SIAM J. SCI. COMPUT.
Vol. 39, No. 4, pp. A1225–A1250

INTERPOLATION OF FUNCTIONS WITH PARAMETER DEPENDENT JUMPS BY TRANSFORMED SNAPSHOTS*

G. WELPER†

Abstract. Functions with jumps and kinks typically arising from parameter dependent or stochastic hyperbolic PDEs are notoriously difficult to approximate. If the jump location in physical space is parameter dependent or random, standard approximation techniques like reduced basis methods, PODs, polynomial chaos, etc., are known to yield poor convergence rates. In order to improve these rates, we propose a new approximation scheme. It relies on snapshots for the reconstruction of parameter dependent functions so that it is efficiently applicable in a PDE context. However, the physical domain of each individual snapshot is transformed before its use in the reconstruction, which allows realignng the moving discontinuities and yields high convergence rates. The transforms are automatically computed by minimizing a training error. In order to show feasibility of this approach, it is tested by numerical experiments with one and two physical dimensions and one parameter dimension.

Key words. reduced order modeling, stochastic PDEs, hyperbolic PDEs, shocks, convergence rates, stability

AMS subject classifications. 41A46, 41A25, 35L67, 65M15

DOI. 10.1137/16M1059904

1. Introduction.

A cornerstone of reduced order modeling, stochastic PDEs, and uncertainty quantification is the efficient approximation of high dimensional PDE solutions $u(x,\mu)$ depending on physical variables $x \in \Omega$ and parametric or random variables $\mu \in \mathcal{P} \subset \mathbb{R}^d$. Many contemporary approximation techniques like, e.g., reduced basis methods [28, 30, 27], POD [31, 32, 33, 16, 17], Karhunen–Loève expansion [18, 19], or polynomial chaos [35, 36, 29] build upon a reconstruction by a truncated sum

$$u(x,\mu) \approx \sum_i c_i(\mu)\psi_i(x), \tag{1}$$

where the choice and computation of $c_i(\mu)$ and $\psi_i(x)$ depends on the specific method at hand: For reduced basis methods $\psi_i(x) = u(x,\mu_i)$ are snapshots and the $c_i(\mu)$ are computed by a Galerkin projection. For POD and Karhunen–Loève expansions one minimizes the error between $u(x,\mu)$ and any truncated representation of the form (1), and in the case of polynomial chaos the functions $c_i(\mu)$ are orthogonal polynomials. Throughout this paper, we refer to methods based on (1) by *separation of variables based methods*. The success of all these methods relies on the fact that for many problems one can truncate this sum to a few summands only for the price of a very small error.

However, this regularity assumption is not always true. An important class of problems are functions $u(x,\mu)$ that have parameter dependent or random jumps or kinks arising, e.g., in parametric or stochastic hyperbolic PDEs. Separation of variables based methods are expected to perform poorly for these types of problems. In fact in Appendix A, we consider a counterexample for which no separation of vari-

*Submitted to the journal's Methods and Algorithms for Scientific Computing section February 3, 2016; accepted for publication (in revised form) April 27, 2017; published electronically July 12, 2017.
http://www.siam.org/journals/sisc/39-4/M105990.html
†Department of Mathematics, Texas A&M University, College Station, Texas 77843-3368 (welper@math.tamu.edu).

ables based method can achieve a convergence rate higher than 1 with respect to the number of summands in a separation of variables decomposition (1). See also [13] for a survey in case of uncertainty quantification.

There are relatively few methods in the literature [5, 25, 14, 34, 1] that directly address this poor performance for parameter dependent jumps and kinks. Instead, much of the work does use separation of variables and focuses on different problems arising in the context of reduced order modeling of parametric hyperbolic PDEs and singularly perturbed problems: Solving the PDE directly in a reduced basis, stabilization, online/offline decompositions, and error estimators; see [12, 11, 24, 37, 26, 7, 6].

The goal of this paper is the construction of an alternative approximation method to replace standard separation of variables in order to achieve higher convergence rates for functions $u(x, \mu)$ with parameter dependent jumps and kinks. In addition, the method relies on snapshots and optionally error estimators as input data so that it can be used efficiently and nonintrusively with existing PDE solvers. Somewhat similar to [25], we allow a transformation $\phi(\mu, \eta) : \Omega \to \Omega$ of the physical domain before we use a snapshot $u(x, \eta)$ in the reconstruction of $u(x, \mu)$, i.e.,

$$u(x, \mu) \approx \sum_{i=0}^{n} c_i(\mu) u(\phi(\mu, \eta_i)(x), \eta_i), \tag{2}$$

where η_i for $i = 0, \dots, n$ are finitly many parameters. The purpose of the additional transform is an alignment of the discontinuities of $u(x, \eta_i)$ with the ones of $u(x, \mu)$. As a result the discontinuities are "invisible" in parameter direction so that very few summands yield accurate approximations. More rigorously, we prove a high order error estimate that does not depend on the regularity of $u(x, \mu)$ itself, but on the regularity of the modified snapshots after alignment which is considerably higher for many practical problems. In addition, because exact alignment is rarely possible in practice, we also take perturbation results into account. Similar to greedy methods for the construction of reduced bases, or neural networks, the transform ϕ is computed by minimizing the approximation error on a training sample of snapshots. Although this might seem prohibitively complicated, in section 4 we discuss some preliminary arguments to avoid being trapped in local minima, and in section 5 some two dimensional (2D) numerical experiments are provided where simple subgradient methods provide good results.

The outlined approximation scheme allows for various realizations with regard to the choices of the coefficients $c_i(\mu)$ or the inner transforms $\phi(\mu, \eta_i)$. Because the main objective of the paper is a proof of principle that one can approximate functions with parameter dependent jumps and kinks with high order from snapshots alone, we make the rather simple choice of Lagrange interpolation polynomials throughout the paper. This allows multivariate parameters but is expected to suffer from the curse of dimensionality. In this paper, we focus on the problems induced by discontinuities alone and leave additional techniques required for high dimensional parameter spaces for future research.

The paper is organized in two parts. In a first part, the transformed based reconstruction (2) is introduced (section 2). We analyze its computational complexity (section 2), error (section 2), and stability (section 3). Throughout this part of the paper, we assume that suitable transforms $\phi(\mu, \eta)$ are given. Their computation is then discussed in a second part of the paper. They are computed by minimizing a training error in section 4.1. Then, in section 4.2, we introduce strategies to avoid local minima. Finally, section 5 provides some numerical experiments. For the sake of

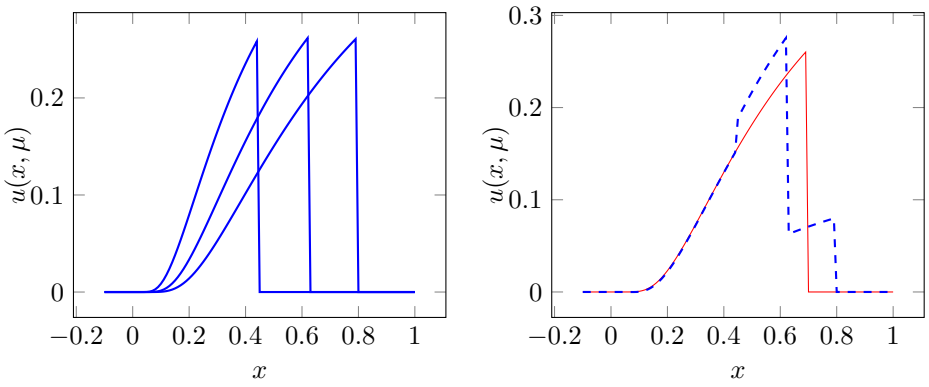

FIG. 1. *Left: Snapshots of the parametric function* (3) *for parameters* $\mu = 0.5, 0.85, 1.2$. *Right: Exact solution (red) and polynomial interpolation (blue, dashed) for* $\mu = 1.0$.

completeness, in Appendix A we consider a counterexample for which no separation of variables based method can achieve high order convergence rates, and in Appendix B we discuss some drawbacks that arise when construction transforms by characteristics.

2. Transformed snapshot interpolation. In order to motivate the new approximation scheme, we consider the following prototype example throughout this section:

$$u(x,\mu) := \psi\left(\frac{x}{0.4+\mu} - 1\right), \qquad \psi(x) := \begin{cases} \exp\left(-\frac{1}{1-x^2}\right) & -1 \le x < -\frac{1}{2}, \\ 0 & \text{else}, \end{cases} \tag{3}$$

where ψ is the standard mollifier cut off at $x = -1/2$. This is not necessarily a solution of a PDE but has the main features we are interested in: a discontinuity that is moving with the parameter. An example for an approximation by separation of variables can be seen in Figure 1, where we recover $u(x,\mu)$ from three snapshots by a polynomial interpolation

$$u(x,\mu) \approx \sum_{i=0}^{n} \ell_i(\mu) u(x,\eta_i),$$

where η_i, $i = 0,\dots,n$ are some interpolation points and ℓ_i, $i = 0,\dots,n$ are the corresponding Lagrange interpolation polynomials. We see the typical "staircasing behavior," which significantly deteriorates the solutions quality. Although our choice of the separation of variables is perhaps overly simplistic, reduced basis methods and other more sophisticated schemes suffer from the same problem; see the discussion in Appendix A.

Unlike this superposition of snapshots resulting in the staircasing phenomena, it seems much more intuitive to compute one snapshot $u(x,\eta)$ and recover the function $u(x,\mu)$ for a different μ by stretching this snapshot such that the left end of the support is fixed and the jump locations match. To state this intuition in mathematical terms, "stretching" one function $u(x,\eta)$ to match a second one $u(x,\mu)$ essentially boils down to a transform $\phi(\mu,\eta) : \Omega \to \Omega$ of the physical variables, depending on "source" parameter $\eta \in \mathcal{P}$ and "target" parameter $\mu \in \mathcal{P}$, so that we obtain the approximation by a *transformed snapshot*

$$u(x,\mu) \approx v_\mu(x,\eta) := u(\phi(\mu,\eta)(x),\eta). \tag{4}$$

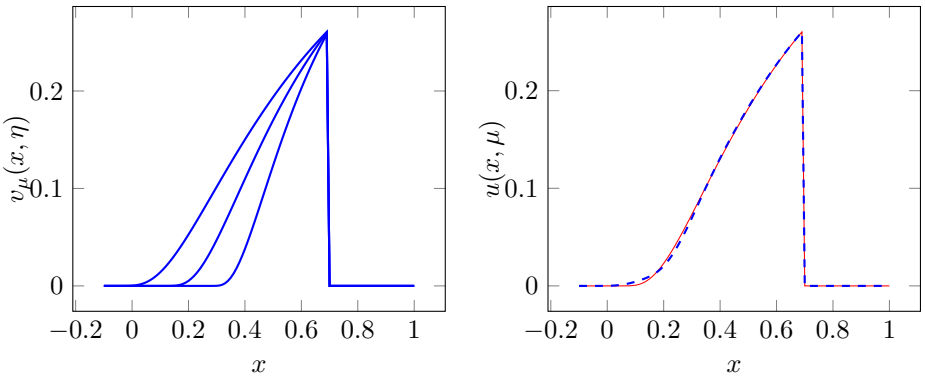

FIG. 2. *Left: Transformed snapshots of the parametric function* (3) *for parameters* $\mu = 1.0$ *and* $\eta = 0.5, 0.85, 1.2$. *Right: Exact solution (red) and transformed snapshot interpolation (blue, dashed) for* $\mu = 1.0$.

In general, even for optimal choices of ϕ, we cannot obtain arbitrarily good approximation errors in this way. A trivial example is $u(\cdot,\mu) = 0$ and $u(\cdot,\eta) = 1$. No matter what ϕ we choose, we cannot match these two functions. However, by construction, the transformed snapshots $v_\mu(x,\eta)$ have jump locations at he same x-values as the target $u(\cdot,\mu)$, as shown in Figure 2. Since these locations are independent of η, the functions $\eta \to v_\mu(x,\eta)$ are smooth. In order to use this observation in an approximation scheme, we make one additional assumption: For the same source and target parameter $\mu = \eta$, nothing needs to be aligned and therefore we assume

$$\phi(\mu,\mu)(x) = x. \tag{5}$$

This implies that for $\eta = \mu$, we have

$$v_\mu(\cdot,\mu) = u(x,\mu).$$

Hence, instead of interpolating the irregular snapshots $\eta \to u(\cdot,\eta)$ at $\eta = \mu$, we can interpolate the smooth transformed snapshots $\eta \to v_\mu(\cdot,\eta)$ at $\eta = \mu$, instead. To this end, let $\eta_i \in \mathcal{P}$ be interpolation points for i in some index set Γ_n of cardinality $|\Gamma_n|$. For the time being, we assume that we already know the transform ϕ and defer its calculation to section 4 below. Then after computing the snapshots $u(\cdot,\eta_i)$, we also know the transformed snapshots $v_\mu(\cdot,\eta_i)$ and their interpolation

$$u(x,\mu) \approx u_n(x,\mu) := \sum_{i\in\Gamma_n} \ell_i(\mu) u(\phi(\mu,\eta_i)(x),\eta_i), \tag{6}$$

denoted by *transformed snapshot interpolation (TSI)* in the following, yields an approximation of $u(\cdot,\mu)$. In general, the ℓ_i are Lagrange interpolation polynomials in a possibly multivariate polynomial space $\mathbb{P}^n$ so that the interpolation problem for interpolation points η_i, $i \in \Gamma_n$, is well-posed. Other interpolation type methods are also conceivable like, e.g., reduced basis methods. See, e.g., [21] for variants that are applicable without coercivity and residual based error estimators.

Besides the snapshots themselves this method requires the knowledge of the transforms $(x,\mu) \to \phi(\mu,\eta_i)(x)$ for finitely many η_i. Thus instead of approximating one single function depending on x and μ we have to find many of them! However, whereas

separation of variables performs poorly for $u(x,\mu)$, it often yield good results for the transforms $\phi(\mu,\eta_i)$: Their smoothness with respect to μ depends on the smoothness of the jump or kink location with respect to the parameter and not the smoothness of $u(x,\mu)$ itself. For example, in (3) the jump location is $j(\mu) = \frac{1}{5} + \frac{1}{2}\mu$ so that a linear transform

$$\phi(\mu,\eta)(x) = x - j(\mu) + j(\eta)$$

is sufficient to align the jumps. As shown in Figure 2, this transform does not align the left end of the supports, but the transformed snapshots are infinitely differentiable with respect to η, nonetheless. Therefore, this mismatch is taken care of by the outer interpolation. The results of the TSI for this example are shown in Figure 2, which achieves a much improved reconstruction around the jump.

Let us summarize the computational steps. Similar to reduced basis methods, the work is split into an offline and online phase.

Offline.

1. Compute the snapshots $u(x,\eta_i)$, $i \in \Gamma_n$.
2. For all $i \in \Gamma_n$, compute the inner transforms $(x,\mu) \to \phi(\mu,\eta_i)(x)$, $i \in \Gamma_n$, as described in section 4 below.

Online.

1. For each new μ, compute $u(\cdot,\mu)$ by (6).

In order to state an error estimate, recall that the Lebesgue constant is the norm of the polynomial interpolation operator in the sup-norm which is given by

$$\Lambda_n := \sup_{\mu\in\mathcal{P}} \sum_{i\in\Gamma_n} |\ell_i(\mu)|. \tag{7}$$

We obtain the following error estimate.

PROPOSITION 2.1. *Assume $u_n(x,\mu)$ is defined by the transformed snapshot interpolation* (6)*. Then for all $\mu \in \mathcal{P}$ the error is bounded by*

$$\|u(\cdot,\mu) - u_n(\cdot,\mu)\|_{L_1} \le (1+\Lambda_n)\Big\| \inf_{p\in\mathbb{P}^n} \|v_\mu - p\|_{L_\infty(\mathcal{P})}\Big\|_{L_1(\Omega)}.$$

Proof. The proof follows directly from $u(x,\mu) = v_\mu(x,\mu)$ and standard interpolation estimates applied to $\eta \to v_\mu(x,\eta)$. ∎

For this proposition as well as the remainder of this paper we choose the L_1-norm to measure errors because it is the most common choice for hyperbolic PDEs. Also note that the given result is just one option of the various estimates for polynomial interpolation. For example, if we assume analytic dependence of $v_\mu(x,\eta)$ on η and use Chebyshev nodes for a one dimensional (1D) parameter, we can achieve exponential convergence rates. The most important observation, however, is that the estimate does not involve any regularity assumption of $u(x,\mu)$ itself. Instead, it relies on the regularity of $v_\mu(x,\eta)$ with respect to η which can be considerably better.

In summary, instead of approximating the nonsmooth function $u(x,\mu)$ directly, for every target $\mu \in \mathcal{P}$ we construct a new smooth function $(x,\eta) \to v_\mu(x,\eta)$ and approximate this function instead. The interpolation condition (5) guarantees that $u(x,\mu) = v_\mu(x,\eta)$ so that this yields accurate approximations of $u(x,\mu)$ itself as depicted in Figure 3. In addition, for our preliminary simple linear interpolation of $v_\mu(x,\eta)$ this allows an offline/online decomposition.

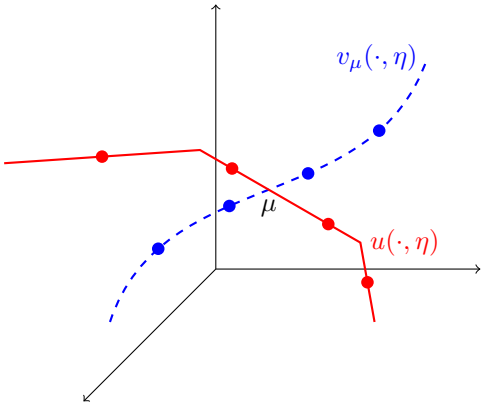

FIG. 3. *Three dimensional coordinate axes: Indicate the space $L_1(\Omega)$, which is infinite dimensional in reality. Red curve: Solution manifold. In the picture we use one dimensional parameters so that this is simply a curve $\mu \to u(\cdot,\mu)$ in $L_1(\Omega)$. The kinks are supposed to indicate that this curve is not smooth. In fact is is not even differentiable for every parameter μ. Blue dashed line: manifold of transformed snapshots, in the picture the curve $\eta \to v_\mu(\cdot,\eta)$. In contrast to the solution manifold this curve is smooth. Red and blue dots: Snapshots and transformed snapshots, respectively. By construction, the red and blue lines intersect at the target $u(\cdot,\mu)$ that we want to reconstruct, so that we can either interpolate the snapshots or the transformed snapshots which are more regular.*

3. Stability. Of course the high order smoothness of $v_\mu(x,\eta)$ with respect to η needed for high approximation orders in Proposition 2.1 requires that jumps and kinks are exactly aligned. However, for any finite approximation of the inner transform ϕ, this is rarely possible. Therefore, we next consider two perturbation results, which allow us to bound the error while taking approximation errors of the inner transform into account. The first one, Lemma 3.1, relies on a measure theoretic argument and allows rather general transforms including ones with kinks as found in, e.g., finite element discretizations. The second one, Corollary 3.2, avoids measure theory but requires the inner transforms $\phi(\mu,\eta)$ to be diffeomorphisms.

In the following, let $\varphi(\mu,\eta)(x)$ be a perturbation of $\phi(\mu,\eta)(x)$. To simplify the arguments below, for the time being, we forget about the parameter dependence and consider two transforms $\varphi,\phi:\Omega\to\Omega$ instead. If we assume that each point $\phi(x)$ can be connected to the point $\varphi(x)$ along a curve $\Phi^s(x)$ for s in the interval $[0,1]$, we can rewrite the perturbation by the fundamental theorem of line integrals

$$(u\circ\phi)(x)-(u\circ\varphi)(x)=\int_0^1 u'(\Phi^s(x))\partial_s\Phi^s(x)\,\mathrm{d}s,$$

so that it remains to estimate the right-hand side. The map $(x,s)\to\Phi^s(x)$ can be regarded as a function from $\Omega\times[0,1]\to\Omega$ such that

$$\Phi^0(x)=\phi(x),\qquad\qquad \Phi^1(x)=\varphi(x), \tag{8}$$

which is a homotopy between ϕ and φ if it is continuous in addition. Furthermore, let λ be the Lebesgue measure and $\mathcal{A}$ the Lebesgue σ-algebra on Ω and let $\Phi^s_*\lambda$ denote the pushforward measure defined by

$$\Phi^s_*\lambda(A)=\lambda((\Phi^s)^{-1}(A))\quad\text{for all } A\in\mathcal{A}.$$

Then we have the following lemma.

LEMMA 3.1. *Assume that $u \in BV(\Omega)$ and Φ^s, $0 \le s \le 1$, given by (8) is measurable and differentiable with respect to s such that*

$$\Phi^s_* \lambda(A) \le c\lambda(A) \quad \text{for all } A \in \mathcal{A} \text{ and } 0 \le s \le 1 \tag{9}$$

and

$$\sup_{\substack{0\le s\le 1\\ x\in\Omega}} |\partial_s \Phi^s(x)| \le C\|\phi - \varphi\|_{L_\infty(\Omega)} \tag{10}$$

for constants $c, C \ge 0$. Then we have

$$\|u \circ \phi - u \circ \varphi\|_{L_1(\Omega)} \le cC\|u\|_{BV(\Omega)}\|\phi - \varphi\|_{L_\infty(\Omega)}. \tag{11}$$

Let us discuss the main assumptions before we prove the proposition. If the speed $|\partial_s \Phi^s(x)|$ of each curve $s \to \Phi^s(x)$ is quasi-uniform, i.e., equivalent to a constant S for all x and s, we have

$$\|\partial_s \Phi^s\|_{L_\infty(\Omega\times[0,1])} \sim \int_0^1 |\partial_s \Phi^s(x)| \,\mathrm{d}s =: l(x),$$

where $l(x)$ is the length of the curve connecting $\phi(x)$ to $\varphi(x)$. In that case assumption (10) states that, up to a constant, the length of each curve $\Phi^s(x)$ is bounded by the distance $|\phi(x) - \varphi(x)|$ of its endpoints.

In the case that the domain Ω is convex, a simple choice of the curves $\Phi^s(x)$ is the convex combination of the end points:

$$\Phi^s(x) = (1-s)\phi(x) + s\varphi(x). \tag{12}$$

In that case, we have $\partial_s \Phi^s(x) = \varphi(x) - \phi(x)$ so that condition (10) is satisfied.

In order to justify the second assumption (9), let us consider the following scenario: Assume that $\phi(x) = x_0 \in \Omega$ and $\varphi(x) = x_1 \in \Omega$ map all of Ω to single points. Furthermore let u be a piecewise constant function with a jump so that x_0 and x_1 are on different sides of this jump. On the one hand we obtain $\|u \circ \phi - u \circ \varphi\|_{L_1(\Omega)} = \|u(x_0) - u(x_1)\|_{L_1(\Omega)} = h\lambda(\Omega)$, where h is the height of the jump. On the other hand we have $\|\phi - \varphi\|_{L_\infty(\Omega)} = |x_0 - x_1|$, which can be made arbitrary small by suitable choices of x_0 and x_1 on each side of the jump. Thus the main statement (11) of the proposition is violated. This counterexample relies on the fact that both transforms concentrate all weight in a single point such that $\Phi^i_* \lambda(\{x_i\}) = \lambda(\Omega)$, $i = 0, 1$, which is ruled out by assumption (9).

Finally, we assume that the outer function $u \in BV(\Omega)$ is of bounded variation. This allows jumps and kinks and is one of the most common norms for stability results of hyperbolic PDEs.

Proof of Lemma 3.1. For the time being, let us assume that $u \in C^1(\Omega)$. Applying the fundamental theorem for line integrals, we obtain

$$(u \circ \phi)(x) - (u \circ \varphi)(x) = \int_0^1 u'(\Phi^s(x))\partial_s \Phi^s(x) \,\mathrm{d}t.$$

Thus, we have

$$\begin{aligned}\|u\circ\phi - u\circ\varphi\|_{L_1(\Omega)} &= \int_\Omega \left| \int_0^1 u'(\Phi^s(x))\partial_s\Phi^s(x)\,\mathrm{d}s \right| \mathrm{d}x \\ &\le \int_\Omega \int_0^1 |u'(\Phi^s(x))||\partial_s\Phi^s(x)|\,\mathrm{d}s\,\mathrm{d}x \\ &\le \sup_{\substack{0\le s\le 1\\ x\in\overline{\Omega}}} |\partial_s\Phi^s(x)| \int_\Omega \int_0^1 |u'(\Phi^s(x))|\,\mathrm{d}s\,\mathrm{d}x \\ &\le C\|\phi-\varphi\|_{L_\infty(\Omega)} \int_0^1 \int_\Omega |u'(\Phi^s(x))|\,\mathrm{d}x\,\mathrm{d}s.\end{aligned}$$

Using the pushforward measure $\Phi^s_*\lambda^{n+1}$ and its bound (9) we conclude that

$$\int_\Omega |u'(\Phi^s(x))|\,\mathrm{d}x = \int_\Omega |u'(y)|\,\mathrm{d}\Phi^s_*\lambda(x) \le c\int_\Omega |u'(y)|\,\mathrm{d}x. \tag{13}$$

Combining the last two estimates and using that $\int_0^1 \mathrm{d}s = 1$ yields

$$\|u\circ\phi - u\circ\varphi\|_{L_1(\Omega)} \le cC\|\phi-\varphi\|_{L_\infty(\Omega)} \int_\Omega |u'(y)|\,\mathrm{d}y,$$

which is equivalent to the estimate (11) we wish to prove.

Finally, we extend the estimate to all $u \in BV(\Omega)$ by using a density argument. To this end, note that for all $\epsilon > 0$ there is a $u_\epsilon \in C^1(\Omega)$ such that

$$\|u-u_\epsilon\|_{L_1(\Omega)} \le \epsilon, \qquad \|u_\epsilon'\|_{L_1(\Omega)} \le \|u\|_{BV(\Omega)} + \epsilon;$$

see, e.g., [38, Theorem 5.3.3]. Thus, to apply a density argument, is suffices to bound $\|u\circ\phi - u_\epsilon\circ\phi\|_{L_1(\Omega)}$ and $\|u\circ\varphi - u_\epsilon\circ\varphi\|_{L_1(\Omega)}$. Analogously to (13) we obtain

$$\begin{aligned}\|u\circ\phi - u_\epsilon\circ\phi\|_{L_1(\Omega)} &= \int_\Omega |u(\Phi^0(x)) - u_\epsilon(\Phi^0(x))|\,\mathrm{d}x \\ &= \int_\Omega |u(y) - u_\epsilon(y)|\,\mathrm{d}\Phi^0_*\lambda(y) \\ &\le c\int_\Omega |u(y)-u_\epsilon(y)|\,\mathrm{d}y \\ &\le c\|u-u_\epsilon\|_{L_1(\Omega)}.\end{aligned}$$

The bound for $\|u\circ\varphi - u_\epsilon\circ\varphi\|_{L_1(\Omega)}$ follows analogously, which completes the proof. ∎

If the transforms $\Phi^s(x)$ can be chosen to be diffeomorphisms, the pushforward measure is explicitly given by the usual transformation rule

$$\Phi^s_*\lambda(A) = \int_A |\det D_x(\Phi^s)^{-1}(x)|\,\mathrm{d}\lambda(x) \tag{14}$$

so that we obtain the following corollary.

COROLLARY 3.2. *Assume that $u \in BV(\Omega)$ and that Φ^s, $0 \le s \le 1$ given by (8) are diffeomorphisms for fixed s and differentiable with respect to s such that*

$$|\det D_x(\Phi^s)^{-1}(x)| \le c \quad \text{for all } A \in \mathcal{A} \text{ and } 0 \le s \le 1 \tag{15}$$

and

$$\sup_{\substack{0\le s\le 1\\ x\in\Omega}} |\partial_s \Phi^s(x)| \le C\|\phi - \varphi\|_{L_\infty(\Omega)}$$

for constants $c, C \ge 0$. Then we have

$$\|u\circ\phi - u\circ\varphi\|_{L_1(\Omega)} \le cC\|u\|_{BV(\Omega)}\|\phi-\varphi\|_{L_\infty(\Omega)}.$$

Proof. We just have to show the bounds (9) of the pushforward. By its explicit formula (14) we have

$$\Phi^s_*\lambda(A) = \int_A |\det D_x(\Phi^s)^{-1}(x)|\,\mathrm{d}\lambda(x) \le c\lambda(A)$$

for all $A \in \mathcal{A}$ and $0 \le s \le 1$ so that the corollary follows from Lemma 3.1. ∎

Let us now consider the transformed snapshot interpolation (6) again. Assume that there is a transform $\phi(\mu,\eta)(x)$ that aligns the jumps and kinks exactly so that we obtain high convergence rates in Proposition 2.1. In general, we have to find a finite approximation to this exact transform, say, $\phi_m(\mu,\eta)(x)$. Note that according to (6) we only need to know this function for the $|\Gamma_n|$ nodes η_i, $i \in \Gamma_n$, so that we have to approximate $|\Gamma_n|$ functions depending on $x \in \Omega$ and a parameter $\mu \in \mathcal{P}$. Of course this is exactly the same problem as approximating a function $u(x,\mu)$ which is our initial problem; however, the regularity of $\phi(\mu,\eta)(x)$ can be much more favorable as we have seen in the introduction in section 2 or as we will see in the numerical experiments below. Therefore, we can apply a more classical separation of variables based approach to find an approximation $\phi_m(\mu,\eta)(x)$ of the inner transform. Replacing the exact transform by the approximate one in the transformed snapshot interpolation yields

$$u(x,\mu) \approx u_{n,m}(x,\mu) := \sum_{i\in\Gamma_n} \ell_i(\mu) u(\phi_m(\mu,\eta_i)(x),\eta_i). \tag{16}$$

Combining the error estimate of Proposition 2.1 with the perturbation result Lemma 3.1 we arrive at the following proposition.

PROPOSITION 3.3. *Assume that $u \in BV(\Omega)$ and that there are curves $\Phi^s(\mu,\eta_i)(x)$, $0 \le s \le 1$ for $x \in \Omega$, $\mu \in \mathcal{P}$, $i \in \Gamma_n$ measurable and differentiable with respect to s such that*

$$\Phi(\mu,\eta_i)^0(x) = \phi(\mu,\eta_i)(x), \qquad \Phi(\mu,\eta_i)^1(x), = \phi_m(\mu,\eta_i)(x) \tag{17}$$

and

$$\Phi(\mu,\eta_i)^s_*\lambda(A) \le c\lambda(A) \quad \text{for all } A \in \mathcal{A} \text{ and } 0 \le s \le 1 \tag{18}$$

and

$$\sup_{\substack{0\le s\le 1\\ x\in\Omega}} |\partial_s \Phi(\mu,\eta_i)^s(x)| \le C\|\phi(\mu,\eta_i) - \phi_m(\mu,\eta_i)\|_{L_\infty(\Omega)} \tag{19}$$

for constants $c, C \geq 0$. Furthermore let $u_{n,m}(x,\mu)$ be defined by the transformed snapshot interpolation (16)*. Then for all $\mu \in \mathcal{P}$ we have the error estimate*

$$\|u(\cdot,\mu) - u_{n,m}(\cdot,\mu)\|_{L_1} \leq (1+\Lambda_n)\Big\| \inf_{p\in\mathbb{P}^n} \|v_\mu - p\|_{L_\infty(\mathcal{P})}\Big\|_{L_1(\Omega)} + cC\Lambda_n \max_{i\in\Gamma_n} \|u(\cdot,\eta_i)\|_{BV(\Omega)} \max_{i\in\Gamma_n} \|\phi(\mu,\eta_i) - \phi_m(\mu,\eta_i)\|_{L_\infty(\Omega)},$$

where Λ_n is the Lebesgue constant (7)*.*

In order to discuss the assumptions, let us consider the simple case that Ω is convex and that there are a diffeomorphisms $\phi(\mu,\eta)$ that align the jumps. For an approximation ϕ_m of ϕ, we can then define $\Phi(\mu,\eta_i)^s$ analogously to the convex combination (12), which directly satisfy (17) and (19). The last condition (18) can then be reformulated by a smooth variant analogously to (15), which is satisfied if ϕ_m is sufficiently close to ϕ in C^1. Note, however, that the proposition does neither require convex domains nor C^1 approximations.

Proof. We have

$$\|u(\cdot,\mu) - u_{n,m}(\cdot,\mu)\|_{L_1(\Omega)} \leq \|u(\cdot,\mu) - u_n(\cdot,\mu)\|_{L_1(\Omega)} + \|u_n(\cdot,\mu) - u_{n,m}(\cdot,\mu)\|_{L_1(\Omega)}.$$

With Proposition 2.1 the first term can be estimated by

$$\|u(\cdot,\mu) - u_n(\cdot,\mu)\|_{L_1} \leq (1+\Lambda_n)\Big\| \inf_{p\in\mathbb{P}^n} \|v_\mu - p\|_{L_\infty(\mathcal{P})}\Big\|_{L_1(\Omega)}.$$

In order to estimate the second term, using the definition of the Lebesgue constant and Lemma 3.1, we obtain

$$\begin{aligned}
&\|u_n(\cdot,\mu) - u_{n,m}(\cdot,\mu)\|_{L_1(\Omega)} \\
&\quad\leq \sum_{i\in\Gamma_n} |\ell_i(\mu)| \Big\| u(\phi(\mu,\eta_i)(x),\eta_i) - u(\phi_m(\mu,\eta_i)(x),\eta_i)\Big\|_{L_1(\Omega)} \\
&\quad\leq \Lambda_n \max_{i\in\Gamma_n} \|u(\phi(\mu,\eta_i)(x),\eta_i) - u(\phi_m(\mu,\eta_i)(x),\eta_i)\|_{L_1(\Omega)} \\
&\quad\leq cC\Lambda_n \max_{i\in\Gamma_n} \|u(\cdot,\eta_i)\|_{BV(\Omega)} \max_{i\in\Gamma_n} \|\phi(\mu,\eta_i) - \phi_m(\mu,\eta_i)\|_{L_\infty(\Omega)}.
\end{aligned}$$

Combining all three estimates completes the proof. ∎

If the μ dependence of $\phi(\mu,\eta)$ is smooth, we can use separation of variables for its approximation. Although there are much more sophisticated methods, possibly the simplest choice is a linear interpolation

$$\phi_m(\mu,\eta)(x) = \sum_{j\in\hat{\Gamma}_n} \hat{\ell}_j(\mu)\phi(\mu_j,\eta)(x) \tag{20}$$

with Lagrange polynomials $\hat{\ell}_j$ and interpolation points μ_j with j in some finite index set $\hat{\Gamma}_m$. Below, we use $\hat{\Gamma}_m = \Gamma_n$, $\mu_i = \eta_i$ and $\hat{\ell}_i = \ell_i$, but one can also use different choices, in principle. With this inner approximation, the error bound of Proposition 3.3 depends of the smoothness of the transformed snapshot $v_\mu(x,\eta)$ with respect to η and of the transforms $(x,\mu) \to \phi(\mu,\eta_i)(x)$ with respect to μ. If both dependencies are analytic, for suitable interpolation points the error decays exponentially.

4. Optimizing the interpolation error.

4.1. Generalized gradients. We still need to find a realistic way to actually compute the inner transform $\phi(\mu, \eta)$. One might be tempted to use a construction based on charactersitics, but that is generally problematic, as shown in Appendix B. Similar to the construction of reduced bases, PODs, or neural networks, we aim at finding an inner transform ϕ that minimizes the approximation error. To this end, we measure the error in the sup-norm with respect to the parameter, which is typical for reduced basis methods but not mandatory. It follows that the overall error is given by

$$\sigma_{\mathcal{P}}(\phi) := \sup_{\mu \in \mathcal{P}} \sigma_\mu(\phi), \tag{21}$$

where

$$\sigma_\mu(\phi) := \|u(\cdot, \mu) - u_n(\cdot, \mu; \phi)\|_{L_1(\Omega)}$$

is the error for one fixed parameter. To make the dependence on the inner transform more explicit, in this section we denote the transformed snapshot interpolation (6) by $u_n(x, \mu; \phi) = u_n(x, \mu)$. In practice it is not possible to minimize the error $\sigma_{\mathcal{P}}(\phi)$ directly because it would require the knowledge of all functions $u(\cdot, \mu)$ for all parameters $\mu \in \mathcal{P}$. To this end, we only assume to know the errors $\sigma_\mu(\phi)$ of a finite training sample $\mu \in \mathcal{T} \subset \mathcal{P}$ so that the overall error (21) is replaced by the training error

$$\sigma_{\mathcal{T}}(\phi) := \sup_{\mu \in \mathcal{T}} \sigma_\mu(\phi). \tag{22}$$

Although surrogates for the training error are available for some singularly perturbed problems [24, 37, 7, 6] we omit these in favor of future research. Instead, we resort to an explicit knowledge of some training snapshots $u(\cdot, \mu)$, $\mu \in \mathcal{T}$ in addition to the snapshots that are used for the reconstruction (6) itself. In contrast to the reduced basis method this severely limits the size of the training sample. Nevertheless, in section 5 we consider examples which yield good results with roughly twice as many training snapshots than reconstruction snapshots, so that the additional burden of the training samples is reasonable.

Because we are explicitly interested in nonsmooth functions u, the error $\sigma_{\mathcal{T}}(\phi)$ is a nontrivial objective function to minimize. The next proposition shows that despite possible jumps of u the error is Lipschitz continuous, nonetheless. The assumptions of this proposition are essentially the same as for Lemma 3.1 and are commented right after it.

PROPOSITION 4.1. *Assume that $u \in BV(\Omega)$ and that there are curves $\Phi^s(\mu, \eta_i)(x)$, $0 \le s \le 1$ for $x \in \Omega$, $\mu \in \mathcal{P}$, $i \in \Gamma_n$ measurable and differentiable with respect to s such that*

$$\Phi^0(\mu, \eta_i)(x) = \phi(\mu, \eta_i)(x), \qquad \Phi^1(\mu, \eta_i)(x) = \varphi(\mu, \eta_i)(x) \tag{23}$$

and

$$\Phi^s(\mu, \eta_i)_* \lambda(A) \le c\lambda(A) \quad \text{for all } A \in \mathcal{A} \text{ and } 0 \le s \le 1$$

and

$$\sup_{\substack{0 \le s \le 1 \\ x \in \Omega}} |\partial_s \Phi(\mu, \eta_i)^s(x)| \le C \|\phi(\mu, \eta_i) - \varphi(\mu, \eta_i)\|_{L_\infty(\Omega)}$$

for constants $c, C \geq 0$. Then we have

$$|\sigma_{\mathcal{T}}(\phi) - \sigma_{\mathcal{T}}(\varphi)| \leq cC\Lambda_n \sup_{i\in\Gamma_n} \|u(\cdot,\eta_i)\|_{BV(\Omega)} \sup_{\substack{\mu\in\mathcal{P}\\ i\in\Gamma_n}} \|\phi(\mu,\eta_i) - \varphi(\mu,\eta_i)\|_{L_\infty(\Omega)}. \tag{24}$$

Proof. Note that the triangle inequality implies that

$$|\sigma_{\mathcal{T}}(\phi) - \sigma_{\mathcal{T}}(\varphi)| = \left|\sup_{\mu\in\mathcal{T}} \sigma_\mu(\phi) - \sup_{\mu\in\mathcal{T}} \sigma_\mu(\varphi)\right| \leq \sup_{\mu\in\mathcal{T}} |\sigma_\mu(\phi) - \sigma_\mu(\varphi)| \tag{25}$$

so that it is sufficient to bound $|\sigma_\mu(\phi) - \sigma_\mu(\varphi)|$. To this end note that

$$\begin{aligned}
&|\sigma_\mu(\phi) - \sigma_\mu(\varphi)| \\
&\quad = \Big|\|u(\cdot,\mu) - u_n(\cdot,\mu;\phi)\|_{L_1(\Omega)} - \|u(\cdot,\mu) - u_n(\cdot,\mu;\varphi)\|_{L_1(\Omega)}\Big| \\
&\quad \leq \Big\|\big[u(\cdot,\mu) - u_n(\cdot,\mu;\phi)\big] - \big[u(\cdot,\mu) - u_n(\cdot,\mu;\varphi)\big]\Big\|_{L_1(\Omega)} \\
&\quad \leq \left\|\sum_{i\in\Gamma_n} \ell_i(\mu)\big[u(\phi(\mu,\eta_i)(x),\eta_i) - u(\varphi(\mu,\eta_i)(x),\eta_i)\big]\right\|_{L_1(\Omega)} \\
&\quad \leq \sum_{i\in\Gamma_n} |\ell_i(\mu)|\, \|u(\phi(\mu,\eta_i)(x),\eta_i) - u(\varphi(\mu,\eta_i)(x),\eta_i)\|_{L_1(\Omega)}.
\end{aligned}$$

Due to the given assumptions, we can now apply Lemma 3.1 to conclude that

$$\begin{aligned}
|\sigma_\mu(\phi) - \sigma_\mu(\varphi)| &\leq cC \sum_{i\in\Gamma_n} |\ell_i(\mu)| \|u(\cdot,\eta_i)\|_{BV(\Omega)} \|\phi(\mu,\eta_i) - \varphi(\mu,\eta_i)\|_{L_\infty(\Omega)} \\
&\leq cC\Lambda_n \sup_{i\in\Gamma_n} \|u(\cdot,\eta_i)\|_{BV(\Omega)} \sup_{i\in\Gamma_n} \|\phi(\mu,\eta_i) - \varphi(\mu,\eta_i)\|_{L_\infty(\Omega)}.
\end{aligned}$$

With (25) this yields claimed estimate (24). ∎

In order to optimize the training error $\sigma_{\mathcal{T}}(\phi)$, we search for a minimizer in a set of candidate transforms $\phi \in \Phi \subset X$ in a Banach space X. The space of continuous functions $C(\mathcal{P}\times\mathcal{P}\times\Omega)$ with the additional restrictions from Proposition 4.1 seems to be a reasonable choice for Φ because in the last proposition the transform error is measured in the supremum norm. In general this objective function is not differentiable so that we cannot rely on standard gradient based optimizers. However, because $\sigma_{\mathcal{T}}(\phi)$ is Lipschitz continuous according to the last proposition, we can use optimization methods from nonsmooth optimization [15, 3] relying on the generalized Clarke gradient [4]. To this end, for a direction $v \in X$, we first define the generalized directional derivative

$$\sigma^\circ(\phi; v) := \limsup_{\varphi\to\phi;\, h\downarrow 0} \frac{\sigma(\varphi + hv) - \sigma(\varphi)}{h},$$

where we suppress the additional $\mathcal{T}$ subscript of σ for simplicity. Note that this limit is well defined because σ is Lipschitz continuous. In order to define a gradient from these directional derivatives, recall that in the differentiable case one can define the gradient $\nabla\sigma \in X^*$ variationally by

$$\partial_v \sigma = \langle \nabla\sigma, v\rangle \quad \text{for all } v \in X,$$

where X^* is the dual space of X and $\langle \cdot, \cdot \rangle$ the corresponding dual pairing. Likewise, in the Lipschitz continuous case we define the generalized gradient by

$$\partial_C \sigma = \{ g \in X^* | \, \sigma^\circ(\phi; v) \geq \langle g, v \rangle \text{ for all } v \in X \}.$$

In the case that σ is differentiable this reduces to the standard gradient, and in the case that σ is convex to the subgradient.

In the literature on nonsmooth optimization one can find several algorithms to minimize $\sigma_{\mathcal{T}}(\phi)$ based on this generalized gradient. For some first numerical tests, we use a simple subgradient method:

$$\phi^{k+1} = \phi^k + h_k \Delta^k, \qquad \Delta^k \in \partial_C \sigma(\phi^k), \qquad \phi^0 = \mathrm{Id}, \tag{26}$$

where $\mathrm{Id}(x) = x$ is the identity transform. Note that this method does not use the full generalized gradient $\partial_C \sigma$ but just one element of it in each step. This is typical for nonsmooth/convex optimization methods because usually the full generalized gradient is not known. Since nonsmooth optimization problems often have kinks at the minimum itself, we cannot use standard techniques to control the step size and use a fixed rule

$$h_k = \alpha k^\beta, \quad \alpha > 0, \quad 0 < \beta < 1 \tag{27}$$

instead. This simple method converges for convex functions [2, 15] and yields good results in the numerical experiments below. More sophisticated methods including convergence analysis for nonconvex problems are available; see, e.g., [15, 3].

4.2. Global minima? Because the objective function $\sigma_{\mathcal{T}}(\phi)$ is nonconvex in general, we must make sure that we do not end up in a suboptimal local minimum. In this section, we discuss some preliminary ideas to overcome this issue for a simple class of 1D problems. To this end, let us consider piecewise constant functions

$$u(x, \mu) = \begin{cases} u_0 & \text{for } x < x_1(\mu), \\ u_i & \text{for } x_i(\mu) \leq x < x_{i+1}(\mu), \\ u_n & \text{for } x_n(\mu) \leq x \end{cases} \tag{28}$$

with smooth parameter dependent jump locations $x_i(\mu)$. We assume that the order $x_0(\mu) < \cdots < x_n(\mu)$ never changes and that the jump locations are well separated, i.e., there is a constant $L \geq 0$ with $|x_i(\mu) - x_{i-1}(\mu)| \geq L$, $i = 1, \ldots, n$.

Transformed snapshot interpolation and training error. Let us first state the transformed snapshot interpolation for these functions and the optimization problem to find the inner transform. Because $u(x, \mu)$ is piecewise constant in x, for transforms $\phi(\mu, \eta)$ that perfectly align the discontinuities the transformed snapshots $v_\mu(x, \eta)$ are constant in η. Therefore, it is sufficient to confine ourselves to one single snapshot for the outer interpolation, say $u(x, \mu_0)$ with node $\mathcal{P}_n = \{\mu_0\}$ so that we obtain

$$u(x, \mu) = v_\mu(x, \mu_0) = u(\phi(\mu, \mu_0), \mu_0). \tag{29}$$

It follows that we have to compute inner transforms $\phi(\mu, \mu_0)$ for all $\mu \in \mathcal{P}$ and the single fixed node μ_0. To this end, we assume to know additional training snapshots $u(\cdot, \mu_1), \ldots, u(\cdot, \mu_m)$ for interpolation points $\mu_1 < \cdots < \mu_m$ with $\mu_0 \leq \mu_1$ for simplicity. Because we just use on snapshot for the outer interpolation the training error (22) reduces to

$$\sigma_{\mathcal{T}}(\phi) = \sup_{1 \leq i \leq m} \sigma_{\mu_i}(\phi) = \sup_{1 \leq i \leq m} \| u(\cdot, \mu_i) - u(\phi(\mu_i, \mu_0), \mu_0) \|_{L_1(\Omega)}. \tag{30}$$

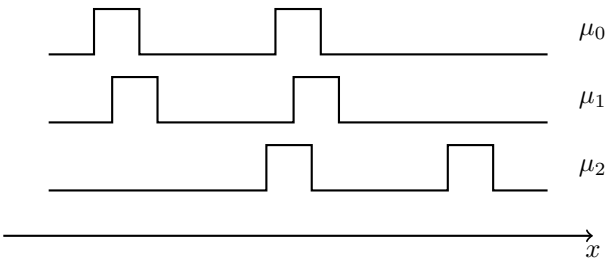

FIG. 4. *Functions* (33) *for various parameters.*

Since all transforms $\phi(\mu_i, \mu_0)$, $1 \leq 1 \leq m$ are independent, we can further simplify this and optimize for each transform $\phi(\mu_i, \mu_0)$ individually, which yields

$$\phi(\mu_i, \mu_0) = \operatorname*{argmin}_{\varphi} \|u(\cdot, \mu_i) - u(\varphi(\cdot), \mu_0)\|_{L_1(\Omega)} \tag{31}$$

for $i = 1, \ldots, n$. Finally, with $\phi(\mu_0, \mu_0)(x) = x$ to ensure the interpolation condition (5), as in (20) we can define the full transform by an interpolation

$$\phi_m(\mu, \mu_0) = \sum_{j=0}^{m} \hat{\ell}_j(\mu)\phi(\mu_j, \mu_0), \tag{32}$$

where $\hat{\ell}_j$ are the Lagrange polynomials for the nodes $\mu_0, \ldots, \mu_m$.

Counterexample: Local minima. It remains to solve the m optimization problems (31). Already for this simple problem, optimization methods relying on local search for updates can easily be mislead into nonoptimal local minima. To this end, consider the example in Figure 4 defined by

$$u(x, \mu) = \chi_{I_1(\mu)} + \chi_{I_2(\mu)}, \qquad I_1(\mu) = [\mu, \mu + 1), \qquad I_2(\mu) = [\mu + 4, \mu + 5), \tag{33}$$

where χ is the characteristic function and the parameter shifts the entire function.

The snapshots μ_0 and μ_2 in Figure 4 are arranged such that the first interval of $u(x, \mu_2)$ intersects the second one of $u(x, \mu_0)$. Therefore the error $\sigma_{\mu_2}(\phi)$ is simply the area of the mismatch between the overlapping intervals plus the area of the two mismatched intervals. Note in particular that any small perturbation of $\phi(\mu_2, \mu_0)$ can align the middle interval and shrink the first mismatched interval whereas the error contribution of the second mismatched interval remains unchanged. Therefore, we cannot expect that optimization schemes exclusively relying on local information converge to a global minimum, which in this case would have zero error.

However, the situation changes when the difference between the snapshot parameter μ_0 and the training parameter μ is small as, e.g., for μ_1 in Figure 4. In this case there are no mismatched intervals and intuitively already simple subgradient based optimization schemes converge to the correct global minimum perfectly aligning the two functions. That this is in fact true is discussed in the following.

Local convexity. We confine ourselves to spatially monotone transforms $\phi(\mu, \mu_0)$ in agreement with our assumption that the order of the jumps $x_i(\mu)$ does not change. Ideally, we search for transforms which exactly match the jump locations

$$\phi(\mu, \mu_0)(x_i(\mu)) = x_i(\mu_0) \Leftrightarrow \phi(\mu, \mu_0)^{-1}(x_i(\mu_0)) = x_i(\mu).$$

Practically, we have to deal with perturbations so that the jumps only approximately match

$$\phi(\mu,\mu_0)^{-1}(x_i(\mu_0)) \approx x_i(\mu).$$

If this matching error is sufficiently small compared to the minimal jump distance L, such that only adjacent intervals of $u(\cdot,\mu)$ and $v_\mu(x,\mu_0)$ overlap, the training error simplifies to

$$\sigma_\mu(\phi) = \sum_{i=1}^{n} |u_i - u_{i-1}||x_i(\mu) - \phi(\mu,\mu_0)^{-1}(x_i(\mu_0))|. \tag{34}$$

This error is convex in $\phi(\mu,\mu_0)^{-1}$, which is not surprising because we assume that we are already close to a minimum. Whereas in principle the convexity allows us to compute optimal transforms this is not yet very practical since it requires very good initial values. However the identity transform $\mathrm{id}(x) = x$ satisfied

$$|x_i(\mu) - id^{-1}(x_i(\mu_0))| = |x_i(\mu) - x_i(\mu_0)|, \tag{35}$$

which is sufficiently small to guarantee the error representation (34) for μ sufficiently close to μ_0 so that id can be used as an initial value in that case.

Formally, in order to obtain a convex optimization problem, we augment the original training error minimization (31) with the following constraints:

$$\begin{aligned} &\sigma_\mu(\phi) \to \min, \\ &\phi(\mu,\mu_0)^{-1} \in C(\Omega) \text{ strictly monotonically increasing}, \\ &|x_i(\mu_0) - \phi(\mu,\mu_0)^{-1}(x_i(\mu_0))| \le B, \quad i = 0,\dots,n \end{aligned} \tag{36}$$

for some constant $B > 0$. Both constraints are clearly convex in $\phi(\mu,\mu_0)^{-1}$. To show convexity of the objective function, note that by the triangle inequality we have

$$|x_i(\mu) - \phi(\mu,\mu_0)^{-1}(x_i(\mu_0))| \le |x_i(\mu) - x_i(\mu_0)| + |x_i(\mu_0) - \phi(\mu,\mu_0)^{-1}(x_i(\mu_0))|.$$

Thus, for μ sufficiently close to μ_0 and B sufficiently small the objective function reduces to (34), which is convex with respect to $\phi(\mu,\mu_0)^{-1}$. According to (35) the identity is allowed by the constraints so that we know a suitable initial value for iterative optimization methods. Moreover for a transform $\hat{\phi}(\mu,\mu_0)$ perfectly aligning the discontinuities, we have

$$\left| x_i(\mu_0) - \hat{\phi}(\mu,\mu_0)^{-1}(x_i(\mu_0)) \right| = |x_i(\mu_0) - x_i(\mu)|,$$

which is also allowed by the constraints. It follows that the optimal error is $\sigma_\mu(\hat{\phi}) = 0$, which is therefore a global minimum.

Note that from the snapshot $u(\cdot,\mu_0)$, one can infer the jump locations $x_i(\mu_0)$. Therefore, in principle we can evaluate the objective function and constraint conditions so that the optimization problem (36) can be solved. In order to avoid an explicit tracking of the discontinuity locations, one can also replace the third line of (36) with the stronger condition $|x - \phi(\mu,\mu_0)^{-1}(x)| \le B$ for all $x \in \Omega$. However, dealing with the inverse $\phi(\mu,\mu_0)^{-1}$ is somewhat impractical so that in the numerical examples below we actually optimize with respect to the $\phi(\mu,\mu_0)$ itself. Although the influence of this change on the convexity is not clear, due to the one to one relation between $\phi(\mu,\mu_0)$ and its inverse, the above analysis shows that this problem has only global minima in a neighborhood of the identity transform $\phi(\mu,\eta)(x) = x$. In addition, in the numerical experiments below we ignore the constraints. The minimizer is in the interior of the constrained domain.

Locality by transitivity. In summary for μ sufficiently close to μ_0, we can reliably find a global minimum of the error $\sigma_\mu(\phi)$ by solving a convex optimization problem, eventually with the identity $\mathrm{id}(x) = x$ as the initial value. But what about larger differences of μ and μ_0 as, e.g., in our counterexample with μ_2 in Figure 4? To this end, recall that the main purpose of the transform $\phi(\mu, \eta)$ is the alignment of jumps and kinks. Thus, if $x(\mu)$ is the location of a jump for parameter μ, we want the transforms to satisfy

$$\phi(\mu, \eta)(x(\mu)) = x(\eta).$$

This condition guarantees that the transformed snapshot $v_\mu(x(\mu), \eta) = u(x(\eta), \mu)$ has a jump at $x(\mu)$ just as the target function $u(x, \mu)$. But this alignment condition is transitive in nature: For three consecutive parameters μ_0, μ_1, and μ_2 we have

$$\big(\phi(\mu_1, \mu_0) \circ \phi(\mu_2, \mu_1)\big)(x(\mu_2)) = x(\mu_0)$$

so that $\phi(\mu_1, \mu_0) \circ \phi(\mu_2, \mu_1)$ correctly aligns the jumps for parameters μ_0 and μ_2. Because this alignment property is a major requirement for the transform $\phi(\mu_2, \mu_0)$, we can define it that way

$$\phi(\mu_2, \mu_0) := \phi(\mu_1, \mu_0) \circ \phi(\mu_2, \mu_1). \tag{37}$$

Let us apply this construction to our example transformed snapshot interpolation (29), where we have one snapshot at μ_0 and m training snapshots at $\mu_1 \leq \cdots \leq \mu_m$. If we enforce transitivity (37), we define

$$\phi(\mu_i, \mu_0) := \phi(\mu_1, \mu_0) \circ \cdots \circ \phi(\mu_i, \mu_{i-1}) \tag{38}$$

so that we are left with the calculation of the "local in μ" transforms $\phi(\mu_i, \mu_{i-1})$. Because they are supposed to align jumps and kinks of $u(\cdot, \mu_i)$ and $u(\cdot, \mu_{i-1})$ we can compute them by solving the optimization problem

$$\phi(\mu_i, \mu_{i-1}) = \operatorname*{argmin}_{\varphi} \|u(\cdot, \mu_i) - u(\varphi(\cdot), \mu_{i-1})\|_{L_1(\Omega)}, \tag{39}$$

which is the same as the original problem (31) with μ_0 replaced by μ_{i-1}. By choosing sufficiently many training snapshots, we can enforce $|\mu_i - \mu_{i-1}|$ to be sufficiently small such that the optimization problem (39) becomes convex. Therefore, we can reliably find global minimizers $\phi(\mu_i, \mu_{i-1})$ and in turn by (37) a transform that perfectly aligns the discontinuities for the parameters μ_0 and μ_i. This leads to a zero training error $\sigma_{\mathcal{T}}(\phi)$ which is therefore a global minimum.

The assumption $\mu_0 < \cdots, \mu_M$ implies that the parameter space is one dimensional. Nonetheless, the argument applies to higher parameter dimensions: If we have a chain of $\mu_1, \mu_2, \ldots, \mu_m$, we can still use (38) to localize the optimization problem. Whereas the chain is canonical in one dimension, in multiple dimensions we have choices, e.g., for a tensor grid of interpolation points we can build a chain by stepwise changing only the first parameter coordinate, then the second, and so on. Other chains are easily conceivable. Further research is required to investigate what influence this choice has on the optimizer and TSI.

In conclusion, for the reconstruction of piecewise constant functions in one dimension from one snapshot, we can find $\phi(\mu, \eta)$ as the global minimum of the training error provided there are sufficiently many training snapshots. The idealized scheme is summarized in Algorithm 1. Of course the argument relies on a couple of assumptions

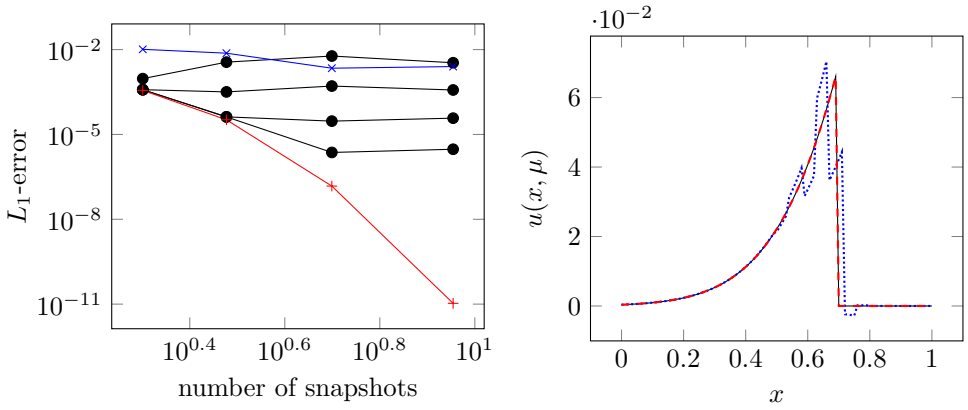

FIG. 5. *Left: Errors of transformed snapshot interpolations of* (40). *Black with dot marks: Snapshots are piecewise linear interpolations with uniform grid sizes* $h = 0.1, 0.01, 0.001, 0.0001$. *Red with* $+$ *marks: Snapshots are exact functions without any approximation. Blue with* $\times$ *marks: Interpolation without transform. Right: Truth solution (black) and interpolation from nine snapshots with (red, dashed) and without (blue dotted) transform.*

that are not true in more general cases. Notably, the functions $u(x, \mu)$ might not be piecewise constant, we eventually want to use more snapshots for the outer interpolation and the parameter and spacial dimensions can be larger than one. Nonetheless, a transitivity property of the transforms is still applicable, which induces some locality in parameter for the minimization of the training error. How to make use of this and to what extend this is helpful for more complicated scenarios is an open problem.

Algorithm 1 Idealized optimization scheme to find the inner transfrom ϕ.

1: For $i = 1, \dots, m$ compute $\phi(\mu_i, \mu_{i-1})$ by solving the optimization problem (36) with μ and μ_0 replaced by μ_i and μ_{i-1}, respectively.
2: For $i = 1, \dots, m$ define $\phi(\mu_i, \mu_0)$ by (37).
3: Define $\phi(\mu, \mu_0)$ by (32).

5. Numerical experiments. In this section, we consider some first numerical tests of the transformed snapshot interpolation (6). First, in section 5.1, a 1D example is presented, where the focus is on the approximation rate, while the inner transforms $\phi(\mu, \eta)$ are given explicitly. Then, in section 5.2 the method is tested with a Riemann problem for the 2D Burgers equation where the solution is explicitly known. Finally, in section 5.3 the method is applied to a shock bubble interaction which is a more challenging test case for the optimizer of the inner transform.

5.1. Cutoff Gaussian. For a first numerical experiment, we consider the parametric function

$$u(x,\mu) := N\left(\frac{x}{0.4+\mu} - 1\right), \qquad N(x) := \begin{cases} 0.4\, e^{-7.0\, x^2} & -1 \le x < -\frac{1}{2}, \\ 0 & \text{else}, \end{cases} \tag{40}$$

which is a scaled and shifted Gaussian, cutoff at a parameter dependent location; see Figure 5. This function is not chosen with a parametric PDE in mind but has a parameter dependent jump and because it is known explicitly it is well suited for analyzing the performance of the transformed snapshot interpolation. For the snapshots we consider to alternatives: First we use the exact function (40), and second

TABLE 1

Errors for example (40) *for different uniform spacial grids and number of snapshots* (n). *The last line contains the maximal* $L_1(\Omega)$ *error of the respective snapshots.*

	Transformed snapshot interpolation					Interpolation
n	0.1	0.01	0.001	0.0001	Exact	Exact
2	$9.4 \cdot 10^{-4}$	$3.83 \cdot 10^{-4}$	$3.81 \cdot 10^{-4}$	$3.81 \cdot 10^{-4}$	$3.61 \cdot 10^{-4}$	$1.02 \cdot 10^{-2}$
3	$3.6 \cdot 10^{-3}$	$3.19 \cdot 10^{-4}$	$4.18 \cdot 10^{-5}$	$4.17 \cdot 10^{-5}$	$3.3 \cdot 10^{-5}$	$7.47 \cdot 10^{-3}$
5	$5.92 \cdot 10^{-3}$	$5.13 \cdot 10^{-4}$	$2.96 \cdot 10^{-5}$	$2.31 \cdot 10^{-6}$	$1.49 \cdot 10^{-7}$	$2.2 \cdot 10^{-3}$
9	$3.43 \cdot 10^{-3}$	$3.74 \cdot 10^{-4}$	$3.77 \cdot 10^{-5}$	$3 \cdot 10^{-6}$	$1.07 \cdot 10^{-11}$	$2.51 \cdot 10^{-3}$
	Maximal $L_1(\Omega)$ error of the snapshots					
	$3.58 \cdot 10^{-3}$	$3.53 \cdot 10^{-4}$	$3.48 \cdot 10^{-5}$	$3.66 \cdot 10^{-10}$		

we interpolate it by piecewise linear functions on a uniform grid. The latter choice should simulate the outcome of PDE solvers which yield similar approximations of the parametric solution. Due to the simplicity of the example, we choose shifts for the inner transform:

$$\phi(\mu, \eta)(x) = x + s(\mu, \eta).$$

Recall that for the transformed snapshot interpolation (16), we only need to know $s(\mu, \eta)$ for interpolation points μ_j, $j \in \hat{\Gamma}_m$, and η_i, $i \in \Gamma_n$. Thus, we can encode ϕ by storing $m \times n$ floating point numbers. For all examples, we choose $\hat{\Gamma}_m = \Gamma_n$ and $\mu_i = \eta_i$. In addition, in this example we only consider the approximation properties of the transformed snapshot interpolation. In order not to interfere with the optimizer for actually finding the transform, we use an explicit formula for $s(\mu, \eta)$ that exactly aligns the jumps and consider the optimizer in the numerical examples below.

The numerical results are summarized in Figure 5. In the case in which we use exact snapshots, we see a more than polynomial convergence rate. Note that after aligning the snapshots, the transformed snapshots $v_\mu(x, \eta)$ are analytic in η so that this behavior is in line with the error bounds of Proposition 2.1. However, for the linearly interpolated snapshots the situation is different. The error first decays and then saturates at a level dependent on the spacial grid resolution. These levels correspond to the maximal error of the snapshots themselves as shown in Table 1. This makes sense because the transformed snapshot interpolation error can hardly be better than the error of the snapshots it relies on.

For a comparison, Figure 5 also contains the error of a simple polynomial interpolation without transform. We see the typical staircasing behavior and an error that is orders of magnitudes worse than the one with previous transform.

5.2. 2D Burgers equation. For a second example, we consider the two dimensional Burgers equation

$$\partial_t u + \nabla \cdot \left(\frac{1}{2} u^2 \boldsymbol{v} \right) = 0$$

with $\boldsymbol{v} = (1, 1)^T$ and the initial condition

$$u(x, y, 0) = \begin{cases} -0.2 & \text{if } x < 0.5 \text{ and } y > 0.5, \\ -1.0 & \text{if } x > 0.5 \text{ and } y > 0.5, \\ 0.5 & \text{if } x < 0.5 \text{ and } y < 0.5, \\ 0.8 & \text{if } x > 0.5 \text{ and } y < 0.5 \end{cases}$$

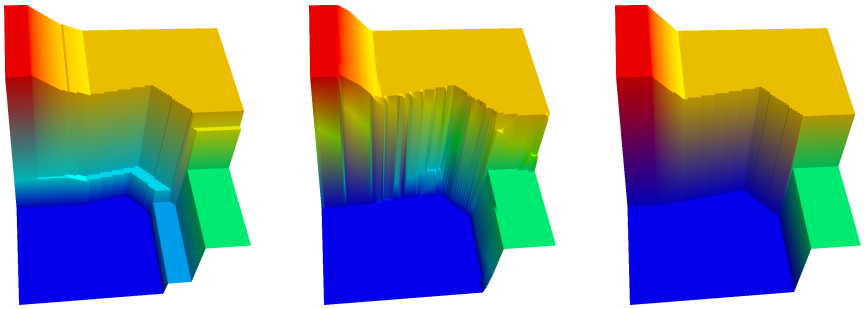

FIG. 6. *Reconstruction of the solution* (41) *for the Burgers equation at the time* $t = 0.45$. *Left: polynomial interpolation; middle: transformed snapshot interpolation; right: exact solution.*

on the unit cube $\Omega = [0,1]^2$. According to [10, 9] the exact solution for this problem is

(41)
$$u(x,y,t) = \begin{cases} -0.2 & \text{if } x < \frac{1}{2} - \frac{3t}{5} \quad \text{and} \quad \begin{cases} y > \frac{1}{2} + \frac{3t}{20}, \\ \text{otherwise}, \end{cases} \\ -1.0 & \text{if } \frac{1}{2} - \frac{3t}{5} < x < \frac{1}{2} - \frac{t}{4} \quad \text{and} \quad \begin{cases} y > -\frac{8x}{7} + \frac{15}{14} - \frac{15t}{28}, \\ \text{otherwise}, \end{cases} \\ -1.0 & \text{if } \frac{1}{2} - \frac{t}{4} < x < \frac{1}{2} + \frac{t}{2} \quad \text{and} \quad \begin{cases} y > \frac{x}{6} + \frac{5}{12} - \frac{5t}{24}, \\ \text{otherwise}, \end{cases} \\ -1.0 & \text{if } \frac{1}{2} + \frac{t}{2} < x < \frac{1}{2} + \frac{4t}{5} \quad \text{and} \quad \begin{cases} y > x - \frac{5}{18t}\left(x + t - \frac{1}{2}\right)^2, \\ \text{otherwise}, \end{cases} \\ \frac{2x-1}{2t} & \\ -1.0 & \text{if } \frac{1}{2} + \frac{4t}{5} < x \quad \text{and} \quad \begin{cases} y > \frac{1}{2} - \frac{t}{10}, \\ \text{otherwise}. \end{cases} \\ 0.8 & \end{cases}$$

For a simple test of the transformed snapshot interpolation, we consider the time t as the parameter of interest so that the snapshots are solutions at various time instances used for the reconstruction of the solution at intermediate times. Note that with this choice of the parameter the solution has exactly the features in question: it has parameter dependent jumps and kinks along nontrivial curves. In addition the exact solution is known which is helpful for an exact assessment of the numerical errors. In order to simulate a numerical solution of the Burger's equation, we sample the snapshots on a 100×100 grid and use a piecewise linear reconstruction from these samples. Also the integrals for evaluating the errors during the optimization of the inner transform ϕ rely on this grid.

Figure 6 shows the results for a reconstruction at time 0.45 from two snapshots at times 0.3 and 0.5 with an additional training snapshot at time 0.4 to define the training error $\sigma_{\mathcal{T}}(\phi)$. For a first test, the inner transforms $\phi(\mu,\eta)$ for $\mu,\eta \in \mathcal{P} := [0.3, 0.5]$ are simply polynomials mapping $\mathbb{R}^2 \to \mathbb{R}^2$. In general this choice does not guarantee that Ω is mapped to itself, but it is easy to enforce that the edges of the rectangular domain are mapped to itself so that small perturbations of the identity are diffeomorphisms. We choose polynomials of degree 3×2 and 2×2 for the x and y-components, respectively. The third order for the x-dependence of the first component is used to ensure that in the figure both the shock in the lower right and the kink in the upper left corner can be aligned simultaneously: With third order polynomials we have four degrees of freedom, two of which are used up by the boundary conditions and the remaining two are used to align the two singularities. For the remaining degrees two seems sufficient. For the optimization of the training error with respect

to ϕ we use a subgradient method (26), (27) with 500 steps of the rather conservative fixed step size

$$h_k = \frac{10^{-3}}{(i+1)^{0.1}}.$$

Compared to a classical polynomial interpolation, the additional transform almost completely removes the artificial staircasing behavior. Also the kinks around the "ramp" in the upper left corner of the figures are much better resolved. The L_1 errors computed by an adaptive quadrature instead of the grid of the snapshots for one to three uniformly spaced intermediate training snapshots are as follows.

L_1-error interpolation	0.0355373675439
L_1-error TSI (1 training snapshot)	0.00739513000396
L_1-error TSI (1 training snapshot)	0.00631731327294
L_1-error TSI (1 training snapshot)	0.0059043074085
maximal snapshot L_1-error	0.0051148730424

In conclusion the additional inner transform reduces the error almost by a factor of five compared to a plain polynomial interpolation. Note that the error of the transformed snapshot interpolation is almost down to the maximal error of the snapshots themselves. As we have verified in Figure 5 for the 1D example of section 5.1 we expect the error to saturate somewhere around this level so that more snapshots or degrees of freedom for the inner transform are not expected to yield major improvements. This is a serious bottleneck for computing convergence rates: Due to the jump discontinuities the maximal error of the snapshots converges with a low rate. The resulting high number of spacial degrees of freedom renders the computation of convergences rates challenging.

5.3. Shock bubble interaction. For a last more complicated example, we consider a compressible Euler simulation of a shock-bubble interaction [22]. Because the code for the above examples relies on piecewise linear interpolation on a uniform grid to represent the snapshots it is straightforward to read them from pictures. For the shock bubble interaction experiments they are frames from a video showing the time evolution of the density provided by [23, 22]. Figure 7 shows the snapshots and the reconstruction at a new time by linear interpolation and transformed snapshot interpolation. Comparable to section 5.2, we simply choose third order polynomials for the inner transform ϕ mapping the edges of the domain to itself. We see that the linear interpolation result basically shows the two bubbles from the original snapshots, whereas the true solution of course just has one bubble. Using the additional transform ϕ, the second picture finds the correct location of the shock and the bubble. Thus despite lots of more complicated fine structure in the pictures, the optimizer reliably finds the correct transform. Having a closer look, the reconstructed bubble appears to be a little blurred. However, we only use third order polynomials which eventually is insufficient for a perfect alignment of the shapes.

Appendix A. Linear width. As an example of the limitations of separation of variables, let us consider their best possible performance for the following simple parametric transport problem:

$$A_\mu u_\mu := u_t + \mu u_x = 0 \qquad \text{for } 0 < t < 1,\ x \in \mathbb{R},$$
$$u(x,0) = g(x) := \begin{cases} 0 & x < 0, \\ 1 & x \geq 0 \end{cases} \qquad \text{for } t = 0$$

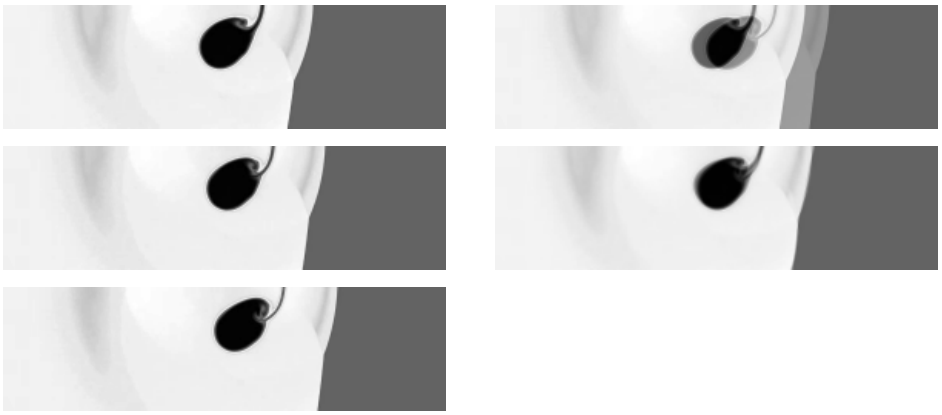

FIG. 7. *Left column: Snapshots for time indices* 3.5, 3.75, *and* 4.0, *where the middle one is only used for the optimization of the transform* ϕ. *Right column: reconstruction at time index* 3.7 *by linear interpolation (top) and transformed snapshot interpolation (bottom).*

with parameter $\mu \in \mathcal{P} = [\mu_{\min}, \mu_{\max}] \subset \mathbb{R}$. Its solution is given by

$$u(x,t) = g(x - \mu t). \tag{42}$$

The typical benchmark for the performance of reduced basis methods is the Kolmogorov n-width

$$d_n(\mathcal{F}) = \inf_{\dim Y = n} \sup_{u \in \mathcal{F}} \inf_{\phi \in Y} \|u - \phi\|_{L_1}$$

of the solution manifold

$$\mathcal{F} := \{u(\cdot, \mu) | \mu \in \mathcal{P}\} = \{g(x - \mu t) | \mu \in \mathcal{P}\}. \tag{43}$$

However, with $X_n := \operatorname{span}\{\psi_i : i = 1, \ldots, n\}$ based on the ψ_i of the separation of variables (1), we conclude that

$$d_n(\mathcal{F}) \leq \sup_{u \in \mathcal{F}} \inf_{\phi \in X_n} \|u(\cdot, \mu) - \phi\|_{L_1} \leq \sup_{u \in \mathcal{F}} \left\| u(\cdot, \mu) - \sum_{i=1}^{n} c_i(\mu) \psi_i \right\|_{L_1}.$$

Therefore, the errors of separation of variables based methods, including but not restricted to reduced basis methods, are lower bounded by the Kolmogorov n-width if the error is measured in the sup-norm with respect to the parameter variable. Of course this measure for the error is not appropriate for all applications, but we use it here to exemplify the limitations of the standard separation of variables based methods.

For our simple model problem the Kolmogorov n-width is bounded according to the following Proposition. The proof is similar to [8]; see also [20].

PROPOSITION A.1. *The Kolmogorov width of the solution manifold* $\mathcal{F}$ *defined in* (43) *satisfies*

$$d_n(\mathcal{F}) \sim n^{-1},$$

i.e., $d_n(\mathcal{F})$ *is equivalent to* n^{-1} *up to a constant.*

Proof. We show the lower bound by comparing the Kolmogorov n-width of the solution manifold $\mathcal{F}$ to the known width of a ball in the L_1-norm. For the construction of this ball, we choose a uniform distribution of $k+1$ snapshots with parameters

$$\mu_i = \mu_{\min} + \frac{i}{k}(\mu_{\max} - \mu_{\min}), \quad i = 0, \dots, k,$$

where k will be chosen below. Note that these specific snapshots are only used for this proof and are not necessarily used in actual approximation methods like, e.g., reduced basis methods. We use the snapshots to define the functions

$$\xi_i = u_{\mu_i} - u_{\mu_{i-1}}, \quad i = 1, \dots, k,$$

which will be the corners of the L_1-ball. From

$$\inf_{y \in Y} \|\xi_i - y\|_{L_1} \le \inf_{y \in Y} \|u_{\mu_i} - y\|_{L_1} + \inf_{y \in Y} \|u_{\mu_{i-i}} - y\|_{L_1} \le 2 \sup_{u \in \mathcal{F}} \inf_{y \in Y} \|u - y\|_{L_1}$$

for any linear space Y of dimension at most n it follows that

$$d_n(\{\xi_0, \dots, \xi_k\}) \le 2 d_n(\mathcal{F}).$$

We complete the set $\{\xi_0, \dots, \xi_k\}$ to a full ball without increasing the Kolmogorov width. To this end assume that λ_i, $i = 1, \dots, k$ satisfy $\sum_{i=1}^k |\lambda_1| \le 1$ and y_i, $i = 1, \dots, k$ are the minimizers of $\inf_{y \in Y} \|\xi_i - y\|_{L_1}$. Then we have

$$\begin{aligned} \inf_{y \in Y} \left\| \sum_{i=1}^k \lambda_i \xi_i - y \right\|_{L_1} &\le \left\| \sum_{i=1}^k \lambda_i \xi_i - \sum_{i=1}^k \lambda_i y_i \right\|_{L_1} \\ &\le \sum_{i=1}^k |\lambda_i| \|\xi_i - y_i\|_{L_1} \le d_n(\{\xi_0, \dots, \xi_k\}) \le 2 d_n(\mathcal{F}) \end{aligned}$$

for any space Y of dimension smaller than n realizing the Kolmogorov width of $\mathcal{F}$. It follows that for the set

$$\mathcal{F}' = \left\{ \sum_{i=1}^k \lambda_i \xi_i : \sum_{i=1}^k |\lambda_1| \le 1 \right\}$$

we have

$$d_n(\mathcal{F}') \le 2 d_n(\mathcal{F}). \tag{44}$$

Next, we show that $\mathcal{F}'$ is in fact an L_1-ball. To this end, note that according to the exact solution (42) the functions ξ_i take the value one on disjoint triangles and zero else. The area of the triangles is $h/2$ with $h = (\mu_{\max} - \mu_{\min})/k$, because the time t is bounded between $0 < t < 1$. Thus, we have

$$\|\xi_i\|_{L_1} = \frac{h}{2}.$$

It follows that

$$\left\| \sum_{i=1}^k \lambda_i \xi_i \right\|_{L_1} = \sum_{i=1}^k |\lambda_i| \|\xi_i\|_{L_1} = \frac{h}{2} \sum_{i=1}^k |\lambda_i|,$$

so that

$$\mathcal{F}' = \operatorname{span}\{\xi_1, \dots, \xi_k\} \cap B_{L_1}^{h/2},$$

where $B_{L_1}^{h/2}$ is the L_1-ball with radius $h/2$. Choosing $k = 2n$, we obtain the Kolmogorov width

$$d_n(\mathcal{F}') = \frac{h}{2};$$

see, e.g., [20]. Using (44) and $h \sim 1/n$ completes the proof of the lower bound.

In order to prove the upper bound, note that $u_\mu - u_{\mu_i}$ is one on a triangular domain and zero otherwise, where u_{μ_i} are the snapshots used in the proof of the lower bound. Calculating the area of the triangle as before, this yields

$$\|u_\mu - u_{\mu_i}\|_{L_1} \le \frac{h}{2}$$

for the parameter μ_i closest to μ. Thus, a piecewise constant approximation by $u_{\mu_0}, \dots, u_{\mu_k}$ yields the upper bound of the proposition. ∎

Appendix B. Inner transforms by characteristics. One idea that comes to mind for the construction of the inner transforms ϕ is to somehow make use of characteristics. But this approach causes difficulties, as we shall discuss in this section, for the Riemann problem for the Burgers equation. Let us choose the height of the jump in the initial condition as parameter, which yields the parametric PDE

$$u_t + \left(\frac{1}{2}u^2\right)_x = 0 \qquad \text{in } \mathbb{R} \times \mathbb{R}^+,$$
$$u(x,0) = g_\mu(x) = \begin{cases} \mu & x \le 0, \\ 0 & x > 0 \end{cases} \qquad \text{for } t = 0.$$

In addition, we assume that $\mu > 0$ so that the solution has a shock along the curve $\frac{1}{2}\mu t$. To write down an explicit solution formula, let $\chi(x,t)$ be the origin (at time $t = 0$) of the characteristic passing through the point (x,t). It is easily seen to be

$$\chi_\mu(x,t) = \begin{cases} x - \mu t, & x \le \frac{1}{2}\mu t, \\ x, & x > \frac{1}{2}\mu t. \end{cases} \tag{45}$$

Because $u(x,t,\mu)$ is constant along characteristics, we obtain

$$u(x,t,\mu) = g_\mu(\chi_\mu(x,t)). \tag{46}$$

The previous discussion aside, a simple idea for an approximation scheme is to encode or approximate $g_\mu(x)$ and the characteristics $\chi_\mu(x,t)$ and then use the exact solution formula (46) to reconstruct $u(x,t,\mu)$. In spirit this is similar to our original idea (4), where the snapshots $u(\cdot,\eta)$ are replaced by $g_\eta(\cdot)$ and the transform $\phi(\mu,\eta)$ by the characteristic χ_μ. However, from the explicit formula (45) for χ_μ, we see that this function has a parameter dependent jump. Thus, in general, we have to face the same difficulties for approximating the parameter dependent characteristic $(x,t,\mu) \to \chi_\mu(x,t)$ as for the original solution $(x,t,\mu) \to u(x,t,\mu)$ so that there is no progress with respect to this issue.

If we want to use characteristics to define the inner transform $\phi(\mu,\eta)$ of the transformed snapshot interpolation, the problems are even more complicated. As for (46) we can follow the characteristics backward in time, but because we do not evaluate

the initial condition g_μ but a snapshot $u(x,t,\eta)$, we then follow the characteristics forward in time with a different parameter. To this end, let $\varphi_\mu(y,t)$ be the position of the characteristic at time t, starting at the initial position y at time $t=0$:

$$\varphi_\mu(y,t) = \begin{cases} y+\mu t, & y \le -\frac{1}{2}\mu t, \\ \frac{1}{2}\mu t, & -\frac{1}{2}\mu t \le y < \frac{1}{2}\mu t, \\ y, & \frac{1}{2}\mu t < y. \end{cases} \tag{47}$$

Then we can transform one solution for parameter η into a solution for parameter μ by

$$u(x,t,\mu) = g_\mu(\chi_\mu(x,t)) = \frac{\mu}{\eta} g_\eta(\chi_\mu(x,t)) = \frac{\mu}{\eta} u(\varphi_\eta(\chi_\mu(x,t),t),t,\eta). \tag{48}$$

However, this formula is only correct for $\mu \ge \eta$. The reason is that the interval of points at $t=0$ that eventually end up in the shock at time t is strictly larger for larger parameters. Thus, for $\mu < \eta$ there is a interval I around the shock location of μ for which $\chi_\mu(I,t)$ is mapped into the shock location $\frac{1}{2}\eta$ by the forward characteristic φ_η. Thus, the right-hand side of (48) has only one single value in the interval I or is undefined whereas the left-hand side has to different values and the formula is thus not correct.

Nonetheless, for $\mu \ge \eta$, we can define the transform

$$\phi(\mu,\eta)(x,t) = \varphi_\eta(\chi_\mu(x,t),t) \tag{49}$$

so that by (48) the transformed snapshot is

$$v_\mu(x,t,\eta) = u(\phi(\mu,\eta)(x,t),\eta) = u(\varphi_\eta(\chi_\mu(x,t),t),t,\eta) = \frac{\eta}{\mu} u(x,t,\mu),$$

which is clearly smooth and in fact even linear in η. Using $\mu \ge \eta$ it is easy to verify that $\phi(\mu,\eta)$ is

$$\phi(\mu,\eta)(x,t) = \begin{cases} x-(\mu-\eta)t, & x \le \frac{1}{2}\mu t, \\ x, & x > \frac{1}{2}\mu t. \end{cases}$$

Note that for our approximation scheme (6) we need to know the function $(x,\mu) \to \phi(\mu,\eta)(x)$ for finitely many $\eta \in \mathcal{P}$. Again, this function has a μ dependent jump so that its approximation poses the same difficulties already encountered for $u(x,t,\mu)$ itself.

However, we are not obliged to use the transform (49) based on characteristics. By noting that the shock location is $\frac{1}{2}\mu t$, simply shifting the whole solution in x-direction by

$$\phi(\mu,\eta)(x,t) = x + \frac{1}{2}(\eta-\mu)t \tag{50}$$

aligns the shocks, i.e., the transformed snapshot $u(\phi(\mu,\eta)(x,t),t,\eta)$ has its shock in the location $\frac{1}{2}\mu t$, which is the shock location for parameter μ. In addition the interpolation condition (5) is obviously satisfied and ϕ is linear so that it is exactly reproduced by an interpolation of type (20). Note that by (45) and (46) the parametric solution is

$$u(x,t,\mu) = \begin{cases} \mu, & x \le \frac{1}{2}\mu t, \\ 0, & x > \frac{1}{2}\mu t, \end{cases}$$

so that the transformed snapshot becomes

$$v_\mu(x,t,\eta) = \begin{cases} \eta, & x \le \frac{1}{2}\mu t \\ 0, & x > \frac{1}{2}\mu t \end{cases} = \frac{\eta}{\mu} u(x,t,\mu),$$

which is the same as for the characteristics based transform, but now for all $\mu, \eta \in \mathcal{P}$. Recall that the error estimate of Proposition 2.1 just requires smoothness with respect to η, which is obviously the case.

In summary, although it might be tempting to use characteristics to construct inner transforms ϕ, this approach is problematic because the resulting transforms are as difficult to approximate as the solution itself. On the other hand, if we do not insist on characteristics, for the given example it is fairly easy to construct transforms that satisfy our needs.

REFERENCES

[1] R. Abgrall and D. Amsallem, *Robust Model Reduction by l_1-norm Minimization and Approximation via Dictionaries: Application to Linear and Nonlinear Hyperbolic Problems*, http://arxiv.org/abs/1506.06178, 2015.

[2] D. P. Bertsekas, *Nonlinear Programming,* 2nd ed., Athena Scientific, Belmont, MA, 1999.

[3] J. V. Burke, A. S. Lewis, and M. L. Overton, *A robust gradient sampling algorithm for nonsmooth, nonconvex optimization*, SIAM J. Optim., 15 (2005), pp. 751–779.

[4] F. Clarke, *Functional Analysis, Calculus of Variations and Optimal Control*, Grad. Texts in Math. 264, Springer-Verlag, New York, 2013.

[5] P. Constantine and G. Iaccarino, *Reduced Order Models for Parameterized Hyperbolic Conservations Laws with Shock Reconstruction*, Technical report, Stanford Center for Turbulence Research Annual Research Briefs, 2012.

[6] W. Dahmen, *How to Best Sample a Solution Manifold?*, Technical report, IGPM preprint 416, 2015.

[7] W. Dahmen, C. Plesken, and G. Welper, *Double greedy algorithms: Reduced basis methods for transport dominated problems*, ESAIM Math. Model. Numer. Anal., 48 (2014), pp. 623–663.

[8] D. L. Donoho, *Sparse components of images and optimal atomic decompositions*, Constr. Approx., 17 (2001), pp. 353–382.

[9] N. Gerhard and S. Müller, *Adaptive multiresolution discontinuous Galerkin schemes for conservation laws: Multi-dimensional case*, Comput. Appl. Math., (2014), pp. 1–29.

[10] J.-L. Guermond, R. Pasquetti, and B. Popov, *Entropy viscosity method for nonlinear conservation laws*, J. Comput. Phys., 230 (2011), pp. 4248–4267.

[11] 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, MD, 2008.

[12] B. Haasdonk and M. Ohlberger, *Reduced basis method for finite volume approximations of parametrized linear evolution equations*, ESAIM Math. Model. Numer. Anal., 42 (2008), pp. 277–302.

[13] G. Iaccarino, P. Pettersson, J. Nordstr, and J. Witteveen, *Numerical methods for uncertainty propagation in high speed flows*, in Proceedings of the European Conference on Computational Fluid Dynamics ECCOMAS CFD 2010, J. C. F. Pereira and A. Sequeira, eds., 2010.

[14] J. D. Jakeman, A. Narayan, and D. Xiu, *Minimal multi-element stochastic collocation for uncertainty quantification of discontinuous functions*, J. Comput. Phys., 242 (2013), pp. 790–808.

[15] K. C. Kiwiel, *Methods of Descent for Nondifferentiable Optimization*, Lecture Notes in Math. 1133, Springer-Verlag, New York, 1985.

[16] K. Kunisch and S. Volkwein, *Galerkin proper orthogonal decomposition methods for parabolic problems*, Numer. Math., 90 (2001), pp. 117–148.

[17] K. Kunisch and S. Volkwein, *Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics*, SIAM J. Numer. Anal., 40 (2002), pp. 492–515.

[18] M. Loève, *Probability Theory,* I, 4th ed., Grad. Texts in Math. 46, Springer-Verlag, New York, 1977.

[19] M. Loève, *Probability Theory,* II, 4th ed., Grad. Texts in Math. 46, Springer-Verlag, New York, 1994.
[20] G. G. Lorentz, M. v. Golitschek, and Y. Makovoz, *Constructive Approximation: Advanced Problems*, Grundlehren Math. Wiss. 304, Springer-Verlag, New York, 1996.
[21] A. Narayan, C. Gittelson, and D. Xiu, *A stochastic collocation algorithm with multifidelity models*, SIAM J. Sci. Comput., 36 (2014), pp. A495–A521.
[22] M. Nazarov, *Entropy Viscosity for High Order Finite Elements*, in preparation, 2015.
[23] M. Nazarov, *private communication*, 2015.
[24] 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 (2009), pp. 157–185.
[25] M. Ohlberger and S. Rave, *Nonlinear reduced basis approximation of parameterized evolution equations via the method of freezing*, C. R. Math., 351 (2013), pp. 901–906.
[26] P. Pacciarini and G. Rozza, *Stabilized reduced basis method for parametrized advection–diffusion PDEs*, Comput. Methods Appl. Mech. Engrg., 274 (2014), pp. 1–18.
[27] A. Patera and G. Rozza, *Reduced Basis Approximation and a Posteriori Error Estimation for Parametrized Partial Differential Equations*, Version 1.0, MIT 2006–2007, to appear, MIT Pappalardo Grad. Monogr. Mech. Eng., 2006–2007.
[28] G. Rozza, D. Huynh, and A. Patera, *Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations*, Arch. Comput. Methods Eng., 15 (2008), pp. 229–275.
[29] W. Schoutens, *Stochastic Processes and Orthogonal Polynomials*, Lecture Notes in Statist. 146, Springer-Verlag, New York, 2000.
[30] S. Sen, K. Veroy, D. Huynh, S. Deparis, N. Nguyen, and A. Patera, *"Natural norm" a posteriori error estimators for reduced basis approximations*, J. Comput. Phys., 217 (2006), pp. 37–62.
[31] L. Sirovich, *Turbulence and the dynamics of coherent structures Part* I: *Coherent Structures*, Quart. Appl. Math., 45 (1987), pp. 561–571.
[32] L. Sirovich, *Turbulence and the dynamics of coherent structures Part* II: *Symmetries and Transformations*, Quart. Appl. Math., 45 (1987), pp. 573–582.
[33] L. Sirovich, *Turbulence and the dynamics of coherent structures Part* III: *Dynamics and Scaling*, Quart. Appl. Math., 45 (1987), pp. 583–590.
[34] T. Taddei, S. Perotto, and A. Quarteroni, *Reduced basis techniques for nonlinear conservation laws*, ESAIM Math. Model. Numer. Anal., 49 (2015), pp. 787–814.
[35] N. Wiener, *The Homogeneous Chaos*, Amer. J. Math., 60 (1938), pp. 897–936.
[36] D. Xiu and G. E. Karniadakis, *The Wiener–Askey polynomial chaos for stochastic differential equations*, SIAM J. Sci. Comput., 24 (2002), pp. 619–644.
[37] M. Yano, A. T. Patera, and K. Urban, *A space-time hp-interpolation-based certified reduced basis method for Burgers' equation*, Math. Models Methods Appl. Sci., 24 (2014), pp. 1903–1935.
[38] W. P. Ziemer, *Weakly Differentiable Functions*, Grad. Texts in Math. 120, Springer-Verlag New York, 1989.