

# Transformed Snapshot Interpolation with High Resolution Transforms

G. Welper\*

## Abstract

In the last few years, several methods have been developed to deal with jump singularities in parametric or stochastic hyperbolic PDEs. They typically use some alignment of the jump-sets in physical space before performing well established reduced order modelling techniques such as reduced basis methods, POD or simply interpolation. In the current literature, the transforms are typically of low resolution in space, mostly low order polynomials, Fourier modes or constant shifts. In this paper, we discuss higher resolution transforms in one of the recent methods, the transformed snapshot interpolation (TSI). We introduce a new discretization of the transforms with an appropriate behaviour near singularities and consider their numerical computation via an optimization procedure.

**Keywords:** Parametric PDEs, shocks, transformations, interpolation, convergence rates, optimization

**AMS subject classifications:** 41A46, 41A25, 35L67, 65M12

## 1 Introduction

One important ingredient in reduced order modeling and PDEs with random coefficients is the numerical approximation of parametric functions  $u(x, \mu)$  with physical variables  $x \in \Omega \subset \mathbb{R}^d$  and deterministic or random parameters  $\mu$  in some parameter space  $\mathcal{P} \subset \mathbb{R}^n$ . The literature provides a rich toolbox for this kind of approximation, see e.g. [11, 4, 37, 21] for an overview. However, if  $u(x, \mu)$  has parameter dependent jumps or kinks, the vast majority of the available methods suffer from low convergence rates [12, 31, 47]. As a result, problems from parametric hyperbolic PDEs, elliptic PDEs with jumping diffusion (in parameter dependent locations) and parametric level-set methods still pose severe challenges.

Besides a large body of literature on stability, offline/online decompositions and error estimators for hyperbolic or singularly perturbed problems [9, 20,

---

\*Department of Mathematics, University of Central Florida, Orlando, FL 32816, USA, email [gerrit.welper@ucf.edu](mailto:gerrit.welper@ucf.edu).

This material is based upon work supported by the National Science Foundation under Grant No. 1912703.

19, 29, 34, 36, 44, 26, 16, 48, 35, 32, 15, 14, 2, 23, 5], in the last few years, several groups have addressed the poor regularity of  $u(x, \mu)$  and developed several methods with drastically improved convergence behaviour. While some use localization strategies [12, 33], currently the majority relies on additional transforms, shifts or similar tools in order to alleviate the issue. All these approaches have in common that they try to align the discontinuities or sharp gradients in physical space before applying classical reduced order modelling techniques as interpolation, POD or reduced bases. This alignment drastically improves the smoothness [25], Kolmogorov  $n$ -width, or decay rate of the singular values, respectively, which results in considerable efficiency gains in the presence of jumps. Some approaches use characteristics [43], Lagrangian formulations [27], extra equations or conditions [17, 30, 18] or jump tracking [42]. Another approach is to use displacement interpolation or optimal transport [22, 39, 40], which naturally builds on transport along the physical variables. One of the more common recent approaches uses a composition of a snapshot  $u(x - s, \mu)$  with a shift  $s$  or alternatively a transform  $u(X(x), \mu)$  in order to align the jump locations in parameter, where  $s$  or  $X$  usually depend on the parameter and may or may not depend on the location  $x$ . These shifted or transformed snapshots are then used for interpolation, reduced bases or further compressed by POD, see e.g. [47, 38, 8, 7, 28, 46]. Alternatively, in [41], instead of a function composition, the authors shift the coefficients of a discretization.

For the latter class of methods, one needs numerical procedures to find the shifts  $s$  or transforms  $X$ . Since the shifts or transforms generally depend on  $x$  and  $\mu$ , one first chooses a suitable discretization. Then, one minimizes an error formula among the chosen discrete representation during the “offline phase”, either between individual (transformed) snapshots or the full reconstruction error of the method. It seems that in all of the current literature the discretizations are of low order in physical space either constants, low order polynomials or Fourier modes. Clearly, these limitations on the transform’s resolutions pose severe limitations on their applicability to many practical problems. As we will see in Section 3, already quite simple scenarios require transforms with high resolution in physical space.

As a first part of this paper, we discuss possible discretizations of high resolution transforms, which in the most general case are functions  $X(x, \mu)$  depending on the physical variables  $x$  and an arbitrary parameter  $\mu$ . As we will discuss in Section 3, these functions share some major properties with  $u$  itself: They may require high resolution in the spacial variables and can be non-smooth in parameter. However, we cannot use the same discretization strategy as for  $u$  without introducing a second layer of transforms. Instead, we will see that these transforms are naturally given as solutions of ODEs (generally different from characteristic ODEs) and that variants of implicit solvers thereof provide efficient ways for their discretization.

In principle, one can then use the same optimization strategies as for low resolution transforms, but as we will see in some experiments below, this results in poor outcomes. Typically, naive implementations of gradient descent optimizers stall after a few steps and the resulting transform is visually indistinguishable

from the initial transform. In order to overcome this problem, in Section 6, we discuss that the transforms  $X$  are functions of  $x$  and must be considered in appropriate function spaces. This entails that the Fréchet derivative of the objective function to optimize the transform is only a functional in a dual space and must be appropriately “lifted” by a Riesz map to obtain a well defined gradient descent method. In this paper, we report on two lifting methods, one by inverting a Laplacian and one by using a multi-level frame.

The paper is organized as follows: To keep it self contained, in Section 2, we briefly review the transformed snapshot interpolation (TSI), which is one of the recently introduced transformed based reduced order modeling techniques for problems with jumps. Then, in Section (3), we consider an example problem to better understand the requirements for high resolution transforms. In Section 4 we discuss the discretization of the transforms and in Section 5 consider their stability. In Section 6, we introduce suitable optimizers to practically find the transforms and finally in Section 7 we provide some numerical experiments.

## 2 Transformed Snapshot Interpolation

In order to keep the paper self contained, let us first briefly review the *transformed snapshot interpolation* (TSI) introduced in [47]. It approximates a parametric function  $u(x, \mu)$  with physical variables  $x \in \Omega \subset \mathbb{R}^d$  and deterministic or random parameters  $\mu \in \mathcal{P} \subset \mathbb{R}^n$  by

$$u(x, \mu) \approx u_m(x, \mu) := u_m(x, \mu; X) := \sum_{\eta \in \mathcal{P}_m} \ell_\eta(\mu) u(X(\eta; \mu, x), \eta), \quad (1)$$

where  $\ell_\eta(\mu)$  are Lagrange basis polynomials with respect to given interpolation points  $\eta \in \mathcal{P}_m \subset \mathcal{P}$ . Similar to reduced basis methods and PODs, this interpolation only requires us to know snapshots  $u(\cdot, \eta)$  at finitely many interpolation points in parameter. In addition we need the transforms  $\eta \rightarrow X(\eta; \mu, x) \subset \Omega$  which can be regarded as curves in physical space, together with an initial value  $X(\mu; \mu, x) = x$ . We often write  $X(\eta) = X(\eta; \mu, x)$  if  $\mu$  and  $x$  are understood from context.

As a notational convention, both  $\mu$  and  $\eta$  refer to parameters, where  $\mu$  usually denotes a target parameter where we want to approximate  $u(\cdot, \mu)$  and  $\eta$  is either a continuous auxiliary parameter in  $\mathcal{P}$  as in the curve  $\eta \rightarrow X(\eta)$  or a discrete “source” parameter in  $\mathcal{P}_m$  at locations where we know the snapshots, e.g. as in the definition of the TSI (1). The distinction between continuous or discrete should always be clear from context.

Let us now come back to the TSI (1). Choosing the trivial transform  $X(\eta; \mu, x) = x$ , the formula reduces to a standard polynomial interpolation. However, this simple approach does not achieve good convergence rates in case  $u(x, \mu)$  has jumps or kinks in parameter dependent locations, because these entail that  $u$  is not even differentiable in the interpolation direction. The extra transform  $X(\eta; \mu, x)$  is used to align these jumps so that they become “invisible”

to the interpolation. In other words, for a target parameter  $\mu$  and interpolation point  $\eta$ , the transform  $X(\eta; \mu, x)$  ensures that the *transformed snapshots*  $(x, \eta) \rightarrow u(X(\eta; \mu, x), \eta)$  have jumps in the same locations as the correct solution  $u(\cdot, \mu)$ , independent of  $\eta$ . In particular, for all  $x$ , which are not in the jump-set of  $u(\cdot, \mu)$ , the transformed snapshots are smooth in  $\eta$ . The TSI (1) is merely a polynomial interpolation of these transformed snapshots in  $\eta$ , and thus provides high order accuracy.

Note that in the TSI (1) the transforms  $X(\eta; \mu, x)$  are only evaluated at finitely many interpolation points  $\eta \in \mathcal{P}_m$  and the continuous variables  $x \in \Omega$  and  $\mu \in \mathcal{P}$ . If the transforms are smooth in  $\mu$ , they can be efficiently discretized in  $\mu$  by interpolation

$$X(\eta; \mu, x) = \sum_{\gamma \in \mathcal{P}_m} \ell_\gamma(\mu) X(\eta; \gamma, x), \quad (2)$$

called *low resolution transforms* in the following. This was the original choice in [47] together with low order polynomials for the remaining finitely many functions  $x \rightarrow X(\eta; \gamma, x)$ ,  $\eta, \gamma \in \mathcal{P}_m$  with zero normal flux as a numerical substitute for the condition  $X(\eta; \mu, x) \in \Omega$ . As we will see e.g. in Section 3, neither the smoothness in  $\mu$  nor the choice of low order polynomials in  $x$  is warranted in some fairly simple examples. Therefore, we will discuss alternative representations of the transforms in Section 4.

Finally, similar to neural network training or greedy algorithms for reduced basis methods, the remaining transforms  $X(\eta; \gamma, x)$ ,  $\eta, \gamma \in \mathcal{P}_m$  are trained by minimizing the worst case error

$$\sup_{\mu \in \mathcal{P}_T} \|u(x, \mu) - u_m(x, \mu; X)\|_{L_1(\Omega)}, \quad (3)$$

Since we do not know  $u(\cdot, \mu)$  for all  $\mu \in \mathcal{P}$ , we cannot calculate the full error and confine ourselves to a training sample  $\mathcal{P}_T \subset \mathcal{P}$ . Of course other  $L_p$ -norms in parameter are also possible. The  $L_1$ -norm for the physical variables is natural for hyperbolic problems and ensures that the objective function of this optimization problem is differentiable almost everywhere, given that  $u$  is of bounded variation, see [47] for more details.

### 3 An Example Problem

In this section, we first consider a simple example problem that highlights some typical difficulties for the low resolution transforms (2). Namely, let us consider the function

$$u(x, \mu) = \begin{cases} 1 & x \leq -(1 - \mu) \\ -1 & x \geq 1 - \mu \\ 0 & \text{else} \end{cases}, \quad (4)$$

for  $\mu < 1$  and

$$u(x, \mu) = \begin{cases} 1 & x \leq 0 \\ -1 & x > 0 \end{cases}, \quad (5)$$

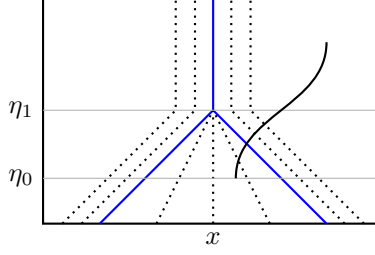


Figure 1: Blue: Jump set of the example function (4), (5) in the  $x - \mu$ -plane. Dotted lines: Curves  $\eta \rightarrow X(\eta; \mu, x)$  defined in (6). Black: Inappropriate transform  $\eta \rightarrow X(\eta; \mu, x)$  crossing a jump.

for  $\mu \geq 1$ , depicted in Figure 1. It serves as a prototype of problems for which the original TSI as introduced in Section 2 is problematic: It starts out with two jumps for  $\mu < 1$  that collide into one single jump for  $\mu \geq 1$ . This poses a problem for the TSI because we cannot properly align a snapshot with a single jump with a snapshot with two jumps. More formally, the TSI interpolates the transformed snapshots  $\eta \rightarrow u(X(\eta; \mu, x), \eta)$ , which should ideally be smooth in the  $\eta$  variable. Since  $\eta \rightarrow X(\eta; \dots)$  itself is smooth, this can only be true if  $X(\eta; \mu, x)$  is not contained in the jump-set of  $u(\cdot, \eta)$  for all  $\eta \in \mathcal{P}$ . Obviously this is impossible for a curve starting between the two jumps in the lower half of Figure 1 indicated by the black solid line crossing a blue jump. In fact, this condition forces all curves  $\eta \rightarrow X(\eta; \mu, x)$ , which start at  $\mu, x$  between the two jumps, to converge to the collision point for  $\eta \rightarrow 1$ , indicated by the dotted lines. The remaining dotted lines complete this to a reasonable transform given by

$$X(\eta; \mu, x) = \begin{cases} x - (\mu - \eta) & x \leq -(1 - \mu) \\ x + (\mu - \eta) & x \geq 1 - \mu \\ x - x \frac{\mu - \eta}{\mu - 1} & \text{else} \end{cases} . \quad (6)$$

In particular, if we confine our parameter range to  $\mu < 1$  or  $\mu \geq 1$ , these transforms never cross jumps, corresponding to the parameter ranges where  $u$  has either one or two jumps. Therefore, we may proceed as follows: We subdivide the parameter domain into two pieces for  $\mu \leq 1$  and apply a TSI on each of them. As simple version of this idea, which finds partitions of  $\mathcal{P}$  via  $h$  or  $hp$  adaption, is discussed in [46]. A more sophisticated algorithm that cuts the parameter domain precisely at the jump collision is in preparation. In this paper, we do not deal with the proper subdivision but with the necessarily singular transforms:

1. For  $\mu = 1$  and  $\eta < 1$ , the transform has a jump at  $x = 0$ .
2. The dependence of the  $\mu$  variable is of the type  $\sim 1/(\mu - 1)$  in the regions between the two jumps and therefore singular at  $\mu = 1$ .

Both problems cannot be handled with the low resolution transforms (2). These were low order polynomials both in the  $x$  and  $\mu$  variables. For the  $x$  variable one can simply increase the spacial resolution of the transforms to the same resolution of the snapshots. Since the latter must be stored anyways, the overall degrees of freedom do not blow up significantly. However, this higher resolution requires some extra care with regard to the optimization of the error (3), as is discussed in Section 6.

With regard to the singularity in the  $\mu$  variable, a simple increase of the resolution is more challenging without increasing the number of snapshots and therefore we discuss an alternative discretization of the transform in Section 4.

Although the example discussed in this section is quite simple, the arising problems are quite typical for problems where the jump-sets cannot be properly aligned between snapshots. As we will see later in Section 4 when we have developed some better tools, singular behaviour of the transforms generally occurs when the jump-set topology changes in parameter.

## 4 Transforms as solutions of ODEs

In this section, we consider discretizations of the transforms, which allow the types of singular behaviour that we observed in our introductory example in Section (3). We have already chosen a notation for the transforms  $\eta \rightarrow X(\eta) = X(\eta; \mu, x)$  with “initial value”  $X(\mu, \mu, x) = x$  that suggests to define them as a solution of an ODE

$$\frac{d}{d\eta} X(\eta) = \Phi(X(\eta), \eta), \quad X(\mu) = x, \quad (7)$$

where we call  $\Phi : \Omega \times \mathcal{P} \rightarrow \mathbb{R}$  the *transport field*. First note that this definition restricts us to one dimensional parameter domains. Nonetheless, higher dimensional parameters can be treated by a tensor product type construction as shown in [46], where we split dimensions of the parameter space one by one. Even for one dimensional parameters, defining transforms via ODEs restricts our possible choices since it induces a semi-group property in the dynamic variable of the ODE:  $X(\eta; \mu, x) = X(\eta; \nu, X(\nu; \mu, x))$ . However, this restriction does not seem to be overly severe and has two major benefits: It was argued in [47] that it greatly helps optimizers to find global minima of (3) and as we shall see in this section, it provides a way to find discretizations of the transforms in view of singularities.

Let us first review the example in Figure 1 again. In order to properly align the jumps, the blue lines must be a solution of the ODE (7), defining the transforms. This implies that the trajectories intersect at the jump collision so that the ODE does not have a unique solution. Therefore, at least one of the assumption of the Picard Lindelöf theorem cannot be valid, which is typically the Lipschitz continuity of  $\Phi$ . This provides another motivation for dealing with high spacial resolution transforms.

One can easily generalize this observation: If the jump-sets of two snapshots  $u(\cdot, \mu)$  and  $u(\cdot, \eta)$  are not homeomorphic, no ODE, which aligns the jumps, can satisfy all assumptions of the Picard Lindelöf theorem. For if one would, the solution  $x \rightarrow X(\eta; \mu, x)$  would be a homeomorphism of the respective jump-sets, contradicting the assumption of having a change in topology.

In order to find a numerical method based on the ODE (7), let us assume for the moment that we know  $\Phi(x, \mu)$  for all possible values of  $x$  and  $\mu$ . Then we still have to solve the ODE (7) to find  $X(\eta)$  and ultimately the TSI interpolation. The easiest choice that comes to mind is Euler's method

$$X(\eta) \approx x + (\eta - \mu)\Phi(x, \mu),$$

which has the drawback that we have to be able to evaluate  $\Phi$  at every possible target parameter  $\mu$ , which are of course unknown at the time we compute the transforms. If however we use the implicit Euler method, we have

$$X(\eta) \approx x + (\eta - \mu)\Phi(X(\eta), \eta), \quad (8)$$

where  $\Phi$  is only evaluated at the parameter  $\eta$ , which for the TSI reconstruction is only required at the finitely many snapshot locations  $\mathcal{P}_m$ .

Of course the implicit Euler method limits trajectories  $X(\eta)$  to lines, so that in the remainder of this section, we construct higher order alternatives. First note that we cannot use higher order ODE solvers out of the box: Runge Kutta methods require knowledge of  $\Phi(\cdot, \eta)$  at intermediate parameters depending on the unknown target  $\mu$  and multi-step methods rely on single step methods to get started.

Given these difficulties, we take an alternative but related route. Assume that we know  $\Phi(\cdot, \eta)$  at the interpolation points  $\eta \in \mathcal{P}_n$ , just as we know the snapshots  $u(\cdot, \eta)$  at all  $\mathcal{P}_m$ . Here  $\mathcal{P}_n$  is another set of finitely many nodes, which can in principle differ from  $\mathcal{P}_m$ . Restricting the ODE (7) to the information we have, we obtain

$$X'(\eta) = \Phi(X(\eta), \eta) \text{ for } \eta \in \mathcal{P}_n \quad X(\mu) = x, \quad (9)$$

As we shall see in Proposition 4.1 below, these conditions are sufficient to uniquely determine the curve  $X(\eta)$ , if we restrict them to be polynomials of degree  $n = |\mathcal{P}_n|$ , denoted by  $\mathbb{P}^n$ . The resulting transforms  $X(\eta)$  are called *high resolution transforms* in the following.

Let us first verify that this construction contains the implicit Euler method as a special case. For one single interpolation point, the transform of (9) is a line and therefore uniquely determined by the initial condition  $X(\mu) = x$  and its slope,  $X'(\eta) = \Phi(X(\eta), \eta)$  at the interpolation point  $\eta$ . This implies that the curve is given by

$$X(\xi) = x + (\xi - \mu)\Phi(X(\eta), \eta)$$

at any point  $\xi \in \mathcal{P}$ , which corresponds the implicit Euler method for  $\xi = \eta$ .

Similar to ODE theory, the next proposition provides some conditions under which the interpolation problem (9) has a unique solution.

**Proposition 4.1.** *Let  $I$  be an interval that contains the interpolation points  $\mathcal{P}_n$  and the initial point  $\mu$ .*

1. *Assume that all  $\Phi(\cdot, \eta) : \mathbb{R}^d \rightarrow \mathbb{R}^d$ ,  $\eta \in \mathcal{P}_n$  are bounded and continuous. Then the interpolation problem (9) has a solution in  $\mathbb{P}^n$  restricted to  $I$ .*
2. *Assume that all  $\Phi(\cdot, \eta) : \mathbb{R}^d \rightarrow \mathbb{R}^d$ ,  $\eta \in \mathcal{P}_n$  are Lipschitz continuous with Lipschitz constant  $L$  and that  $\Lambda_{n-1}L|I| < 1$ , where  $\Lambda_{n-1} := \max_{\mu \in I} \sum_{\eta \in \mathcal{P}_n} |\ell_\eta(\mu)|$  is the Lebesgue constant for the interpolation points  $\mathcal{P}_n$ . Then the interpolation problem (9) has a unique solution in  $\mathbb{P}^n$  restricted to  $I$ .*

Note that except for the discrete locations of the jump-set topology changes, the maps  $\Phi(\cdot, \eta)$  can be expected to be continuous. Close to the topology changes, the Lipschitz constant will deteriorate, which is discussed at the end of this Section. In addition, we are only interested in  $\Phi(x, \eta)$  with  $x \in \Omega$ , so that the boundedness assumption can be met by suitable modifications of  $\Phi$  outside of  $\Omega$ . Therefore, with rather mild conditions the first part of the Proposition ensures the global existence of transforms which satisfy (9).

As argued at the beginning of this section, we expect non-unique solutions at the jump-set topology changes for the ODE (7). Likewise, we cannot expect global uniqueness for the discrete variant (9) either. Nonetheless, locally this is possible by classical arguments as stated by the second part of the proposition.

The proof is essentially a discrete version of the Picard-Lindelöf theorem. To see this more clearly and for later reference, by the uniqueness of polynomial interpolation, the problem (9) is equivalent to the following continuous variant: find  $x \in \mathbb{P}_n$  such that

$$X'(\eta) = [\mathcal{I}_{n-1}\Phi(X(\cdot), \cdot)](\eta) \text{ for } \eta \in \mathcal{P}, \quad X(\mu) = x, \quad (10)$$

or equivalently

$$X(\xi) = x + \int_{\mu}^{\xi} [\mathcal{I}_{n-1}\Phi(X(\cdot), \cdot)](\gamma) d\gamma =: F[X](\xi), \quad (11)$$

where  $\mathcal{I}_{n-1}$  is the interpolation operator on the  $n$  points  $\mathcal{P}_n$ . Note that this is not a standard ODE because the right hand side depends  $X(\gamma)$  for all  $\gamma \in \mathcal{P}_n$  instead of  $X(\eta)$  at the single point  $\eta$ . Nonetheless, the Picard-Lindelöf theorem carries over verbatim.

*Proof.* We use the usual fixed point argument applied to the fixed point function  $F$  defined in (11) on the space  $\mathbb{P}_n(I)$  of polynomials up to degree  $n$ , restricted to the interval  $I$  and equipped with the norm  $\|X\| := \max_{\gamma \in I} |X(\gamma)|$ . Due to the interpolation  $\mathcal{I}_{n-1}$  in the definition of  $F$  and the subsequent integration  $F$  maps  $\mathbb{P}_n(I)$  to itself.

We first show Part 1 of the proposition. Instead of the usual contraction mapping principle, we exploit the finite dimensional nature of  $\mathbb{P}_n$  and use Schauder's



fixed point theorem. To this end, note that the continuity of  $\Phi(\cdot, \eta)$  implies the continuity of  $F$ . In addition, we have

$$\begin{aligned}\|F[X]\| &= \max_{\xi \in I} \left| x + \int_{\mu}^{\xi} [\mathcal{I}_{n-1} \Phi(X(\cdot), \cdot)](\gamma) d\gamma \right| \\ &\leq |x| + \Lambda_{n-1} \|\Phi(X(\cdot), \cdot)\| \int_{\mu}^{\xi} d\gamma \\ &\leq |x| + \Lambda_{n-1} M |I|,\end{aligned}$$

where  $M$  is the upper bound of  $|\Phi(\cdot, \eta)|$ ,  $\eta \in \mathcal{P}_n$  and  $\Lambda_{n-1}$  is the Lebesgue constant for the interpolation  $\mathcal{I}_{n-1}$ . It follows that  $F$  maps the ball of all polynomials  $p \in \mathbb{P}_n$  with  $\|p\| \leq |x| + \Lambda_{n-1} M |I|$  to itself. In conclusion, Schauder's fixed point theorem implies the existence of a fixed point of the integral equation (11) and thus the discrete definition (9) of the transforms.

Since Schauder's fixed point theorem does not guarantee uniqueness, we continue to proof part 2 of the proposition using the classical contraction mapping principle. To show that  $F$  is a contraction, first note that

$$\begin{aligned}\|\mathcal{I}_{n-1} \Phi(X(\cdot), \cdot) - \mathcal{I}_{n-1} \Phi(Y(\cdot), \cdot)\| &\leq \Lambda_{n-1} \max_{\eta \in \mathcal{P}} |\Phi(X(\eta), \eta) - \Phi(Y(\eta), \eta)| \\ &\leq \Lambda_{n-1} L \|X - Y\|,\end{aligned}$$

where again  $\Lambda_{n-1}$  is the Lebesgue constant and  $L$  the Lipschitz constant of  $\Phi$ . It follows that

$$\|F[X] - F[Y]\| \leq \Lambda_{n-1} L \|X - Y\| \int_{\mu}^{\xi} d\gamma \leq \Lambda_{n-1} L |I| \|X - Y\|. \quad (12)$$

Since  $\Lambda_{n-1} L |I| < 1$ , the proposition follows from Banach's fixed point theorem.  $\square$

**Remark 4.2.** *Since  $X(\eta)$  is a polynomial, which is uniquely defined on the interval  $I$ , we may extend it to the real line  $\mathbb{R}$  still satisfying the discrete and continuous interpolation problems (9) and (10). However, beyond the interval  $I$  in Proposition 4.1 part 2, the solution is not necessarily unique in the sense that there can be two solutions  $x$  and  $y$  of (9) with different initial values  $X(\mu) = x$  and  $Y(\mu) = y$  that intersect outside of  $I$ .*

Let us compare the new transform  $X(\eta)$  constructed in the above Proposition with the original choice (2) defined via polynomial interpolation. The original choice is polynomial in  $\mu$ , whereas the “ideal” transform in the example of Section 3 is rational in  $\mu$ . In contrast, both the new transform and the “ideal” one are polynomial in  $\eta$ . In addition, the stability function of the implicit Euler method is a rational function of  $\eta - \mu$  and thus for a fixed interpolation node  $\eta$  and variable target  $\mu$  a rational function of  $\mu$ . Again, this matches the “ideal” transform of the example.

For a practical application of the new transform, we need an algorithm to solve the interpolation problem (9). Since the fixed point iteration for the map  $F : \mathbb{P}_n \rightarrow \mathbb{P}_n$  in the proof of Proposition 4.1 is defined on polynomials, it can be carried out practically and it remains to devise an algorithm to compute  $Y = F[X] \in \mathbb{P}_n$ . This is reminiscent of implicit ODE solvers as e.g. the implicit Euler method (8), which are also often solved by fixed point methods. We carry out the fixed point iteration in a Newton basis, which easily leads to an algorithm that is sufficient for the first experiments presented in this paper, although more clever approaches might be possible.

Although we integrate polynomials, we cannot readily apply standard quadrature rules because the integration bounds are variable and not known beforehand. As an alternative, we first differentiate the definition (11) of  $F$  to obtain

$$Y'(\xi) := \frac{d}{d\xi} F[X](\xi) = [\mathcal{I}_{n-1} \Phi(X(\cdot), \cdot)](\xi) \quad (13)$$

The derivative  $Y'(\xi)$  is a polynomial in  $\mathbb{P}_{n-1}$  and thus completely specified by its values at the interpolation points  $\eta \in \mathcal{P}_n$ , which are

$$Y'(\eta) = [\mathcal{I}_{n-1} \Phi(X(\cdot), \cdot)](\eta) = \Phi(X(\eta), \eta),$$

Of course these are the same interpolation conditions as in the original problem (9), with the difference that currently  $X$  is not necessarily a fixed point. Now, let us represent  $Y$  in a Newton basis with respect to the interpolation points  $\{\mu\} \cup \mathcal{P}_n = \{\mu, \eta_1, \dots, \eta_n\}$ , i.e.

$$Y(\xi) = \sum_{i=0}^n a_i \omega_i(\xi), \quad \omega_0(\xi) = 1 \quad \omega_i(\xi) = (\xi - \mu) \prod_{j < i} (\xi - \eta_j). \quad (14)$$

First note that choosing  $\xi = \mu$  the definition of  $F$  implies that  $a_0 = x$ . For the remaining  $a_1, \dots, a_n$ , we first differentiate  $Y'(\xi) = \sum_{i=1}^n a_i \omega'_i(\xi)$  and plug in the interpolation points to obtain the system

$$\begin{pmatrix} \omega'_1(\eta_1) & \dots & \omega'_n(\eta_1) \\ \vdots & & \vdots \\ \omega'_1(\eta_n) & \dots & \omega'_n(\eta_n) \end{pmatrix} \begin{pmatrix} a_1 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} \Phi(X(\eta_1), \eta_1) \\ \vdots \\ \Phi(X(\eta_n), \eta_n) \end{pmatrix}. \quad (15)$$

Note that for Newton interpolation we have  $\omega_j(\eta_i) = 0$  for  $j > i$ , which leads to triangular Vandermonde type matrices for standard interpolation problems. However, this does not imply that also  $\omega'_j(\eta_i) = 0$  for  $j > i$ . Therefore, in our case we deal with a full matrix. Nonetheless, we can compute an LU decomposition beforehand and solve the system efficiently for every new right hand side.

Now that we can compute  $Y = F[X]$ , we can numerically solve the interpolation problem (9) by the fixed point iteration

$$X^{k+1} = F[X^k], \quad X^0 = x. \quad (16)$$

The following Corollary to Proposition 4.1 states that this converges to a solution of the interpolation problem.

**Corollary 4.3.** *Assume that all assumptions of Proposition 4.1 part 2 are satisfied. Then the fixed point iteration (16) converges to a solution of the interpolation problem (9).*

*Proof.* The corollary is a direct consequence of the proof of Proposition 4.1 part 2. □

The fixed point iteration (16) provides us with polynomials  $\eta \rightarrow X(\eta; \mu, x)$  for  $\eta \in \mathcal{P}$ . For the TSI (1), we only need the transforms at the locations  $\eta \in \mathcal{P}_m$ , which we can easily evaluate even if  $\mathcal{P}_m \neq \mathcal{P}_n$ , i.e. we use different nodes for the transform field than for the snapshots.

As discussed earlier, we expect the Lipschitz constant  $L$  to deteriorate near jump-set topology changes so that a plain fixed point iteration as in the last corollary is generally insufficient. Nonetheless, we can counterbalance this with a small interval length  $I$  in order to meet the assumptions of Proposition 4.1. The obvious choice is to add more snapshots close to the singularity, but that clearly defeats the purpose of this section. Alternatively, we use this observation to create better initial values for the fixed point iteration. To this end, note that neither the definition of  $F[X]$  nor the assumptions of Proposition 4.1 need  $\Phi(\cdot, \eta)$  for the full continuum of parameters  $\eta$ , but only for the few discrete  $\eta \in \mathcal{P}_n$ . For clarity, let us call them  $\Phi_0, \dots, \Phi_n$ . Once we have fixed these functions, the interval length  $I$  only enters the equations via the Newton basis (14) for the interpolation points  $\{\mu\} \cup \mathcal{P}_n$ . Therefore, in order to obtain a shorter interval  $I$ , we rescale these interpolation points by

$$\{\mu\} \cup \{\mu + s(\eta - \mu) : \eta \in \mathcal{P}_n\} \quad (17)$$

which match the original choice with  $s = 1$  and shrink to  $\{\mu\}$  for  $s = 0$ . For  $s$  sufficiently small, the fixed point iteration (16) will then be successful for the unchanged  $\Phi_0, \dots, \Phi_n$ . We use the outcome as an improved initial value for a new fixed point iteration with larger scaling  $s$ , although this case may no longer be covered by Corollary 4.3. In the numerical experiments, we choose an increasing sequence of scaling factors  $s_i$  and perform fixed point iterations (16) for the corresponding scaled parameters, each using the last iterate from the previous scaling as initial value.

## 5 A note about stability

Since we are dealing with (near) singular jump-sets, we have to be careful about the stability of the TSI reconstruction with respect to perturbations of the transforms. Such perturbations are generally inevitable because of numerical errors due to iterative optimizers for (3) and discretizations of the transport fields  $\Phi$ .

Let us first consider a heuristic motivation in  $1d$ . Assume we have two transforms  $X_i(x) := X_i(\eta; \mu, x)$ ,  $i = 0, 1$ . Ignoring higher order terms, and

using the substitution  $y = X_0(x)$  we can informally estimate the perturbation error between two transformed snapshots  $v \circ X_i(x) := u(X_i(x), \eta)$  by

$$\begin{aligned}
\|v \circ X_1 - v \circ X_0\|_{L_1(\Omega)} &\approx \int_{\Omega} |u'(X_0(x))| |X_0(x) - X_1(x)| dx \\
&= \int_{\Omega} |u'(y)| |y - X_1(X_0^{-1}(y))| \frac{1}{|X_0'(x)|} dy \\
&\leq \sup_{y \in \Omega} \left[ \frac{|y - X_1(X_0^{-1}(y))|}{|X_0'(x)|} \right] \int_{\Omega} |u'(y)| dy \\
&= \sup_{x \in \Omega} \left[ \frac{|X_0(x) - X_1(x)|}{|X_0'(x)|} \right] \|u\|_{BV(\Omega)}.
\end{aligned} \tag{18}$$

If  $X_0'$  is bounded from above and below, this gives us an estimate of the type

$$\|v \circ X_1 - v \circ X_0\|_{L_1(\Omega)} \leq C \|X_0(x) - X_1(x)\|_{L_{\infty}(\Omega)} \|u\|_{BV(\Omega)}.$$

for some constant  $C$ . However, this argument is not fully satisfactory for two reasons.

1. In the vicinity of jump-set topology changes  $X_0'(x)$  is typically not uniformly bounded from above and below.
2. If  $X_i$  has a jump or sharp gradient, reasonable perturbations may not be close in the  $L_{\infty}$ -norm.

If we are more careful, we see that wherever  $X_0$  has a large gradient so that we possibly have a large  $L_{\infty}$  error of the transforms, the denominator of the right hand side of (18) is also large. The argument does not carry over to sharp gradients of  $X_1$ , but with an analogous argument, we could have had  $X_1'$  in the denominator instead. Therefore, in the following, we consider the question if a judicious choice of the denominator can at least in principle provide some control of the perturbation error in the face of sharp gradients.

The choice of the denominator, now for arbitrary spacial dimensions, is done by selecting a homotopy, i.e. a continuous transition,  $X_s(\eta; \mu, x)$ ,  $0 \leq s \leq 1$  between the two transforms so that the boundary cases  $X_0(\eta; \mu, x)$  and  $X_1(\eta; \mu, x)$  are our original two perturbed transforms. A simple candidate is a convex combination  $X_s = (1-s)X_0 + sX_1$ . The following stability result is a rigorous version of our heuristic argument above and a slight refinement from a proposition in [47].

**Proposition 5.1.** *Assume that  $u \in BV(\Omega)$  and that*

$$\int_{X_s(\eta; \mu, \cdot)^{-1}(A)} dx \leq B|A|$$

*for some  $B \geq 0$ , all measurable sets  $A$ , all  $\eta \in \mathcal{P}_n$ ,  $\mu \in \mathcal{P}$  and  $s \in \{0, 1\}$ . Using  $\dot{X} := \frac{d}{ds} X_s$ , define*

$$S(\mu, \eta) := \sup_A \frac{1}{|A|} \int_0^1 \int_{X_s(\eta; \mu, \cdot)^{-1}(A)} |\dot{X}_s(\eta; \mu, x)| dx ds. \tag{19}$$

Then, we have

$$\|u_m(\cdot, \mu; X_0) - u_m(\cdot, \mu; X_1)\|_{L_1(\Omega)} \leq \Lambda_n \max_{\eta \in \mathcal{P}_m} [S(\mu, \eta) \|u(\cdot, \eta)\|_{BV(\Omega)}],$$

independent of  $B$  and where  $\Lambda_m$  is the Lebesgue constant.

The proof is essentially the same as in [47] and given in Appendix A. For sufficiently regular transforms  $X_s$ , we can simplify the stability factor  $S(\mu, \eta)$ : With  $X_s(x) = X_s(\eta; \mu, x)$  we have

$$\begin{aligned} S(\mu, \eta) &= \sup_A \frac{1}{|A|} \int_0^1 \int_{X_s^{-1}(A)} |\dot{X}_s(x)| dx ds \\ &= \sup_A \int_0^1 \frac{1}{|A|} \int_A |\dot{X}_s(X_s^{-1}(y))| |\det DX_s^{-1}(y)| dy ds \\ &= \int_0^1 \sup_{y \in \Omega} |\dot{X}_s(X_s^{-1}(y))| |\det DX_s^{-1}(y)| ds \\ &= \int_0^1 \sup_{x \in \Omega} \frac{|\dot{X}_s(x)|}{|\det DX_s(x)|} ds. \end{aligned} \tag{20}$$

Note the similarity to our informal calculation (18): For the nominator, we have  $|\dot{X}_s(x)| \approx |X_0(x) - X_1(x)|$  and in 1d the denominator simplifies to  $|X'_s(x)|$ . In addition, because of the  $s$  dependence, we now have some explicit control over the denominator by choosing a suitable homotopy.

In order to better understand the stability factor  $S(\mu, \eta)$ , let us first consider the integral bound  $X_s(\eta; \mu, \cdot)^{-1}(A)$  in (19), or equivalently the denominator in (20). To this end, we consider a set  $A \subset \Omega$  and observe how its volume changes when it is transformed along the flow  $\eta \rightarrow A_\eta := X_s(\eta; \mu, A)$ . We expect the stability factor to be small if the volume  $|A_\mu| = |X_s(\eta; \mu, \cdot)^{-1}(A_\eta)|$  is not much larger than the volume  $|A_\eta|$ . This implies that contracting sets from  $\eta \rightarrow \mu$  is expected to be stable, while growing them is probably unstable.

To clarify this point, let us consider the example in Figure 1 again. Say we have two parameters  $\mu_c$ , where the two jumps collide, and  $\mu_2 < \mu_c$  for which we have two separate jumps. Let us first try to approximate  $u(\cdot, \mu_c)$  form a transformation of  $u(\cdot, \mu_2)$ . If we choose any set  $A_{\mu_2}$  between the two jumps, the transform indicated in the figure shrinks it to zero volume. By our analysis this is stable and intuitively all perturbations in the set  $A_{\mu_2}$  are squashed to measure zero and thus irrelevant. Vice versa, using a transform of the snapshot  $u(\cdot, \mu_c)$  to approximate  $u(\cdot, \mu_2)$  is problematic. Our stability bound becomes large matching the intuition that the snapshot does not provide any information on how to fill the void between the two jumps of the target  $u(\cdot, \mu_2)$ .

Let us next consider the problem with the  $L_\infty$ -norm for a simple example problem. We choose  $X_s(\eta; \mu, x) = X_0(\eta; \mu, x - s\epsilon v)$  for some direction  $v$ , so that the perturbation  $X_1$  is a shift of  $X_0$ , which generally has large  $L_\infty$  error if  $X_0$  has sharp gradients or jumps. Note that with this definition we also have chosen a homotopy between the two transforms. Next, we assume that

$\|DX_s(\eta; \mu, \cdot)^{-1}\|_2 \leq c$  for some constant  $c$ . This allows the transform  $X_s(\eta; \mu, \cdot)$  to have sharp gradients and places us in the situation where the sets  $A_\eta$  grow at most moderately for which we expect stable behaviour by the discussion above. In the example of Figure 1, this includes the case of approximating  $u(\cdot, \mu_c)$  from  $u(\cdot, \mu_2)$  using a perturbed transform that misses the correct collision location in physical space by  $\epsilon$ .

Computing the stability factor (20) yields

$$S(\mu, \lambda) = \int_0^1 \sup_{x \in \Omega} \frac{|\dot{X}_s(x)|}{|\det DX_s(x)|} ds \leq \int_0^1 \sup_{x \in \Omega} s \epsilon \frac{|DX_s(x)|_2 |v|}{|\det DX_s(x)|} ds.$$

where  $|\cdot|_2$  denotes the  $\ell_2(\mathbb{R}^d)$  matrix norm. With the singular values  $\sigma_1^s(x) \geq \dots \geq \sigma_d^s(x) \geq c$  of  $DX_s$ , this simplifies to

$$S(\mu, \lambda) = \int_0^1 s \epsilon |v| \sup_{x \in \Omega} \frac{\sigma_1^s(x)}{\prod_{i=1}^d \sigma_i^s(x)} ds \leq \epsilon |v| c^{1-d} \int_0^1 s ds = \frac{1}{2} \epsilon |v| c^{1-d}.$$

Thus, even if the  $L_\infty$  error of the perturbed transforms is large, we may nonetheless have a small TSI error. Of course the result relies on a carefully chosen homotopy  $X_s$ . How to choose this in the general case is beyond the scope of this paper.

## 6 Optimizing Singular Transforms

### 6.1 Difficulties with the optimization of the transform

In Section 4, we discussed singularities of the transforms at jump-set topology changes and efficient ways to approximate them. However, we still need a way to automatically calculate these transforms from the available data, i.e. the snapshots and training snapshots. As in (3), we use an optimization problem

$$\min_{\Phi \in \mathcal{T}} \sigma(\Phi), \quad \sigma(\Phi) := \|u(\cdot, \mu) - u_m(\cdot, \mu; \Phi)\|_{L_1(\Omega)}, \quad (21)$$

where the notation  $u_m(x, \mu; \Phi) = u_m(x, \mu)$  is used to emphasize the dependence of the TSI on the choice of the transport field  $\Phi$  and  $\mathcal{T} \subset \mathcal{B}$  is an admissible set of transforms in a Banach space  $\mathcal{B}$  as discussed below. Note that the constraints  $\mathcal{T}$  can be used to ensure that the resulting curves  $X(\eta)$  do not leave the spacial domain  $\Omega$  and that we only need to know  $\Phi(\cdot, \eta)$  for finitely many  $\eta \in \mathcal{P}_n$ . We therefore only need a spacial discretization of  $\Phi$  by selecting a basis, or more generally a dictionary,  $\{\psi_i\}_{i \in \mathcal{I}}$  for some index set  $\mathcal{I}$ , with  $\text{span}\{\psi_i\}_{i \in \mathcal{I}} = \mathcal{B}_n \subset \mathcal{B}$  and optimize

$$\min_{c \in \mathbb{R}^N} \sigma(c), \quad \sigma(c) := \sigma\left(\sum_{i=1}^N c_i \psi_i\right). \quad (22)$$

Instead of strictly enforcing that  $x \rightarrow X(\eta; \mu, x)$  maps  $\Omega$  to itself, we usually assume that  $\nu \cdot \Phi(\cdot, \eta) = 0$ , where  $\nu$  is the outward normal, so that locally the

curves  $X(\eta; \mu, x)$  do not flow through the boundary. As we have seen above, already simple problems require sharp spacial gradients so that we want to choose  $\mathcal{B}_n$  with resolutions matching the high fidelity model of the snapshots. In this section, we discuss the implications for the optimization of (21).

It has been shown in [47] that we can expect the map  $\sigma$  to be Lipschitz continuous. Although more sophisticated algorithms for such non-smooth optimization problems are available [24, 6], we start with a simple gradient descent method:

$$\mathbf{c}^{n+1} = \mathbf{c}^n - \alpha \sigma'(\mathbf{c}^n). \quad (23)$$

Unfortunately, this simple choice does not provide satisfactory results. For example, Figure 2 shows the convergence behaviour for a very simple one dimensional test case: The gradient descent optimizer reduces the error a little bit but then stalls way before any reasonable alignment has been achieved. Since the example has no singularities, the figure also shows the results for a coarse discretization of  $X(\eta; \mu, x)$  by second order polynomials in  $x$ . As we see, this coarse resolution does not pose a severe obstacle to the optimizer.

## 6.2 Optimization in Banach Spaces

To better understand the issue of the higher resolution, we first go all the way to infinite resolution and optimize (21) over the full function space  $\mathcal{B}$ . We solve the optimization problem  $\min_{\Phi \in \mathcal{B}} \sigma(\Phi)$  by a descent method so that

$$\Phi^{n+1} = \Phi^n - \alpha d^n$$

for some descent direction  $d^n$ . Usually, we choose the gradient  $d^n = \sigma'(\Phi)$ , but this time we have to be more careful: Since we optimize in a Banach space,  $\sigma'(\Phi)$  is a Fréchet derivative and thus an element of the dual space  $\mathcal{B}'$  of  $\mathcal{B}$ . If  $\mathcal{B} = \mathbb{R}^n$ , we can easily identify  $(\mathbb{R}^n)' = \mathbb{R}^n$ , but for a generic Banach space  $\mathcal{B}$  this is not always possible. This implies that a term  $\Phi^n - \alpha \sigma'(\Phi)$  does not necessarily make sense.

Thus, the descent direction  $d^n$  depends on the Banach space  $\mathcal{B}$ . A good choice of  $\mathcal{B}$  should depend e.g. on the existence of minimizers or their stability, which is beyond the scope of this paper. Nonetheless, one can easily gain some insight about the space  $\mathcal{B}'$ . Formally, the variation of the error in the direction  $v \in \mathcal{B}$  is given by

$$\partial_v \sigma(\Phi) = \int_{\Omega} \text{sign} [u(x, \mu) - u_m(x, \mu; \Phi)] \partial_v u_m(x, \mu; \Phi) dx$$

where using the definition of the TSI, we have

$$\partial_v u_m(x, \mu; \Phi) = \sum_{\eta \in \mathcal{P}_m} \ell_{\eta}(\mu) Du(X(\eta; \mu, x), \eta) DX(\eta; \mu, x) v$$

Note in particular that this derivative contains the derivative  $Du(x, \eta)$  with respect to  $x$ . Since  $u$  has jumps we may assume that  $u$  is of bounded variation so

that  $Du$  is a measure. At the jumps this measure has a singular component, e.g. a Dirac delta in  $d = 1$  or generally a measure supported on a  $d - 1$  dimensional manifold. This implies that the dual space  $\mathcal{B}'$  must be rich enough to contain these singular measures, which makes an identification of  $\mathcal{B}$  and  $\mathcal{B}'$  difficult.

Of course the optimization in Banach spaces is well understood and a descent direction  $d^n$  can be constructed along the following lines, see e.g. [45]. First observe that the value of the objective function after one step is given by

$$\sigma(\Phi^{n+1}) = \sigma(\Phi^n) - \alpha \langle \sigma'(\Phi^n), d^n \rangle + o(\alpha)$$

so that we obtain an error reduction for sufficiently small  $\alpha$  if  $\langle \sigma'(\Phi^n), d^n \rangle > 0$ . For example, we can choose the Riesz representation of the Fréchet defined by

$$d^n = \|\sigma(\Phi^n)\|_{\mathcal{B}'} w, \quad w = \operatorname{argmax}_{v \in \mathcal{B}} \frac{\langle \sigma'(\Phi^n), v \rangle}{\|v\|_{\mathcal{B}}}, \quad (24)$$

so that  $\langle \sigma'(\Phi^n), d^n \rangle = \|\sigma'(\Phi^n)\|_{\mathcal{B}'}^2$  and we obtain

$$\sigma(\Phi^{n+1}) = \sigma(\Phi^n) - \alpha \|\sigma'(\Phi^n)\|_{\mathcal{B}'}^2 + o(\alpha).$$

Recall that in our case the dual space  $\mathcal{B}$  should contain singular measures supported on  $d - 1$  dimensional manifolds. Via the trace theorem, measures of such type are contained in  $H^{-1}(\Omega)$ , which is a particularly simple choice because it is a Hilbert space and the computation of the Riesz representer reduces to solving the Poisson equation. More precisely, since  $\Phi(\cdot, \eta) : \Omega \rightarrow \mathbb{R}^d$  is vector valued, we choose  $\mathcal{B} = \prod_{\eta \in \mathcal{P}_m} \{v \in H^1(\Omega)^d : \nu \cdot v = 0 \text{ on } \partial\Omega\}$ , where  $\nu$  is the outward unit normal vector and the “slip” boundary conditions ensure that locally the resulting curves  $X(\eta)$  do not leave  $\Omega$ . The product is used to obtain one function for each  $\Phi(\cdot, \eta)$  with  $|\mathcal{P}_n| = n$  values for  $\eta$ . Thus, the Riesz representation (24) is given by  $-\Delta^{-1} \sigma'(\Phi^n)$  and we obtain the optimization scheme

$$\Phi^{n+1} = \Phi^n + \alpha \Delta^{-1} \sigma'(\Phi^n), \quad (25)$$

where  $-\Delta^{-1}$  solves the vector valued Poisson equation with the given boundary conditions. This is discretized by a Galerkin method in the obvious way. We will refer to this method as *Laplace smoothing* in the following.

As an alternative to inverting a Laplacian, we can carefully choose the dictionaries  $\{\psi_i\}_{i \in \mathcal{I}}$  for the discretization, similar to e.g. wavelet methods for elliptic equations [13, 10]. To this end, let us assume that the  $\psi_i$  are a frame for  $\mathcal{B}'$ , so that

$$A \|\ell\|_{\mathcal{B}'}^2 \leq \sum_{i \in \mathcal{I}} |\langle \ell, \psi_i \rangle| \leq B \|\ell\|_{\mathcal{B}'}^2$$

for all  $\ell \in \mathcal{B}'$  and constants  $A, B \geq 0$ . Then, there is a dual frame  $\{\bar{\psi}_i\}_{i \in \mathcal{I}}$  such that

$$B^{-1} \|\Phi\|_{\mathcal{B}}^2 \leq \sum_{i \in \mathcal{I}} |\langle \Phi, \bar{\psi}_i \rangle| \leq A^{-1} \|\Phi\|_{\mathcal{B}}^2$$

for all  $\Phi \in \mathcal{B}$  and

$$\Phi = \sum_{i \in \mathcal{I}} \langle \Phi, \bar{\psi}_i \rangle \psi_i.$$



Note that  $\Phi \in \mathcal{B}$  if and only if the frame coefficients  $c_i = \langle \Phi, \bar{\psi}_i \rangle$  are in  $\ell_2(\mathbb{R}^{\mathcal{I}})$  so that in the original discretization (22) we now optimize over  $\mathbf{c} \in \ell(\mathbb{R}^{\mathcal{I}})$  for which we easily can identify the space with its dual. Indeed, for the resulting iteration

$$\mathbf{c}^{n+1} = \mathbf{c}^n - \alpha \mathbf{d}^n, \quad \mathbf{d}_i^n = \partial_i \sigma(\mathbf{c}) = \left\langle \sigma' \left( \sum_{i=1}^{\infty} c_i \psi_i \right), \psi_i \right\rangle \quad (26)$$

we have the error reduction

$$\begin{aligned} \sigma(\mathbf{c}^{n+1}) &= \sigma(\mathbf{c}^n) - \alpha \sum_{i=1}^{\infty} \left| \left\langle \sigma' \left( \sum_{i=1}^{\infty} c_i \psi_i \right), \psi_i \right\rangle \right|^2 + o(\alpha) \\ &\leq \sigma(\mathbf{c}^n) - \alpha A \left\| \sigma' \left( \sum_{i=1}^{\infty} c_i \right) \right\|_{\mathcal{B}'}^2 + o(\alpha) \end{aligned}$$

for  $\alpha$  sufficiently small.

For simplicity, in the numerical experiments, we do not explicitly construct a frame, but rather define the transform as  $X(\eta; \mu, x) = x + \sum_{\ell} (X^{\ell}(\eta; \mu, x) - x)$ , where  $X^{\ell}(\dots)$  are curves generated by transform fields  $\Phi^{\ell}$  with spacial resolution level  $\ell$  and  $X^{\ell}(\dots) - x$  the updates from the respective levels. This is referred to as *multilevel smoothing* in the following.

## 7 Numerical Experiments

In Section 7.1, we first choose a simple problem without singularity to see how the optimizers of Section 6 perform in comparison to the original method recalled in Section 2. Then, in Section 7.2, we consider a test with a singularity where the original low resolution transforms fail. In Section 7.3, we compare different orders of the transforms and in Section 7.4 we consider a  $2d$  example.

In all examples, the finite element discretizations of the snapshots and transform fields are handled by the Deal.II library [3] and all TSI components are then put together in Tensorflow [1]. This setup allows all gradients of the error to be automatically computed by the back-propagation algorithm in Tensorflow.

### 7.1 Testing the Optimizers

In this section, we first consider a very simple example in order to see how the optimizers of Section 6 applied to high resolution transforms of Section 4 perform in comparison to a simple gradient descent method and low resolution transforms as described in [47] or Section 2. To this end, we consider

$$u(x, \mu) = \begin{cases} 1 & x \leq \mu \\ -1 & x > \mu \end{cases} \quad \begin{aligned} \mathcal{P}_m &= \mathcal{P}_n = \{-0.2, 0.2\} \\ \mathcal{P}_T &= \{0\} \end{aligned} \quad \Omega = [-1.5, 1.5].$$

The snapshots and high resolution transforms are discretized by piecewise linear functions on 32 cells and the low resolution transforms are second order polynomials. We use five fixed point iterations (16) and no scaling (17).

Since the objective function typically has a kink at the minimum, its value from gradient descent steps typically oscillates near the minimum. For all optimization methods in this section, we select a learning rate that shows slight oscillations at the end to make them comparable. This is not very accurate but sufficient to demonstrate the main point of this experiment: Despite the much higher number of degrees of freedom, the Laplace and multilevel smoothing result in a convergence behaviour that is comparable to the original gradient descent method applied to low resolution transforms in [47].

The concrete results are reported in Figure 2. First observe that a simple gradient descent method without smoothing for high resolution transforms stalls way before it reaches an acceptable minimum. Second, the value of the objective function never converges to zero. Instead, the TSI training error typically saturates at a level comparable to the discretization error of the snapshots. This effect will be observed for all numerical experiments in this paper and is discussed more thoroughly in [47].

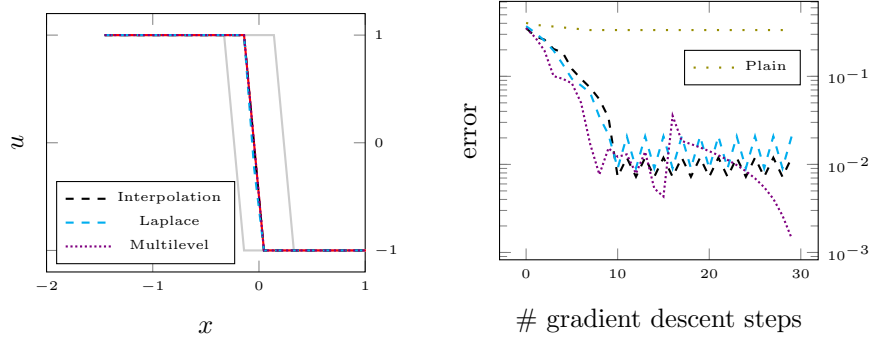


Figure 2: Results for the 1d Example in Section 7.1. Left: Snapshots (gray), exact solution (solid red) and reconstructions at  $\mu = 0$ : TSI with Laplace smoothing (cyan, dashed), multilevel smoothing (violet, dotted) and a TSI with low resolution transform (2) (black, dashed). Right: corresponding gradient descent errors with the same color coding. In addition the figure shows the errors for high resolution transforms without smoothing (olive dotted).

## 7.2 Testing Transform Resolutions

Next, we test high versus low resolution transforms near a singularity. We choose the example

$$u(x, \mu) = \begin{cases} \frac{x-l}{-(1-\mu)-l} & x \leq -(1-\mu) \\ \frac{x-r}{(1-\mu)-r} & x \geq 1-\mu \\ 0 & \text{else} \end{cases}, \quad \begin{matrix} \mathcal{P}_m = \{0.6\} \\ \mathcal{P}_n = \{0.6\} \\ \mathcal{P}_T = \{0.9\} \end{matrix} \quad \begin{matrix} \Omega = [-1.5, 1.5] \\ = [l, r] \end{matrix},$$

with snapshots and high resolution transforms discretized by piecewise linear functions on 128 cells. For the low resolution transforms we choose  $x \rightarrow X(\eta; \mu, x)$  as third order polynomials. They are defined by four values: Two are used to ensure that the boundary positions do not move and two are optimized to track the two jumps. We only compare the low and high resolution transforms at the training parameter  $\mathcal{P}_T$  because by (2) the low order transform is independent of the target  $\mu$  for reconstructions from one single snapshot. For the high resolution transforms, we use three fixed point iterations in (16) and scaling  $s \in \{0.7^i : i = 0 \dots 3\}$  in increasing size.

In Figure 3 left, we see the TSI reconstructions at  $\mathcal{P}_T$ . We see that the high resolution transforms provide a good reconstruction, whereas the low resolution transforms do not. We have chosen the degree of the low resolution transforms sufficiently high so that in principle they can align the jumps. However, this requires a rather sharp gradient, which would create a big oscillation in the polynomial. As a result, the composition  $x \rightarrow u(X(\eta; \mu, x), \eta)$  cannot be linear in the regions where is exact solution  $x \rightarrow u(\mu, x)$  is. This can be seen by carefully looking at the plot in the linear regions. Thus, the optimizer must balance the fit in the linear regions with the alignment, which is not possible to do accurately. In contrast, by Figure 3 right, the high resolution transforms are almost piecewise linear with kinks at the jump locations. Therefore they can simultaneously align the jumps and provide an accurate fit in the linear regions. The effect can also be clearly seen in the training errors in Figure 4 left, where the low resolution transforms saturate much earlier than the high resolution transforms.

## 7.3 Testing Transform Orders

In this section, we compare different polynomial orders in  $\eta$  of the transforms. To this end, we choose the example

$$u(x, \mu) = \begin{cases} \frac{x-l}{-p(\mu)-l} & x \leq -p(\mu) \\ \frac{x-r}{p(\mu)-r} & x \geq p(\mu) \\ 0 & \text{else} \end{cases}, \quad \begin{matrix} \mathcal{P}_m = \{0.6\} \\ \mathcal{P}_n = \{0.6, 0.7\} \\ \mathcal{P}_T = \{0.8, 0.96\} \end{matrix} \quad \begin{matrix} \Omega = [-1.5, 1.5] \\ = [l, r] \end{matrix},$$

where  $p(\mu)$  is a second order polynomial given by  $p(0.6) = 0.4$ ,  $p(0.8) = 0.3$  and  $p(1) = 0$ . All the other parameters are set as in the example of Section 7.2. Note in particular that the jump-set is now a parabola in parameter so that a first

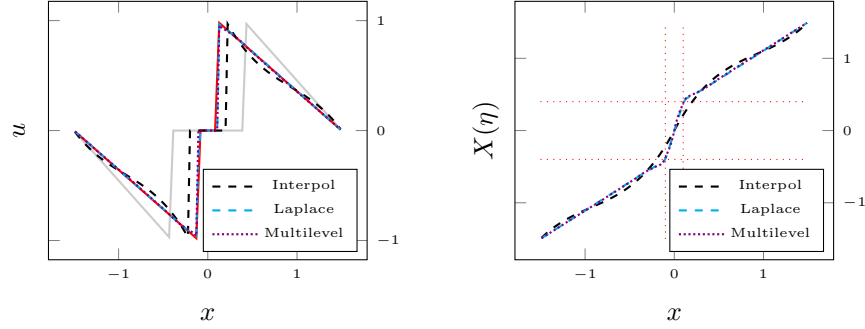


Figure 3: Results for the Example in Section 7.2. Left: Snapshots (gray), exact function at  $\mathcal{P}_T$  (red) and TSI with low resolution (black dashed) and high resolution with Laplace smoothing (cyan, dashed) and multilevel smoothing (violet, dotted). Right: The corresponding transform with the same color coding. The red lines are the locations of the jumps of the training snapshot (vertically) and the target snapshots (horizontally).

order reconstruction as in the last section can no longer accurately recover  $u$ . For a comparison, we choose Laplace smoothing once with first order transforms (in  $\eta$ ) with  $\mathcal{P}_n = \{0.6\}$  and once with second order with  $\mathcal{P}_n = \{0.6, 0.7\}$ . To be fair, both are trained on the same training set  $\mathcal{P}_T$  as given above.

The results for  $\mu = 0.86$  are shown in Figure 5. In the function plot the second order reconstruction looks marginally better, but a look at the training errors reveals that they are better by a factor of about 6. The final training error for the second order case is about twice as much as for the example of Section 7.2. This is encouraging because the profiles are identical, only the dynamic in parameter is changed, and in the second order example we add the training error at two training parameters, whereas we only use one in the previous example.

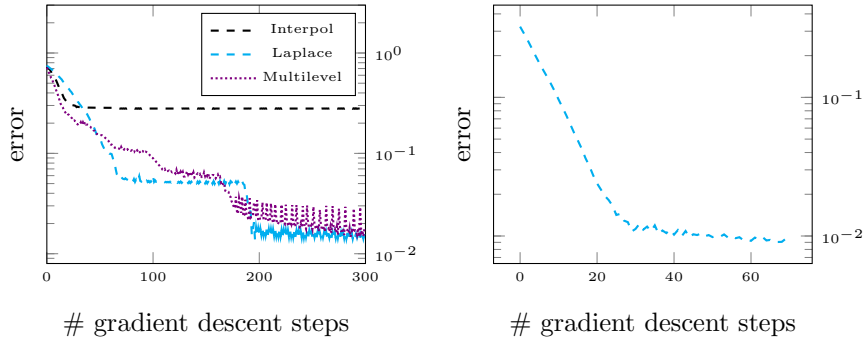


Figure 4: Gradient descent errors for the examples in Sections 7.2 and 7.4. The left picture uses the color coding of Figure 3.

In principle, at this place we should present some plots of convergence rates. However, these are difficult to achieve because of the error saturation. Even if we achieve high order reconstructions, the errors need to be matched by the discretizations of the snapshots in physical space. Since this is quite difficult because of the jumps, we omit a more thorough convergence analysis.

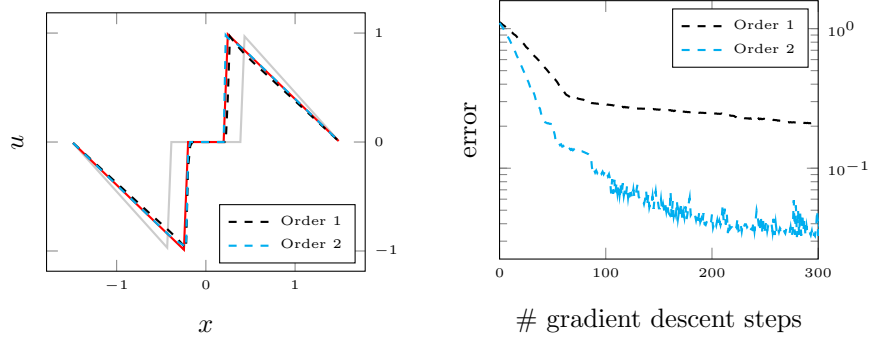


Figure 5: Results for the Example in Section 7.3. Left: Snapshots (gray), exact function at  $\mathcal{P}_T$  (red) and TSI reconstruction with linear in  $\eta$  (black dashed) and quadratic (cyan dashed) transforms. Right: The corresponding gradient descent errors.

#### 7.4 2d Example

Finally, we consider one example in two spacial dimensions. We use the function

$$u(x, y, \mu) = \begin{cases} 1 & \left(\frac{r}{2\mu}\right)^2 + \left(\frac{s}{\mu}\right)^2 \leq 1 \\ 0 & \text{else} \end{cases}, \quad \begin{pmatrix} r \\ s \end{pmatrix} = \begin{pmatrix} \cos(1.5\mu) & -\sin(1.5\mu) \\ \sin(1.5\mu) & \cos(1.5\mu) \end{pmatrix}$$

which is a rotated and scaled ellipse, with parameters

$$\begin{aligned} \mathcal{P}_m &= \{c(1), c(0.3)\} \\ \mathcal{P}_n &= \{c(1)\} & c(\lambda) &= 0.2\lambda + 0.05(1 - \lambda) & \Omega &= [-1, 1]^2. \\ \mathcal{P}_T &= \{c(0.6), c(0)\} \end{aligned}$$

The snapshots and high resolution transforms are discretized by piecewise bilinear functions on a rectangular grid with  $64 \times 64$  cells. We use three fixed point iterations in (16) and scaling  $s \in \{0.7^i : i = 0 \dots 2\}$  in increasing size.

Figure 6 shows a solution plot for  $\mu = c(0.25)$  together with some of the curves  $\eta \rightarrow X(\eta; \mu, x)$  originating on the boundary of the correct ellipse. Figure 4 right contains the corresponding training errors. We see that also in this 2d example, the method can achieve a good alignment with a moderate number of gradient descent steps.

As a final remark, note that for all experiments we use a fixed step size of a conservative magnitude. This is necessary because of the non-smooth nature

of the objective function. Normally, near a minimum, the gradient becomes small so that the update between two gradient descent steps converges to zero. In our case however, we expect a kink at the minimum, similar to minimizing an absolute value  $|x|$ . Therefore, the gradient does not become small, which is compensated by a small step size. The effect can be seen in the convergence plots, where in the end the error oscillates around an equilibrium with amplitude depending on the step size. On the other hand, far away from the minimum, the conservative choice of the step size hinders a faster convergence.

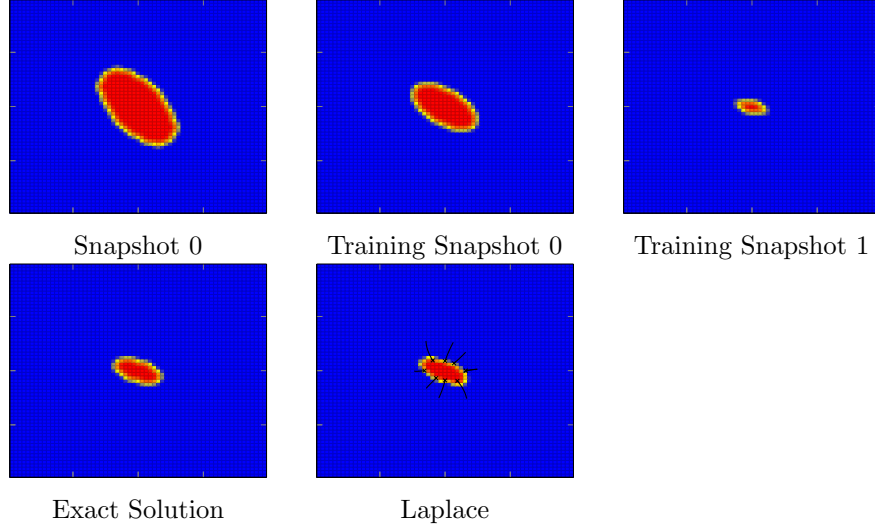


Figure 6: Results for numerical tests in Section 7.4. The top row contains the input data and the bottom row the exact solution and TSI reconstruction. The black lines in the last plot indicate several transforms  $\eta \rightarrow X(\eta; \mu, x)$  with initial values  $x$  at the respective black  $\times$ s.

## Appendix A Proof of Proposition 5.1

The following proof of Proposition 5.1 is almost verbatim from [47, Section 3].

Let us first estimate  $\|v \circ X_0 - v \circ X_1\|_{L^1(\Omega)}$  for  $X_s(x) := X_s(\eta, \mu, x)$  for some fixed  $\mu$  and  $\eta$  and some function  $v \in C^1(\Omega)$ . Applying the fundamental theorem for line integrals, we obtain

$$(v \circ X_0)(x) - (v \circ X_1)(x) = \int_0^1 v'(X_s(x)) \dot{X}_s(x) ds.$$

Note that we can regard  $|\dot{X}(s)|$  as a density and thus define a transition kernel  $\tau(s, A) = \int_A |\dot{X}_s(x)| dx$ . Then, with the pushforward  $\kappa(s, A) = (X_s)_* \tau(s, A) = \tau(s, X_s^{-1}(A))$  and the Lebesgue measure  $\lambda_1$  on the real line, the definition of

$S(\mu, \lambda)$  implies

$$(\lambda_1 \otimes \kappa)([0, 1] \times A) = \int_0^1 \kappa(s, A) ds = \int_0^1 \int_{X_s^{-1}(A)} |\dot{X}_s(x)| dx ds \leq S(\mu, \eta)|A|.$$

Thus, using the Fubini theorem for transition kernels we obtain

$$\begin{aligned} & \|v \circ X_0 - v \circ X_1\|_{L_1(\Omega)} \\ & \leq \int_{\Omega} \int_0^1 |v'(X_s(x))| |\dot{X}_s(x)| ds dx = \int_0^1 \int_{\Omega} |v'(X_s(x))| \tau(s, dx) ds \\ & = \int_0^1 \int_{\Omega} |v'(y)| \kappa(s, dy) ds = \int_{[0, 1] \times \Omega} |v'(y)| d(\lambda_1 \otimes \kappa)(s, y) \quad (27) \\ & \leq S(\mu, \eta) \int_{\Omega} |v'(y)| dy = S(\mu, \eta) \|v\|_{BV(\Omega)}. \end{aligned}$$

Next, we extend this estimate from functions  $v \in C^1(\Omega)$  to functions  $v \in BV(\Omega)$  by a density argument. To this end, note that for all  $\epsilon > 0$  there is a  $v_{\epsilon} \in C^1(\Omega)$  such that

$$\|v - v_{\epsilon}\|_{L_1(\Omega)} \leq \epsilon \quad \|v'_{\epsilon}\|_{L_1(\Omega)} \leq \|v\|_{BV(\Omega)} + \epsilon.$$

Thus, to apply a density argument, it suffices to bound  $\|v \circ X_0 - v_{\epsilon} \circ X_0\|_{L_1(\Omega)}$  and  $\|v \circ X_1 - v_{\epsilon} \circ X_1\|_{L_1(\Omega)}$ . Since by assumption  $(X_s)_* \lambda(A) \leq B\lambda(A)$  for all measurable sets  $A$ , we conclude that

$$\begin{aligned} \|v \circ X_0 - v_{\epsilon} \circ X_0\|_{L_1(\Omega)} &= \int_{\Omega} |v(y) - v_{\epsilon}(y)| d(X_0)_* \lambda(y) \\ &\leq B \|v - v_{\epsilon}\|_{L_1(\Omega)} \end{aligned}$$

The bound for  $\|v \circ X_1 - v_{\epsilon} \circ X_1\|_{L_1(\Omega)}$  follows analogously.

Finally, recalling that  $X_s(x) = X_s(\eta; \mu, x)$ , we apply the estimate (27) to the full TSI and obtain

$$\begin{aligned} & \|u_m(\cdot, \mu; X_0) - u_m(\cdot, \mu, X_1)\|_{L_1(\Omega)} \\ & \leq \sum_{\eta \in \mathcal{P}_m} |\ell_{\eta}(\mu)| \|u(X_0(\eta; \mu, \cdot), \eta) - u(X_1(\eta; \mu, \cdot), \eta)\|_{L_1(\Omega)} \\ & \leq \Lambda_n \max_{\eta \in \mathcal{P}_m} \|u(X_0(\eta; \mu, \cdot), \eta) - u(X_1(\eta; \mu, \cdot), \eta)\|_{L_1(\Omega)} \\ & \leq \Lambda_n \max_{\eta \in \mathcal{P}_m} [S(\mu, \eta) \|u(\cdot, \eta)\|_{BV(\Omega)}]. \end{aligned}$$

## References

- [1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp,

- G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.
- [2] R. Abgrall and D. Amsallem. Robust model reduction by  $l_1$ -norm minimization and approximation via dictionaries: Application to linear and nonlinear hyperbolic problems. <http://arxiv.org/abs/1506.06178>, 2015.
- [3] W. Bangerth, R. Hartmann, and G. Kanschat. Deal.ii – a general-purpose object-oriented finite element library. *ACM Trans. Math. Softw.*, 33(4), 2007.
- [4] P. Benner, S. Gugercin, and K. Willcox. A survey of projection-based model reduction methods for parametric dynamical systems. *SIAM Review*, 57(4):483–531, 2015.
- [5] J. Brunken, K. Smetana, and K. Urban. (parametrized) first order transport equations: Realization of optimally stable petrov-galerkin methods. 2018.
- [6] J. Burke, A. Lewis, and M. Overton. A robust gradient sampling algorithm for nonsmooth, nonconvex optimization. *SIAM Journal on Optimization*, 15(3):751–779, 2005.
- [7] N. Cagniard, R. Crisovan, Y. Maday, and R. Abgrall. Model order reduction for hyperbolic problems: a new framework. 2017.
- [8] N. Cagniard, Y. Maday, and B. Stamm. Model Order Reduction for Problems with large Convection Effects. In B. N. Chetverushkin, W. Fitzgibbon, Y. Kuznetsov, P. Neittaanmäki, J. Periaux, and O. Pironneau, editors, *Contributions to Partial Differential Equations and Applications*. 2019. <http://hal.upmc.fr/hal-01395571>.
- [9] Q.-Y. Chen, D. Gottlieb, and J. S. Hesthaven. Uncertainty analysis for the steady-state flows in a dual throat nozzle. *Journal of Computational Physics*, 204(1):378 – 398, 2005.
- [10] A. Cohen. *Numerical Analysis of Wavelet Methods*. JAI Press, 2003.
- [11] A. Cohen and R. DeVore. Approximation of high-dimensional parametric pdes. *Acta Numerica*, 24:1–159, 005 2015.
- [12] P. Constantine and G. Iaccarino. Reduced order models for parameterized hyperbolic conservation laws with shock reconstruction. Technical report, Stanford Center for Turbulence Research Annual Research Briefs 2012, 2012. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.727.9918&rep=rep1&type=pdf>.



- [13] W. Dahmen. Multiscale and wavelet methods for operator equations. In C. Canuto, editor, *Multiscale Problems and Methods in Numerical Simulations*. Springer, 2003.
- [14] W. Dahmen. How to best sample a solution manifold? In G. E. Pfander, editor, *Sampling Theory, a Renaissance: Compressive Sensing and Other Developments*, pages 403–435. Springer International Publishing, 2015.
- [15] W. Dahmen, C. Plesken, and G. Welper. Double greedy algorithms: Reduced basis methods for transport dominated problems. *ESAIM: Mathematical Modelling and Numerical Analysis*, 48:623–663, 5 2014.
- [16] B. Després, G. Poëtte, and D. Lucor. *Robust Uncertainty Propagation in Systems of Conservation Laws with the Entropy Closure Method*, pages 105–149. Springer International Publishing, Cham, 2013.
- [17] J.-F. Gerbeau and D. Lombardi. Reduced-order modeling based on approximated lax pairs. Technical report, INRIA, 2012. <http://arxiv.org/pdf/1211.4153v1>.
- [18] J.-F. Gerbeau and D. Lombardi. Approximated lax pairs for the reduced order integration of nonlinear evolution equations. *Journal of Computational Physics*, 265:246 – 269, 2014.
- [19] B. Haasdonk and M. Ohlberger. Reduced basis method for explicit finite volume approximations of nonlinear conservation laws. In *Proceedings of the 12th International Conference on Hyperbolic Problems: Theory, Numerics, Application*, College Park, Maryland, USA, June 09-13 2008.
- [20] B. Haasdonk and M. Ohlberger. Reduced basis method for finite volume approximations of parametrized linear evolution equations. *ESAIM: M2AN*, 42(2):277–302, 2008.
- [21] J. S. Hesthaven, G. Rozza, and B. Stamm. *Certified Reduced Basis Methods for Parametrized Partial Differential Equations*. Springer, 2015.
- [22] A. Iollo and D. Lombardi. Advection modes by optimal mass transfer. *Phys. Rev. E*, 89:022923, Feb 2014.
- [23] S. Jin, D. Xiu, and X. Zhu. A well-balanced stochastic galerkin method for scalar hyperbolic balance laws with random inputs. *Journal of Scientific Computing*, 67(3):1198–1218, 2016.
- [24] K. C. Kiwiel. *Methods of descent for nondifferentiable optimization*, volume 1133 of *Lecture Notes in Mathematics*. Springer-Verlag, Berlin, 1985.
- [25] Q. Li, J.-G. Liu, and R. Shu. Sensitivity analysis of burger’s equation with discontinuous shocks. 2018.

- [26] S. Mishra and C. Schwab. Sparse tensor multi-level monte carlo finite volume methods for hyperbolic conservation laws with random initial data. *Math. Comp.*, 81:1979–2018, 2012.
- [27] R. Mojjani and M. Balajewicz. Lagrangian basis method for dimensionality reduction of convection dominated nonlinear flows. 2017.
- [28] N. J. Nair and M. Balajewicz. Transported snapshot model order reduction approach for parametric, steady-state fluid flows containing parameter dependent shocks. Preprint <https://arxiv.org/abs/1712.09144>, 2017.
- [29] N.-C. Nguyen, G. Rozza, and A. Patera. Reduced basis approximation and a posteriori error estimation for the time-dependent viscous burgers’ equation. *Calcolo*, 46(3):157–185, 2009.
- [30] M. Ohlberger and S. Rave. Nonlinear reduced basis approximation of parameterized evolution equations via the method of freezing. *Comptes Rendus Mathématique*, 351(23–24):901 – 906, 2013.
- [31] M. Ohlberger and S. Rave. Reduced basis methods: Success, limitations and future challenges. In D. Handlovičová, A. Ševčovič, editor, *Proceedings of the Conference Algoritmy 2016*, pages 1–12. Publishing House of Slovak University of Technology in Bratislava, 2016.
- [32] P. Pacciarini and G. Rozza. Stabilized reduced basis method for parametrized advection–diffusion PDEs. *Computer Methods in Applied Mechanics and Engineering*, 274(0):1 – 18, 2014.
- [33] B. Peherstorfer. Model reduction for transport-dominated problems via online adaptive bases and adaptive sampling. 2018.
- [34] P. Pettersson, Q. Abbasb, G. Iaccarino, and J. Nordström. Efficiency of shock capturing schemes for burgers’ equation with boundary uncertainty. In *Seventh South African Conference on Computational and Applied Mechanics SACAM10*, 2010.
- [35] P. Pettersson, G. Iaccarino, and J. Nordström. A stochastic galerkin method for the euler equations with roe variable transformation. *Journal of Computational Physics*, 257, Part A:481 – 500, 2014.
- [36] R. Pulch and D. Xiu. Generalised polynomial chaos for a class of linear conservation laws. *Journal of Scientific Computing*, 51(2):293–312, 2012.
- [37] A. Quarteroni, A. Manzoni, and F. Negri. *Reduced Basis Methods for Partial Differential Equations*. Springer, 2015.
- [38] J. Reiss, P. Schulze, and J. Sesterhenn. The shifted proper orthogonal decomposition: A mode decomposition for multiple transport phenomena. <https://arxiv.org/abs/1512.01985>, 2015.

- [39] D. Rim and K. Mandli. Displacement interpolation using monotone rearrangement. *SIAM/ASA Journal on Uncertainty Quantification*, 6(4):1503–1531, 2018.
- [40] D. Rim and K. Mandli. Model reduction of a parametrized scalar hyperbolic conservation law using displacement interpolation. 2018.
- [41] D. Rim, S. Moe, and R. J. LeVeque. Transport reversal for model reduction of hyperbolic partial differential equations. <https://arxiv.org/abs/1701.07529>, 2017.
- [42] P. Schulze, J. Reiss, and V. Mehrmann. Model reduction for a pulsed detonation combustor via shifted proper orthogonal decomposition. 2018.
- [43] Taddei, T., Perotto, S., and Quarteroni, A. Reduced basis techniques for nonlinear conservation laws. *ESAIM: M2AN*, 49(3):787–814, 2015.
- [44] J. Tryoen, O. L. Maître, and A. Ern. Adaptive anisotropic spectral stochastic methods for uncertain scalar conservation laws. *SIAM Journal on Scientific Computing*, 34(5):A2459–A2481, 2012.
- [45] M. Ulbrich. Optimization methods in banach spaces. In M. Hinze, R. Pinnau, M. Ulbrich, and S. Ulbrich, editors, *Optimization with PDE Constraints*. Springer Netherlands, 2009.
- [46] G. Welper.  $h$  and  $hp$ -adaptive interpolation by transformed snapshots for parametric and stochastic hyperbolic pdes. Submitted, 2017.
- [47] G. Welper. Interpolation of functions with parameter dependent jumps by transformed snapshots. *SIAM Journal on Scientific Computing*, 39(4):A1225–A1250, 2017.
- [48] M. Yano, A. T. Patera, and K. Urban. A space-time  $hp$ -interpolation-based certified reduced basis method for burgers’ equation. *Mathematical Models and Methods in Applied Sciences*, 24(09):1903–1935, 2014.