–216.

[28] J. W. Cooley and J. W. Tukey, *An algorithm for the machine calculation of complex Fourier series*, Mathematics of Computation, 19 (1965), pp. 297–301.

[29] J. Dawson, *Particle simulation of plasmas*, Reviews of Modern Physics, 55 (1983).

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

[31] J. Dick, F. Y. Kuo, and I. H. Sloan, *High-dimensional integration: The quasi-Monte Carlo way*, Acta Numerica, 22 (2013), pp. 133–288.

[32] B. Engquist and A. Majda, *Absorbing boundary conditions for numerical simulation of waves*, Proceedings of the National Academy of Sciences, 74 (1977), pp. 1765–1766.

[33] R. Everson and L. Sirovich., *The Karhunen-Loève procedure for gappy data*, Journal of the Optical Society of America, 12 (1995), pp. 1657–1664.

[34] H. R. Fairbanks, A. Doostan, C. Ketelsen, and G. Iaccarino, *A low-rank control variate for multilevel Monte Carlo simulation of high-dimensional uncertain systems*, Journal of Computational Physics, 341 (2017), pp. 121–139.

[35] G. J. Fix and G. Strang, *An analysis of the finite element method*, Englewood Cliffs, N.J. : Prentice-Hall, Englewood Cliffs, N.J., 1973.

[36] E. L. Geist and T. Parsons, *Probabilistic analysis of tsunami hazards*, Natural Hazards, 37 (2006), pp. 277–314.

[37] E. L. Geist, T. Parsons, U. S. ten Brink, and H. J. Lee, *Tsunami probability*, in The Sea, E. N. Bernard and A. R. Robinson, eds., vol. 15, Harvard University Press, 2009, pp. 201–235.

[38] R. Ghanem, *The nonlinear Gaussian spectrum of log-normal stochastic processes and variables*, Journal of Applied Mechanics, 66 (1999), pp. 964–973.

[39] R. G. Ghanem and P. D. Spanos, *Stochastic Finite Elements: A Spectral Approach*, Springer, 1991.

[40] H. Ghofrani, G. M. Atkinson, K. Goda, and K. Assatourians, *Stochastic finite-fault simulations of the 2011 Tohoku, Japan, Earthquake*, Bulletin of the Seismological Society of America, 103 (2013), pp. 1307–1320.

[41] M. B. Giles, *Multilevel Monte Carlo path simulation*, Operations Research, 56 (2008), pp. 607–617.

[42] F. González, R. LeVeque, and L. Adams, *Probabilistic Tsunami Hazard Assessment (PTHA) for Crescent City, CA, Final Report on Phase I, January 13*, 2013.

[43] F. I. González, E. L. Geist, B. Jaffe, U. Knolu, H. Mofjeld, C. E. Synolakis, V. V. Titov, D. Arcas, D. Bellomo, D. Carlton, T. Horning, J. Johnson, J. Newman, T. Parsons, R. Peters, C. Peterson, G. Priest, A. Venturato, J. Weber, F. Wong, and A. Yalciner, *Probabilistic tsunami hazard assessment at Seaside, Oregon, for near-and far-field seismic sources*, Journal of Geophysical Research – Oceans, 114 (2009), p. C11023.

[44] F. I. Gonzaléz, R. J. LeVeque, and L. M. Adams, *Probabilistic Tsunami Hazard Assessment (PTHA) for Crescent City, CA. Final Report for Phase I.* http://hdl.handle.net/1773/22366, 2013.

[45] W. Götz and H. Druckmüller, *A fast digital Radon transform - an efficient means for evaluating the Hough transform*, Pattern Recognition, 29 (1996), pp. 711–718.

[46] J. C. Gower and G. B. Dijksterhuis, *Procrustes problems*, vol. 3, Oxford University Press, Oxford, 2004.

[47] A. Greenbaum, *Iterative Methods for Solving Linear Systems*, Society for Industrial and Applied Mathematics, 1997.

[48] M. Guatteri, P. M. Mai, G. C. Beroza, and J. Boatwright, *Strong Ground-Motion Prediction from Stochastic-Dynamic Source Models*, Bulletin of the Seismological Society of America, 93 (2003), pp. 301–313.

[49] J. M. Hammersley and D. C. Handscomb, *Monte Carlo Methods*, Springer Netherlands, 1964.

[50] F. Harlow, *A machine calculation method for hydrodynamic problems*, Los Alamos Scientific Laboratory Report LAMS, (1955).

[51] S. Helgason, *Integral Geometry and Radon Transforms*, Springer New York, New York, NY, 2011.

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

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

[54] S. P. Huang, S. T. Quek, and K. K. Phoon, *Convergence study of the truncated Karhunen-Loève expansion for simulation of stochastic processes*, International Journal for Numerical Methods in Engineering, 52 (2001), pp. 1029–1043.

[55] M. A. Jaimes, E. Reinoso, M. Ordaz, B. Huerta, R. Silva, E. Mendoza, and J. C. Rodríguez, *A new approach to probabilistic earthquake-induced tsunami risk assessment*, Ocean & Coastal Management, 119 (2016), pp. 68–75.

[56] B. T. Kelley and V. K. Madisetti, *The fast discrete Radon transform. I. Theory*, IEEE Transactions on Image Processing, 2 (1993), pp. 382–400.

[57] M. Kirby and D. Armbruster, *Reconstructing phase space from PDE simulations,*, Zeitschrift für angewandte Mathematik und Physik, 43 (1992), pp. 999–1022.

[58] K. Kunisch and S. Volkwein, *Control of Burgers equation by a reduced order approach using proper orthogonal decomposition*, Journal of Optimization Theory and Applications, 102 (1999), pp. 345–371.

[59] D. Lavallée, P. Liu, and R. J. Archuleta, *Stochastic model of heterogeneity in earthquake slip spatial distributions*, Geophysical Journal International, 165 (2006), pp. 622–640.

[60] R. J. LeVeque, *Large time step shock-capturing techniques for scalar conservation laws*, SIAM Journal on Numerical Analysis, 19 (1982), pp. 1091–1109.

[61] R. J. LeVeque, *Convergence of a large time step generalization of Godunov's method for conservation laws*, Communications on Pure and Applied Mathematics, 37 (1984), pp. 463–477.

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

[63] R. J. LeVeque, *Wave propagation algorithms for multidimensional hyperbolic systems*, Journal of Computational Physics, 131 (1997), pp. 327–353.

[64] R. J. LeVeque, *Finite Volume Methods for Hyperbolic Problems*, Cambridge University Press, Cambridge, 1st ed., 2002.

[65] R. J. LeVeque, D. L. George, and M. J. Berger, *Tsunami modelling with adaptively refined finite volume methods*, Acta Numerica, 20 (2011), pp. 211–289.

[66] R. J. LeVeque, K. Waagan, F. I. González, D. Rim, and G. Lin, *Generating random earthquake events for probabilistic tsunami hazard assessment*, Pure and Applied Geophysics, 173 (2016), pp. 3671–3692.

[67] J. Li, J. Li, and D. Xiu, *An efficient surrogate-based method for computing rare failure probability*, Journal of Computational Physics, 230 (2011), pp. 8683–8697.

[68] S. Lloyd, *Least squares quantization in PCM*, IEEE Transactions on Information Theory, 28 (1982), pp. 129–137.

[69] F. Løvholt, G. Pedersen, S. Bazin, D. Kühn, R. E. Bredesen, and C. Harbitz, *Stochastic analysis of tsunami runup due to heterogeneous coseismic slip and dispersion*, Journal of Geophysical Research, 117 (2012).

[70] Y. Maday and B. Stamm, *Locally adaptive greedy approximations for anisotropic parameter reduced basis spaces*, SIAM Journal on Scientific Computing, 35 (2013), pp. A2417–A2441.

[71] P. M. Mai and G. C. Beroza, *A spatial random field model to characterize complexity in earthquake slip*, Journal of Geophysical Research, 107 (2002), pp. ESE10–1 – ESE10–21.

[72] J. G. Marichal-Hernández, J. P. Lüke, F. L. Rosa, and J. M. Rodríguez-Ramos, *Fast approximate 4D: 3D discrete Radon transform, from light field to focal stack with $\mathcal{O}(n^4)$ sums*, Journal of Electronic Imaging, 21 (2012), pp. 1091–1109.

[73] F. Matus and J. Flusser, *Image representation via a finite Radon transform*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 15 (1993), pp. 996–1006.

[74] R. I. McLachlan and G. R. W. Quispel, *Splitting methods*, Acta Numerica, 11 (2002), p. 341434.

[75] B. J. Meade, *Algorithms for the calculation of exact displacements, strains, and stresses for triangular dislocation elements in a uniform elastic half space*, Computers & Geosciences, 33 (2007), pp. 1064–1075.

[76] D. Melgar, R. M. Allen, S. Riquelme, J. Geng, F. Bravo, J. C. Baez, H. Parra, S. Barrientos, P. Fang, Y. Bock, M. Bevis, D. J. Caccamise, C. Vigny, M. Moreno, and R. Smalley, *Local tsunami warnings: Perspectives from recent large events*, Geophysical Research Letters, 43 (2016), pp. 1109–1117.

[77] D. Melgar, R. J. LeVeque, D. S. Dreger, and R. M. Allen, *Kinematic rupture scenarios and synthetic displacement data: An example application to the Cascadia Subduction Zone*, Journal of Geophysical Research – Solid Earth, 121 (2016), pp. 6658–6674.

[78] D. Motazedian and G. M. Atkinson, *Stochastic Finite-Fault Modeling Based on a Dynamic Corner Frequency*, Bulletin of the Seismological Society of America, 95 (2005), pp. 995–1010.

[79] A. Narayan, C. Gittelson, and D. Xiu, *A stochastic collocation algorithm with multifidelity models*, SIAM Journal on Scientific Computing, 36 (2014), pp. A495–A521.

[80] F. Natterer, *The Mathematics of Computerized Tomography*, Society for Industrial and Applied Mathematics, 2001.

[81] F. Nobile, R. Tempone, and C. G. Webster, *A sparse grid stochastic collocation method for partial differential equations with random input data*, SIAM Journal on Numerical Analysis, 46 (2008), pp. 2309–2345.

[82] Y. Okada, *Surface deformation due to shear and tensile faults in a half-space*, Bulletin of the Seismological Society of America, 75 (1985), pp. 1135–1154.

[83] A. Olsson and G. Sandberg, *Latin Hypercube Sampling for Stochastic Finite Element Analysis*, Journal of Engineering Mechanics, 128 (2002), pp. 121–125.

[84] A. F. Oskooi, L. Zhang, Y. Avniel, and S. G. Johnson, *The failure of perfectly matched layers, and towards their redemption by adiabatic absorbers*, Optics Express, 16 (2008), pp. 11376–11392.

[85] H. M. Park and M. W. Lee, *An efficient method of solving the Navier-Stokes equations for flow control*, International Journal for Numerical Methods in Engineering, 41 (1998), pp. 1133–1151.

[86] A. Paul-Dubois-Taine and D. Amsallem, *An adaptive and efficient greedy procedure for the optimal training of parametric reduced-order models*, International Journal for Numerical Methods in Engineering, 102 (2015), pp. 1262–1292.

[87] B. Peherstorfer, T. Cui, Y. Marzouk, and K. Willcox, *Multifidelity importance sampling*, Computer Methods in Applied Mechanics and Engineering, 300 (2016), pp. 490–509.

[88] W. H. Press, *Discrete Radon transform has an exact, fast inverse and generalizes to operations other than sums along lines*, Proceedings of the National Academy of Sciences, 103 (2006), pp. 19249–19254.

[89] R. Pulch and D. Xiu, *Generalised polynomial chaos for a class of linear conservation laws*, Journal of Scientific Computing, 51 (2011), pp. 293–312.

[90] J. Radon, *On determination of functions by their integral values along certain multiplicities*, Berichte der Sächsischen Akadamie der Wissenschaft, 69 (1917), pp. 262–277.

[91] M. Rathinam and L. Petzold, *Dynamic iteration using reduced order models: A method for simulation of large scale modular systems*, SIAM Journal on Numerical Analysis, 40 (2002), pp. 1446–1474.

[92] M. Rathinam and L. Petzold, *A new look at proper orthogonal decomposition*, SIAM Journal on Numerical Analysis, 41 (2003), pp. 1893–1925.

[93] S. Ravindran, *Reduced-order adaptive controllers for fluid flows using proper orthogonal decomposition*, Journal of Scientific Computing, 15 (2000), pp. 457–478.

[94] B. S. Reddy and B. N. Chatterji, *An FFT-based technique for translation, rotation, and scale-invariant image registration*, IEEE Transactions on Image Processing, 5 (1996), pp. 1266–1271.

[95] J. Reiss, P. Schulze, and J. Sesterhenn, *The shifted proper orthogonal decomposition: A mode decomposition for multiple transport phenomena*, Preprint, `arXiv:1512.01985` (2015).

[96] D. Rim, S. Moe, and R. LeVeque, *Transport reversal for model reduction of hyperbolic partial differential equations*, Preprint, `arXiv:1701.07529` (2017).

[97] C. W. Rowley, I. G. Krevrekidis, J. E. Marsden, and K. Lust, *Reduction and reconstruction for self-similar dynamical systems*, Nonlinearity, (2003).

[98] C. W. Rowley and J. E. Marsden, *Reconstruction equations and the Karhunen-Loève expansion for systems with symmetry*, Physica D, (2000).

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

[100] A. Sarri, S. Guillas, and F. Dias, *Statistical emulation of a tsunami model for sensitivity analysis and uncertainty quantification*, Natural Hazards and Earth System Sciences, 12 (2012), pp. 2003–2018.

[101] C. Schwab and R. A. Todor, *Karhunen-Loève approximation of random fields by generalized fast multipole methods*, Journal of Computational Physics, 217 (2006), pp. 100–122.

[102] N. Smaoui and D. Armbruster, *Symmetry and the Karhunen-Loève analysis*, SIAM Journal on Scientific Computing, 18 (1997), pp. 1526–1532.

[103] R. C. Smith, *Uncertainty Quantification: Theory, Implementation, and Applications*, Society for Industrial and Applied Mathematics, 2014.

[104] G. Strang, *On the construction and comparison of difference schemes*, SIAM Journal on Numerical Analysis, 5 (1968), pp. 506–517.

[105] D. Sulsky, Z. Chen, and H. Schreyer, *A particle method for history-dependent materials*, Computer Methods in Applied Mechanics and Engineering, 118 (1994), pp. 179–196.

[106] A. L. Thomas, *Poly3D : a three-dimensional, polygonal element, displacement discontinuity boundary element computer program with applications to fractures, faults, and cavities in the Earth's crust*, M.Sc. Thesis, Stanford University, (1993).

[107] E. F. Toro, *Riemann Solvers and Numerical Methods for Fluid Dynamics : A Practical Introduction*, Dordrecht, Dordrecht, 2009.

[108] L. Trefethen, *Spectral Methods in MATLAB*, Society for Industrial and Applied Mathematics, 2000.

[109] J. H. Tu, C. W. Rowley, D. M. Luchtenburg, S. L. Brunton, and J. N. Kutz, *On dynamic mode decomposition: Theory and applications*, Journal of Computational Dynamics, 1 (2014), pp. 391–421.

[110] USGS, *USGS earthquake archive*, 2010, `http://earthquake.usgs.gov/earthquakes/eqinthenews/2010/us2010tfan/finite_fault.php`.

[111] B. van Leer, *Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov's method*, Journal of Computational Physics, 32 (1979), pp. 101–136.

[112] K. Veroy and A. Patera, *Certified real-time solution of the parametrized steady incompressible Navier-Stokes equations: Rigorous reduced-basis a posteriori error bounds*, International Journal for Numerical Methods in Fluids, 47 (2005), pp. 773–788.

[113] C. Villani, *Optimal Transport: Old and New*, vol. 338, Springer Science & Business Media, 2008.

[114] C. J. Vogl and R. J. LeVeque, *A high-resolution finite volume seismic model to generate seafloor deformation for tsunami modeling*, Preprint, `arXiv:1701.01430 [cs.CE]` (2017).

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

[116] R. C. Witter, Y. Zhang, K. Wang, G. R. Priest, C. Goldfinger, L. L. Stimely, J. T. English, and P. A. Ferro, *Simulating tsunami inundation at Bandon, Coos County, Oregon, using hypothetical Cascadia and Alaska earthquake scenarios*, (2011).

[117] D. Xiu, *Numerical methods for stochastic computations: A spectral method approach*, Princeton University Press, Princeton, N.J., 2010.

[118] A. York, D. Sulsky, and H. Schreyer, *Fluid-membrane interaction based on the material-point method*, International Journal for Numerical Methods in Engineering, 48 (2000), pp. 901–924.