

Deep neural networks on diffeomorphism groups for optimal shape reparameterization

Elena Celledoni · Helge Glöckner ·
Jørgen Riseth · Alexander Schmeding

August 31, 2023

Abstract One of the fundamental problems in shape analysis is to align curves or surfaces before computing geodesic distances between their shapes. Finding the optimal reparametrization realizing this alignment is a computationally demanding task, typically done by solving an optimization problem on the diffeomorphism group. In this paper, we propose an algorithm for constructing approximations of orientation-preserving diffeomorphisms by composition of elementary diffeomorphisms. The algorithm is implemented using PyTorch, and is applicable for both unparametrized curves and surfaces. Moreover, we show universal approximation properties for the constructed architectures, and obtain bounds for the Lipschitz constants of the resulting diffeomorphisms.

Keywords optimal reparametrization · shape analysis · deep learning diffeomorphism group

Mathematics Subject Classification (2000) 65K1 · 58d15 · 46T10

Elena Celledoni
Department of Mathematics, NTNU, Trondheim
E-mail: elena.celledoni@ntnu.no

Helge Glöckner
Institut für Mathematik, Universität Paderborn
E-mail: glockner@math.uni-paderborn.de

Jørgen Riseth
Department of Numerical Analysis and Scientific Computing, Simula Research Laboratory
Department of Mathematics, University of Oslo
E-mail: jorgenr@simula.no

Alexander Schmeding
Department of Mathematics, NTNU, Trondheim
E-mail: alexander.schmeding@ntnu.no

1 Introduction

The geometric shape is of major importance in object recognition. The field of shape analysis gives a formal definition of the shape of an object, and provides tools to compare these shapes. One way to represent objects mathematically is to consider their contours as parametric curves or surfaces defined on some compact domain $\Omega \subseteq \mathbb{R}^n$. Since the same shape may be outlined by different parametric curves that are equivalent under composition by some diffeomorphism, one typically have to solve an optimization problem on the group of orientation preserving diffeomorphisms, $\text{Diff}^+(\Omega)$, to compute shape distances.

In this article, we propose an algorithm for solving optimization problems on $\text{Diff}^+(\Omega)$ by composition of elementary diffeomorphisms. Using concepts from deep learning, we restate the reparametrization problem such that it may be solved by training a residual neural network. This approach admits an effective, unified framework for optimal reparametrization of both curves and surfaces. We motivate the approach with results on infinite-dimensional Lie groups of diffeomorphisms by describing global charts for certain diffeomorphism groups, thus implying universal approximation properties. Furthermore, we establish a priori estimates for the composition of multiple diffeomorphisms. These estimates are of independent interest beyond the applications considered in the present paper. Finally, we provide an implementation of the algorithm based on the PyTorch-framework [31] and test it for several reparametrization problems. The source code for our implementation of the algorithms is available at <https://github.com/jorgenriseth/funcshape.git>¹

1.1 Related work

The methods proposed in this article were developed in the framework of shape analysis, but have their foundations in optimal control and deep neural networks. Our approach is a so-called *learning-free* method, which does not attempt to draw conclusions to unseen instances based on features learned from big data sets, but rather attempt to find an alignment between a single pair of shapes. As such, the method is conceptually similar to that of a *deep image prior* [38], where untrained deep convolutional networks are used for image reconstruction tasks, such as denoising, based on a single image.

The neural network is structured to have inherently encoded some of the important properties of diffeomorphisms, such as invertibility. To achieve this, we base the network structure on residual neural networks [19], where each layer is of the form $x \mapsto x + f(x)$. Here $f : \mathbb{R}^d \rightarrow \mathbb{R}^d$ depends on parameters, e.g. for so-called dense layers $f(x) := \sigma(Ax + b)$, where A is a $d \times d$ matrix, $b \in \mathbb{R}^d$ and σ is a scalar nonlinearity applied componentwise.

¹ This repository will be the subject of change following any feedback during the review process. At the point of publication this url will be replaced by one pointing to a fixed version.

Residual networks may further be related to approximations of diffeomorphisms, by interpreting them as flows of ordinary differential equations [16, 39, 6]. Assuming the parameters to be time-dependent functions $A(t)$ and $b(t)$, each layer may be interpreted as a forward Euler discretization of the non-autonomous differential equation

$$\dot{x}(t) = f(x, t), \quad x(0) = x, \quad f(x, t) := \sigma(A(t)x + b(t)), \quad (1.1)$$

$$x \mapsto \psi_L, \quad \psi_k = \psi_{k-1} + hf(\psi_{k-1}, t_k), \quad k = 1, \dots, L, \quad \psi_0 = x,$$

where h is the step-size often taken to be equal to 1, and with $t_k = kh$. For f Lipschitz continuous in x , the Euler method is known to converge on bounded time intervals to the solution of the differential equation as $h \rightarrow 0$. A sufficient condition for the invertibility of the layer map is that hf is Lipschitz continuous, with Lipschitz constant $\text{Lip}(hf) < 1$ (also called 1-Lipschitz). For activation functions σ with bounded derivative, most layer types, e.g. dense layers or convolutional layers, can be made 1-Lipschitz by appropriate scaling [7]. See for example [5] for a concrete strategy for choosing such a scaling.

The necessity of invertible layers in neural networks is not restricted to the ODE-interpretation of residual neural networks. For example, normalizing flows are a class of machine learning models that may be used to artificially generate data [37]. Given m independent observations of n random variables one seeks an estimate of their underlying probability density $\rho(x)$. The approximation of the probability density is achieved by mapping x to a new set of variables $y = \varphi^{-1}(x)$ with known (for example normal) distribution $\mu(y)$. Then

$$\rho(x) = J_y(x)\mu(y(x)),$$

where J_y is the Jacobian of the map φ^{-1} . The map φ^{-1} is built by means of a gradient ascent flow increasing the log-likelihood of the sample data with respect to its density.

While the evolution and growth of the field of deep learning have primarily been driven by empirical results throughout the last decade, there are multiple approximation theorems which provide a theoretical basis for the effectiveness of neural networks. For example, consider the space of one-layer neural networks of the form

$$\psi : [0, 1]^n \rightarrow \mathbb{R}, \quad \psi(x) := \sum_{j=1}^N \alpha_j \sigma(w_j^T x + b_j), \quad (1.2)$$

with parameters $\alpha_j, b_j \in \mathbb{R}$, $\mathbf{w}_j \in \mathbb{R}^n$, and some activation function $\sigma : \mathbb{R} \rightarrow \mathbb{R}$. In 1989, Cybenko [11] proved that such networks are dense in the space of continuous functions with respect to the supremum norm as $N \rightarrow \infty$, provided that σ is continuous and satisfies

$$\sigma(t) \rightarrow \begin{cases} 1 & t \rightarrow +\infty, \\ 0 & t \rightarrow -\infty. \end{cases}$$

In [1], the authors provide a similar approximation result for diffeomorphisms isotopic to the identity, defined on a compact manifold Ω . They show that such diffeomorphisms may be expressed as a finite composition of exponential maps of vector fields that have been scaled by smooth functions, as long as the vector fields belong to a bracket-generating family of functions. In [2], this result is extended to more general maps on Ω .

1.2 Structure of the Article

The article is structured as follows: Section 2 provides background on shape analysis and the definition of the proposed deep-learning approach. Section 3 gives universal approximation results and the theoretical basis for the deep learning approach. Section 4 describes a priori estimates of the C^k -distance between the identity diffeomorphism and the network φ , given in terms of elementary diffeomorphisms. Section 5 provides details on how the proposed algorithm is implemented. Section 6 shows the results of applying the algorithm to test examples involving both parametric curves, surfaces and n -manifolds. Finally, section 7 discusses some of the choices made in this article and potential ideas for further improvements of the algorithm.

2 Background and definition of the proposed numerical method

2.1 Shape space and shape metrics

Orientation-Preserving Diffeomorphisms

Consider the space of immersions from $\Omega = [0, 1]$ into euclidean space \mathbb{R}^d ,

$$\mathcal{P} := \text{Imm}(\Omega, \mathbb{R}^d) = \{c \in C^\infty(\Omega, \mathbb{R}^d) : c'(t) \neq 0, \forall t \in \Omega\}. \quad (2.1)$$

To be able to identify two curves representing the same shape, consider the set of orientation-preserving diffeomorphisms, which consists of monotonically increasing functions from Ω onto itself,

$$\text{Diff}^+(\Omega) = \{\varphi \in C^\infty(\Omega, \Omega) : \varphi(0) = 0, \varphi(1) = 1, \varphi' > 0\}. \quad (2.2)$$

The group $\text{Diff}^+(\Omega)$ is an infinite-dimensional Lie group. Section 3 provides more details on this group. For now, we note that $\text{Diff}^+(\Omega)$ has a canonical right group action on \mathcal{P} by composition,

$$\mathcal{P} \times \text{Diff}^+(\Omega) \rightarrow \mathcal{P}, \quad (c, \varphi) \mapsto c \circ \varphi.$$

We shall refer to this as reparametrization of c by φ . With respect to the reparametrization action, we can now describe shape spaces.

Shape Space

Elements of shape space are interpreted as *unparametrized curves*. The shape space is the set of $\text{Diff}^+(\Omega)$ -orbits

$$\mathcal{S} = \mathcal{P} / \text{Diff}^+(\Omega).$$

Our goal is to obtain geometrically sound deformations between elements in \mathcal{S} and compute distances thereof, by means of a metric $d_{\mathcal{S}}$. Such a metric is typically constructed by defining a reparametrization-invariant metric $d_{\mathcal{P}}$ on \mathcal{P} . Due to reparametrization invariance (cf. [8]),

$$d_{\mathcal{S}}([c_1], [c_2]) = \inf_{\varphi \in \text{Diff}^+(\Omega)} d_{\mathcal{P}}(c_1, c_2 \circ \varphi) \quad (2.3)$$

defines a metric on \mathcal{S} . Hence, to compute the distance between two shapes, we need to solve an optimization problem. The algorithm presented in this article aims to solve this problem.

The SRVT and Q-transform for Curves

One way to define a suitable distance function $d_{\mathcal{P}}$ is by exploiting certain transformations which transform familiar metrics to obtain reparametrization-invariant (Riemannian) metrics. One such transformation is the square-root velocity transform (SRVT) for curves, introduced in [35]. It is given by

$$\mathcal{R}: \mathcal{P} \rightarrow C^\infty(\Omega, \mathbb{R}^d) \setminus \{0\}, \quad c \mapsto \frac{c'}{\sqrt{|c'|}}.$$

It is easy to see that $\mathcal{R}(c \circ \varphi)(t) = \sqrt{|\varphi'(t)|} (\mathcal{R}(c) \circ \varphi)(t)$. Hence we can pull back the L^2 -inner product on function spaces to obtain a reparametrization-invariant distance:

$$d_{\mathcal{P}}(c_1, c_2) := \|\mathcal{R}(c_1) - \mathcal{R}(c_2)\|_{L^2(\Omega, \mathbb{R}^d)} = \left(\int_{\Omega} |\mathcal{R}(c_1)(t) - \mathcal{R}(c_2)(t)|^2 dt \right)^{1/2}.$$

The SRVT was generalized to curves on Lie groups and homogeneous spaces [9]. Another transformation with similar properties as the SRVT, which allows us to construct a different metric, is the so-called *Q-transform* [25, 3]:

$$Q: \mathcal{P} \rightarrow C^\infty(\Omega, \mathbb{R}^d), \quad c(\cdot) \mapsto \sqrt{|c'(\cdot)|} c(\cdot). \quad (2.4)$$

Surfaces embedded in 3D space

To extend the framework for computing shape distances to parametric surfaces embedded in \mathbb{R}^3 , we will assume for simplicity that the surfaces are defined on the unit square $\Omega = [0, 1]^2$. The main differences between shapes of curves and of surfaces pertain the group of diffeomorphisms. Further, we need analogues to the SRVT and Q-transform.

Denote by f_x, f_y the partial derivatives of a parametric surface f and by

$$a_f(\mathbf{x}) = |f_x(\mathbf{x}) \times f_y(\mathbf{x})|$$

the area scaling factor of the surface. Now the pre-shape space for surfaces is

$$\mathcal{P} = \{f \in C^\infty(\Omega, \mathbb{R}) : a_f(x) > 0 \forall x \in \Omega\}. \quad (2.5)$$

The group of orientation-preserving diffeomorphisms consists of smooth invertible maps from Ω onto itself such that the Jacobian determinant J_φ is positive,

$$\text{Diff}^+(\Omega) = \{\varphi \in C^\infty(\Omega, \Omega) : J_\varphi(\mathbf{x}) > 0 \forall \mathbf{x} \in \Omega, \varphi(s) \in \partial\Omega \forall s \in \partial\Omega\}.$$

Its tangent space $T_\varphi \text{Diff}^+(\Omega)$ at φ consists of smooth boundary-respecting vector fields on Ω (see [26]).

The SRNF and Q-transform of Surfaces

The square root normal field (SRNF), defined in [20], is a generalization of the SRVT to surfaces. It is defined by

$$\mathcal{N}(f)(\mathbf{x}) = \sqrt{a_f(\mathbf{x})} \mathbf{n}_f(\mathbf{x})$$

where $\mathbf{n}_f(\mathbf{x})$ is the normal vector field on f . Applied to a reparametrized surface, the transform satisfies the relation

$$\mathcal{N}(f \circ \varphi) = \sqrt{J_\varphi} \mathcal{N}(f) \circ \varphi.$$

Following the same derivation as was given for curves, we may now define a reparametrization-invariant pre-shape distance by

$$d_{\mathcal{P}}(f_1, f_2) = \|\mathcal{N}(f_1) - \mathcal{N}(f_2)\|_{L^2(\Omega, \mathbb{R}^3)},$$

since the factor $\sqrt{J_\varphi}$ cancels the Jacobian determinant introduced by the change of variables in the integral. While the SRNF shares some properties of the SRVT, for example the translation invariance, it is not invertible. In [22], the authors present several examples of surfaces that represent different shapes, but have the same image under the SRNF. However, the SRNF framework has shown some merit in practice as shown in [36] and the references therein.

Alternatively, we can use the Q -transform for surfaces, which was introduced in [23]. The Q -transform for surfaces is defined by

$$Q(f)(\mathbf{x}) = \sqrt{a_f(\mathbf{x})} f(\mathbf{x})$$

and satisfies the same property of reparametrization-invariance with respect to the L^2 -norm.

2.2 The optimization problem on the diffeomorphism group

The metric on shape space is induced by suitable metrics on the pre-shape space. In the approaches sketched above, this metric is the geodesic distance of a Riemannian metric arising as pullback of the L^2 -metric via one of the transformations mentioned. To compute the metric on shape space one needs to find the elements in the $\text{Diff}^+(\Omega)$ -orbits which are closest to each other. However, due to equivariance of the transformations, this can equivalently be formulated as the following optimization problem for the L^2 -distance (exploiting that $(C^\infty(\Omega, \mathbb{R}^d), L^2)$ is a pre-Hilbert space and the set in which we compute the distance is an open (but non-convex) subset of this space). Assume that $f_1, f_2 \in \mathcal{P}$, and we want to compute $d_S([f_1], [f_2])$. Let \mathcal{T} be one of the transforms discussed above and $q_i := \mathcal{T}(f_i), i = 1, 2$. Then we need to solve the following optimization problem:

$$\varphi^* = \arg \inf_{\varphi \in \text{Diff}^+(\Omega)} E(\varphi), \quad E(\varphi) := \|q_1 - \sqrt{\varphi^T}(q_2 \circ \varphi)\|_{L^2}^2. \quad (2.6)$$

Note that the optimization problem has been formulated as the *squared* L^2 -distance. Solving this optimization problem, we can compute the metric distance on shape space if both q_1 and q_2 are contained in a convex subset of the image of the transformation. While this is not always the case (for example, it does not hold for a pair of curves which get mapped by the SRVT to elements q_1 and $-q_1$), the assumption always holds locally for mappings which are not too far apart. In any case, solving efficiently this optimization problem is crucial in shape analysis and the main purpose of the present paper is to provide tools from machine learning for its solution.

2.3 A Riemannian gradient descent method

A gradient descent approach to solve (2.6) is obtained by representing the gradient of the functional E by means of an orthonormal basis of $T_{\text{id}}\text{Diff}^+(\Omega)$. Suppose $f_j \in T_{\text{id}}\text{Diff}^+(\Omega), j = 1, 2, \dots$ is an orthonormal basis of $T_{\text{id}}\text{Diff}^+(\Omega)$ with respect to an inner product $\langle \cdot, \cdot \rangle_{\text{id}}$ on $T_{\text{id}}\text{Diff}^+(\Omega)$. Consider the left multiplication L_φ for some $\varphi \in \text{Diff}^+(\Omega)$ mapping $\psi \mapsto \varphi \circ \psi$. We denote with $dL_\varphi : T_{\text{id}}\text{Diff}^+(\Omega) \rightarrow T_\varphi\text{Diff}^+(\Omega)$ the corresponding derivative mapping at the identity. The gradient of E with respect to $\langle \cdot, \cdot \rangle_{\text{id}}$ is defined as the unique vector field $\text{grad } E$ on $\text{Diff}^+(\Omega)$ satisfying

$$dE|_\varphi(w_\varphi) = \langle dL_\varphi^{-1}(\text{grad } E(\varphi)), dL_\varphi^{-1}w_\varphi \rangle_{\text{id}}, \quad \forall w_\varphi \in T_\varphi\text{Diff}^+(\Omega).$$

We can formally represent $\text{grad } E(\varphi)$ using the basis of $T_{\text{id}}\text{Diff}^+(\Omega)$, by

$$dL_\varphi^{-1}(\text{grad } E(\varphi)) = \sum_{j=1}^{\infty} \lambda_j f_j, \quad (2.7)$$

with $\lambda_j = dE|_\varphi(dL_\varphi(f_j))$.

In practice, the proposed gradient descent algorithm on $\text{Diff}^+(\Omega)$ is obtained by truncating (2.7) and projecting the gradient on the subspace of $T_{\text{id}}\text{Diff}^+(\Omega)$ spanned by a finite number M of basis elements. This gives

$$dL_{\varphi}^{-1}\widehat{\text{grad}}E(\varphi) = \sum_{j=1}^M \lambda_j f_j,$$

and we obtain the following update rule for the gradient descent iteration:

$$\varphi^{(n+1)} = \varphi^{(n)} \circ \left(\text{id} - \eta dL_{\varphi^{(n)}}^{-1}\widehat{\text{grad}}E(\varphi^{(n)}) \right), \quad n = 0, 1, \dots,$$

with $\varphi_0 = \text{id}$, where η is a scalar parameter optimized to speed-up convergence and to guarantee the invertibility of $\varphi^{(n+1)}$. Written in terms of the basis, the gradient descent iteration becomes

$$\varphi^{(n+1)} = \varphi^{(n)} \circ \left(\text{id} - \eta \sum_{j=1}^M \lambda_j^n f_j \right), \quad n = 0, 1, \dots, \quad (2.8)$$

and $\{\lambda_j^n\}_{j=1}^M$ are coefficients determined at each iteration n . In [33], this update rule was shown to be equivalent to an algorithm proposed for reparametrization of surfaces in [24] and to a similar algorithm created for curves in [35].

2.4 Main method: Deep learning of diffeomorphisms

It can be observed that after L iterations of the gradient descent algorithm (2.8), the diffeomorphism $\varphi^{(L)}$ is the composition of L elementary diffeomorphisms defined as the identity plus a weighted sum of M basis functions. This observation may be exploited to create a deep learning approach to the reparametrization problem.

The goal is to find a minimizer of the loss function in (2.6). However, instead of using the whole group $\text{Diff}^+(\Omega)$, we fix some $L, M \in \mathbb{N}$ and restrict ourselves to functions of the form

$$\varphi = \varphi_L \circ \dots \circ \varphi_1, \quad \text{where} \quad \varphi_\ell := \text{id} + \sum_{n=1}^M w_n^\ell f_n, \quad \ell = 1, \dots, L, \quad (2.9)$$

for a set of basis function $\{f_j\}_{j=1}^M \subset T_{\text{id}}\text{Diff}^+(\Omega)$. Here $\{w_j^\ell\}_{j=1}^M$ are coefficients to be found by the optimization algorithm.

The computational graph of the function φ in (2.9) has a very similar structure to that of a deep residual network [19]. Both the input-values x_0 and output-values x_L of the network are points in the domain Ω . The function evaluation is defined by the recursive relation

$$\varphi(x_0) = x_L, \quad x_{\ell+1} = \varphi(x_\ell) = x_\ell + F_{w^\ell}(x_\ell), \quad \ell = 0, \dots, L-1.$$

By using a discrete analogue of (2.6) as the loss function, then the problem of finding an optimal reparametrization between two curves — i.e. finding the optimal coefficients $\{w_j^\ell\}_{j=1}^M$ — coincides with that of training a neural network. To ensure that $\varphi \in \text{Diff}^+(\Omega)$, we need to restrict the set of weights to some subset $\mathcal{W} \subset \mathbb{R}^{ML}$. Further details on this restriction are given in section 5.1.

The increased performance of the deep learning approach (2.9) can be seen in figure 2.1, which compares the proposed method to the gradient descent algorithm. This figure indicates that the deep learning approach converges towards a significantly better solution than gradient descent. A preliminary implementation of this approach and a proof of concept was presented in [33].

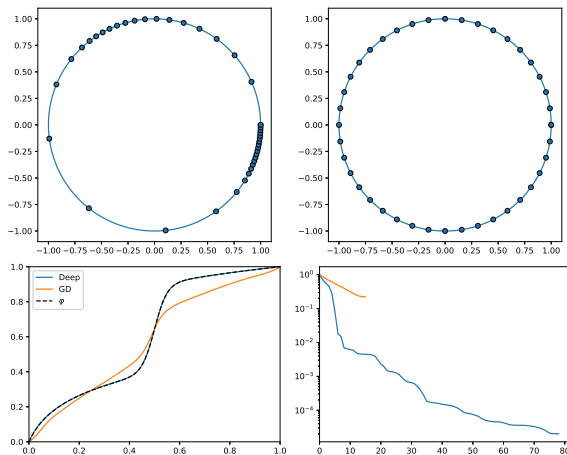


Fig. 2.1 Comparison between the Riemannian gradient descent (GD) and the deep residual network approach for reparametrization of curves. The GD approach used 6 basis functions, whereas the network approach used 6 basis functions and 6 layers. (Top): A curve c_2 (left) and the same curve with a different parametrization $c_1 = c_2 \circ \varphi$ (right). The goal is to reconstruct φ . (Bottom left): The diffeomorphisms found by the different algorithms, compared to the target diffeomorphism φ . The diffeomorphism found by the deep neural network approach is visually indistinguishable from the target. (Bottom right): Convergence plots for the two algorithms, showing the error of the cost function (see section 5.4) as compared to the iteration number of the optimization algorithm. The error is given relative to the initial error.

3 Theoretical justification

This section provides the main theoretical motivation for the proposed deep learning approach for optimization on $\text{Diff}^+(\Omega)$. We shall first recall relevant (infinite-dimensional) Lie group structures on diffeomorphism groups. Then we show that finite compositions of diffeomorphisms of the type $\varphi_\ell = \text{id} + f_\ell$,

$\ell = 1, \dots, L$ with f_ℓ a 1-Lipschitz vector field can be used to describe the whole group of diffeomorphisms of a cube $\Omega = [0, 1]^d$ (Section 3.2). If Ω is, more generally, a compact convex subset of \mathbb{R}^d with dense interior (like a disk $\Omega = \{(x, y) \in \mathbb{R}^2 : x^2 + y^2 \leq r^2\}$ in \mathbb{R}^2), analogous diffeomorphisms generate the group of diffeomorphisms fixing the boundary of Ω (see Section 3.1). We are mostly interested in the case of a cube $[0, 1]^d$; in fact, only the unit interval $[0, 1]$ and square $[0, 1]^2$ are used in the numerical calculations. As explained in Section 2.4, eventually our methodology is implemented first restricting f_ℓ to a finite-dimensional subspace of the Lie algebra of vector fields and then constraining f_ℓ to be 1-Lipschitz, see Section 6 for details. This introduces further approximations which we discuss briefly but do not analyze in detail in the present work.

3.1 Diffeomorphisms fixing the boundary

Lie group structures on diffeomorphism groups of compact manifolds have been studied in many works (see, e.g., [26, 17, 27, 30]), including the case of manifolds with boundary or corners. In this section, we consider a compact convex subset $\Omega \subseteq \mathbb{R}^d$ with dense interior (having in mind the main example $\Omega = [0, 1]^d$). A map on Ω is smooth if it extends to a smooth map on an open subset in \mathbb{R}^d ; we can therefore speak about the group $\text{Diff}(\Omega)$ of all C^∞ -diffeomorphisms of Ω . Following [14], we consider the subgroup

$$\text{Diff}_\partial(\Omega) := \{\phi \in \text{Diff}(\Omega) : \phi(k) = k, \forall k \in \partial\Omega\}$$

of all smooth diffeomorphisms fixing the topological boundary $\partial\Omega$ pointwise. It has been shown in [14] that if we endow $\text{Diff}_\partial(\Omega)$ with the compact-open C^∞ -topology, the group becomes a manifold modeled on the space

$$C_\partial^\infty(\Omega, \mathbb{R}^d) = \{f \in C^\infty(\Omega, \mathbb{R}^d) : f(k) = 0, \forall k \in \partial\Omega\}.$$

This space is an infinite-dimensional locally convex space (with respect to the compact-open C^∞ -topology) and its elements can be identified with vector fields which vanish on the boundary. This structure turns the space $\text{Diff}_\partial(\Omega)$ into an infinite-dimensional Lie group. Using a canonical identification yields $T_{\text{id}} \text{Diff}_\partial(\Omega) = C_\partial^\infty(\Omega, \mathbb{R}^d)$. Further, $\text{Diff}_\partial(\Omega)$ admits a global chart

$$\kappa : \text{Diff}_\partial(\Omega) \rightarrow C_\partial^\infty(\Omega, \mathbb{R}^d), \quad \phi \mapsto \phi - \text{id}_\Omega, \quad (3.1)$$

which arises as a restriction of the map $\kappa : C^\infty(\Omega, \mathbb{R}^d) \rightarrow C^\infty(\Omega, \mathbb{R}^d), f \mapsto f - \text{id}_\Omega$. For vector fields $f \in \kappa(\text{Diff}_\partial(\Omega))$, the inverse $\kappa^{-1}(f) = f + \text{id}_\Omega$ is in $\text{Diff}_\partial(\Omega)$ and we can obtain all elements in $\text{Diff}_\partial(\Omega)$ in this way.

By the preceding, every diffeomorphism fixing the boundary can be expressed as a vector field (vanishing on the boundary) plus the identity. Thus, one only needs to approximate vector fields and can ignore the non-linearity of the infinite-dimensional manifold. However, there are two practical problems for us to solve:

1. Describe the image of the global chart κ .
2. Establish approximation results by sampling from a finite-dimensional subspace of the Lie algebra.

As the global chart is highly dependent on the geometry of Ω , there is no easy comprehensive way to describe its image. However, it is easy to provide sufficient conditions for a vector field to lie inside the image of κ .

Example 3.0.1. Let $\Omega = [a, b]$ be a compact interval. Then a vector field f in $C_{\partial}^{\infty}([a, b], \mathbb{R})$ is in the image of κ if $f'(x) > -1$ for all $x \in [a, b]$ since then $\text{id}_{[a, b]} + f$ will be monotonically increasing.

The situation is more complicated for higher-dimensional sets. We do not have any global information concerning the set Ω then. However, there is a convenient sufficient condition ensuring that a map is in Ω , which we describe now.

Fix any norm on \mathbb{R}^d to calculate Lipschitz constants for functions from $\Omega \subseteq \mathbb{R}^d$ to \mathbb{R}^d , operator norms, and open balls $B_r(x) := \{y \in \mathbb{R}^d : \|y - x\| < r\}$ for $x \in \mathbb{R}^d$ and $r > 0$. Setting

$$\|f\|_{\infty, \text{op}} := \max_{x \in \Omega} \|f'(x)\|_{\text{op}} \quad (3.2)$$

for $f \in C_{\partial}^{\infty}(\Omega, \mathbb{R}^d)$, we obtain a continuous seminorm on $C_{\partial}^{\infty}(\Omega, \mathbb{R}^d)$. Hence

$$\mathcal{U}_1 := \{f \in C_{\partial}^{\infty}(\Omega, \mathbb{R}^d) : \text{Lip}(f) < 1\} = \{f \in C_{\partial}^{\infty}(\Omega, \mathbb{R}^d) : \|f\|_{\infty, \text{op}} < 1\}$$

is an open 0-neighborhood in $C_{\partial}^{\infty}(\Omega, \mathbb{R}^d)$.

Lemma 3.1. *The map $\text{id}_{\Omega} + f$ is an element of $\text{Diff}_{\partial}(\Omega)$ for all $f \in \mathcal{U}_1$. Thus $\mathcal{U}_1 \subseteq \kappa(\text{Diff}_{\partial}(\Omega))$ and it makes sense to consider $\kappa^{-1}(f) = \text{id}_{\Omega} + f$.*

Proof. To keep the notation short, we write $\kappa^{-1}(f) := f + \text{id}_{\Omega}$, though strictly speaking it makes only sense to use the inverse of the chart once the proof has been concluded. Given $f \in \mathcal{U}_1$, the map $\kappa^{-1}(f) : \Omega \rightarrow \mathbb{R}^d$ is injective, since $\kappa^{-1}(f)(x) = \kappa^{-1}(f)(y)$ with $x \neq y$ in Ω would entail that

$$\|x - y\| = \|f(y) - f(x)\| \leq \text{Lip}(f)\|x - y\| < \|x - y\|,$$

contradiction. Since $\kappa^{-1}(f)(x) = x$ for each $x \in \partial\Omega$, we have

$$\kappa^{-1}(f)(\partial\Omega) = \partial\Omega. \quad (3.3)$$

Using the injectivity of $\kappa^{-1}(f)$, we obtain for the interior Ω° of Ω

$$\kappa^{-1}(f)(\Omega^{\circ}) \cap \partial\Omega = \emptyset. \quad (3.4)$$

Since $\|f'(x)\|_{\text{op}} < 1$, we have $\kappa^{-1}(f)'(x) = \text{id}_{\mathbb{R}^d} + f'(x) \in \text{GL}(\mathbb{R}^d)$ for each $x \in \Omega^{\circ}$, whence $\kappa^{-1}(f)|_{\Omega^{\circ}}$ is a local C^{∞} -diffeomorphism at each $x \in \Omega^{\circ}$ (by the Inverse Function Theorem) and hence an open map.

Next, we pick an x_0 in the interior Ω° of Ω . Then the distance

$$r := \min\{\|x_0 - y\| : y \in \partial\Omega\}$$

is positive. Note that

$$B_r(x_0) := \{y \in \mathbb{R}^d : \|x_0 - y\| < r\} \subseteq \Omega^\circ. \quad (3.5)$$

In fact, since $B_r(x_0) \cap \partial\Omega = \emptyset$, the disjoint open sets $B_r(x_0) \cap \Omega^\circ \neq \emptyset$ and $Q := B_r(x_0) \cap (\mathbb{R}^d \setminus \Omega)$ cover $B_r(x_0)$. Since $B_r(x_0)$ is connected, $Q = \emptyset$ follows.

There exists $y_0 \in \partial\Omega$ such that $\|x_0 - y_0\| = r$. Since $f(y_0) = 0$, we have

$$\|f(x_0)\| = \|f(x_0) - f(y_0)\| \leq \text{Lip}(f)\|x_0 - y_0\| < r.$$

Hence

$$\|\kappa^{-1}(f)(x_0) - x_0\| = \|f(x_0)\| < r,$$

whence $\kappa^{-1}(f)(x_0) \in \Omega^\circ$, by (3.5). Since $x_0 \in \Omega^\circ$ was arbitrary,

$$\kappa^{-1}(f)(\Omega^\circ) \subseteq \Omega^\circ$$

is established. Now $\kappa^{-1}(f)(\Omega)$ is compact and hence closed in \mathbb{R}^d . Furthermore, since $\kappa^{-1}(f)(\partial\Omega) = \partial\Omega$, we deduce that

$$\kappa^{-1}(f)(\Omega^\circ) = \kappa^{-1}(f)(\Omega^\circ) \cap \Omega^\circ = \kappa^{-1}(f)(\Omega) \cap \Omega^\circ$$

is closed in Ω° . Since Ω° is connected and its non-empty subset $\kappa^{-1}(f)(\Omega^\circ)$ is both open and closed in Ω° , we must have $\kappa^{-1}(f)(\Omega^\circ) = \Omega^\circ$. Being a bijective local C^∞ -diffeomorphism, the map $\kappa^{-1}(f)|_{\Omega^\circ} : \Omega^\circ \rightarrow \Omega^\circ$ is a C^∞ -diffeomorphism. Since $f|_{\partial\Omega} = 0$, we have $\kappa^{-1}(f)|_{\partial\Omega} = \text{id}_{\partial\Omega}$, whence $\kappa^{-1}(f)(\Omega) = \Omega$. It only remains to show that $\kappa^{-1}(f) : \Omega \rightarrow \Omega$ is a C^∞ -diffeomorphism. By the preceding, $\kappa^{-1}(f) : \Omega \rightarrow \Omega$ is a continuous bijective self-map of the compact topological space Ω and hence a homeomorphism. Note that $\kappa^{-1}(f)'(x) = \text{id}_{\mathbb{R}^d} + f'(x) \in \text{GL}(\mathbb{R}^d)$ for all $x \in \Omega$. If $k \in \mathbb{N}_0$ and $\kappa^{-1}(f)$ is a C^k -diffeomorphism, then

$$(\kappa^{-1}(f)^{-1})'(x) = (\kappa^{-1}(f)'(\kappa^{-1}(f)^{-1}(x)))^{-1}$$

is a $\text{GL}(\mathbb{R}^d)$ -valued C^k -function of $x \in \Omega$, entailing that $\kappa^{-1}(f)^{-1}$ is C^{k+1} . Thus $\kappa^{-1}(f)^{-1}$ is smooth and thus $\kappa^{-1}(f) \in \text{Diff}_\partial(\Omega)$. \square

Remark 3.2. The same reasoning may be used to show that, for each $\ell \in \mathbb{N}$, the set $\{f \in C_\partial^\ell(\Omega, \mathbb{R}^d) : \text{Lip}(f) < 1\}$ is an open 0-neighborhood in $C_\partial^\ell(\Omega, \mathbb{R}^d)$ and $\text{id}_\Omega + f \in \text{Diff}_\partial^{C^\ell}(\Omega)$ for all f in this 0-neighborhood.

The main disadvantage of the group $\text{Diff}_\partial(\Omega)$ in the applications we wish to investigate is that its elements fix the boundary pointwise. Depending on the application, this can be quite unnatural. Hence, we also consider the full group of C^∞ -diffeomorphisms for the special case of a cube $[0, 1]^d \subseteq \mathbb{R}^d$.

3.2 Diffeomorphisms of a cube

In this section, we exclusively consider the cube $\Omega := [0, 1]^d$. Then the group $\text{Diff}(\Omega)$ of smooth diffeomorphisms is an infinite-dimensional Lie group (cf. [26]). We shall follow the complementary approach of [13], where diffeomorphism groups of convex polytopes are discussed. Our goal is to generalize Lemma 3.1 to the diffeomorphism group of a cube.

For the cube, the diffeomorphism group $\text{Diff}(\Omega)$ is an infinite-dimensional Lie group modeled on the Fréchet space $C_{\text{str}}^\infty(\Omega, \mathbb{R}^d) \subseteq C^\infty(\Omega, \mathbb{R}^d)$ of smooth vector fields

$$f = (f_1, \dots, f_n): \Omega \rightarrow \mathbb{R}^d$$

which are *tangent to the boundary* in the sense that $f_j(x_1, \dots, x_d) = 0$ whenever $x_j = 0$ or $x_j = 1$ (such vector fields are also called boundary respecting, or stratified vector fields). For example, a smooth vector field $f: [0, 1] \rightarrow \mathbb{R}$ is tangent to the boundary if it vanishes at the vertices 0 and 1. A smooth vector field $f = (f_1, f_2): [0, 1]^2 \rightarrow \mathbb{R}^2$ is tangent to the boundary if $f_1(x, y)$ vanishes whenever $x = 0$ or $x = 1$, and $f_2(x, y)$ vanishes whenever $y = 0$ or $y = 1$. The space of stratified vector fields is a closed vector subspace of $C^\infty(\Omega, \mathbb{R}^d)$. As in every Lie group, the connected component $\text{Diff}(\Omega)_0$ of the neutral element id_Ω is an open subgroup of $\text{Diff}(\Omega)$. We recall from [13]:

Lemma 3.3. *Let $\Omega = [0, 1]^d$. Then $\kappa: \text{Diff}(\Omega)_0 \rightarrow C_{\text{str}}^\infty(\Omega, \mathbb{R}^d), \phi \mapsto \phi - \text{id}_\Omega$ has open image and is a chart of $\text{Diff}(\Omega)$, when considered as a map onto the image. In particular, $\text{Diff}_\partial(\Omega)$ is a Lie subgroup of $\text{Diff}(\Omega)$.*

Again, it is not straightforward to determine the image of the chart κ . But we can extend the sufficient criterion encountered for boundary-fixing diffeomorphisms. To this end, we consider the open subset

$$\mathcal{V}_1 = \{f \in C_{\text{str}}^\infty(\Omega, \mathbb{R}^d) : \text{Lip}(f) < 1\}.$$

We prove the following (see [13, Proposition 1.5] for a generalized version):

Proposition 3.4. *For every $f \in \mathcal{V}_1$, the map $f + \text{id}_\Omega$ is contained in $\text{Diff}(\Omega)$. Hence $\mathcal{V}_1 \subseteq \kappa(\text{Diff}(\Omega)_0)$.*

Proof. The proof is by induction on $d \in \mathbb{N}$. If $d = 1$, then $C_{\text{str}}^\infty(\Omega, \mathbb{R}) = C_{\partial}^\infty(\Omega, \mathbb{R})$ and the assertion follows from Lemma 3.1. Now assume that $d \geq 2$. Fix $f \in \mathcal{V}_1$. Arguing as in the proof of Lemma 3.1, we see that $\text{id}_\Omega + f$ is injective. Note that each $(d - 1)$ -dimensional face F of Ω is diffeomorphic to $[0, 1]^{d-1}$ via a translation; for example, the upper face $[0, 1] \times \{1\}$ of the square $[0, 1]^2$ can be translated to $[0, 1] \times \{0\} \cong [0, 1]$. Hence $(\text{id}_\Omega + f)(F) = (\text{id}_F + f|_F)(F) = F$ by induction. As the boundary $\partial\Omega$ is the union of the $(d - 1)$ -dimensional faces, we deduce that $(\text{id}_\Omega + f)(\partial\Omega) = \partial\Omega$. Thus (3.3) holds. Next, we note that $\text{id}_\Omega + f$ has invertible derivative $f'(x)$ at each $x \in \Omega^\circ$, whence it is an open map on the interior Ω° . To see that $(\text{id}_\Omega + f)(\Omega) \subseteq \Omega$, consider the projection pr_j onto the j th component for $j \in \{1, \dots, d\}$. By compactness of Ω , the minimum $m_j := \min_{y \in \Omega} (\text{pr} \circ (\text{id}_\Omega + f))(y)$ exists. Since

$\text{id}_{\Omega^\circ} + f|_{\Omega^\circ}$ is an open map, the minimum cannot be attained on the interior of the cube. Thus m_j is attained on the boundary $\partial\Omega$. As the boundary is mapped to itself and does not contain points with negative coordinates, we conclude that $m_j \geq 0$ for every j , whence $x_j + f_j(x) \geq 0$ for each $x = (x_1, \dots, x_d) \in \Omega$. Considering the maximum instead, we conclude that also $x_j + f_j(x) \leq 1$, whence $x + f(x) \in \Omega$ for all $x \in \Omega$. Thus $\text{id}_\Omega + f$ maps the sets Ω and $\partial\Omega$ into themselves, respectively. Arguing as in the proof of Lemma 3.1, we see that $(\text{id}_\Omega + f)(\Omega^\circ)$ is both open and closed in the connected set Ω° , whence $(\text{id}_\Omega + f)(\Omega^\circ) = \Omega^\circ$. Then $(\text{id}_\Omega + f)(\Omega) = (\text{id}_\Omega + f)(\Omega^\circ) \cup (\text{id}_\Omega + f)(\partial\Omega) = \Omega^\circ \cup \partial\Omega = \Omega$, showing that $\text{id}_\Omega + f$ is surjective as a map to Ω and hence bijective. The proof can now be completed as in the proof of Lemma 3.1, and we deduce that $\text{id}_\Omega + f \in \text{Diff}(\Omega)$. \square

We have the following (strict) inclusions concerning the sets we are working with

$$\begin{array}{ccc} \mathcal{U}_1 & \longrightarrow & \mathcal{V}_1 \\ \downarrow & & \downarrow \\ \kappa(\text{Diff}_\partial(\Omega)) & \longrightarrow & \kappa(\text{Diff}(\Omega)_0) \longrightarrow C^\infty(\Omega, \mathbb{R}^d). \end{array}$$

In practice, we work with vector fields to approximate diffeomorphisms. As the algebra of vector fields is infinite dimensional, we have to replace it by a finite-dimensional subspace. To increase the supply of diffeomorphisms we can approximate, our method incorporates *compositions* of diffeomorphisms. Since $\kappa^{-1}(\mathcal{U}_1)$ and $\kappa^{-1}(\mathcal{V}_1)$ are identity neighborhoods in the respective diffeomorphism groups, they generate the group,² i.e. every diffeomorphism can be written as a finite (but maybe arbitrarily long) composition of elements near the identity. Passing to the intersection with a finite-dimensional subspace, the generating property is lost, but compositions will nevertheless yield a much broader spectrum of diffeomorphisms which can be reached.

Remark 3.5. These observations have been exploited in the *boundaryless* case for example in [1]. Since $\text{Diff}(\Omega)_0$ is simple in this case, there are several characterizations available for a subset of vector fields to generate a sufficiently rich subgroup (see e.g. [1, 2] for a treatment of flows in the framework of sub-Riemannian geometry on manifolds *without* boundary). Note that it is not clear how these techniques (e.g. sub-Riemannian geometry) can be generalized to manifolds with boundary.

Since our method relies on compositions of multiple diffeomorphisms, the next section provides insight and estimates into such compositions.

4 Estimates concerning multiple compositions

If we consider compositions $\phi := \phi_L \circ \dots \circ \phi_1$ of diffeomorphisms $\phi_1, \dots, \phi_L \in \text{Diff}_\partial(\Omega)$ for a large number L of factors, we'd like to be able to control the

² We are referring to the connected component of the identity of the respective diffeomorphism group here.

distance $\|\phi - \text{id}_\Omega\|_{C^k}$ of the composite from the identity id_Ω in terms of the distances $\|\phi_1 - \text{id}_\Omega\|_{C^k}, \dots, \|\phi_L - \text{id}_\Omega\|_{C^k}$ (where the norms $\|\cdot\|_{C^k}$ are as in (4.9)). We show that such an estimate is possible using a closed formula which applies to all L . Let Ω be $[0, 1]^d$, or a compact convex subset of \mathbb{R}^d with dense interior. For certain positive integers $M_0 \leq M_1 \leq M_2 \leq \dots$ which can be calculated recursively, we have:

Proposition 4.1. *For all $k \in \mathbb{N}_0$ and all $L \in \mathbb{N}$,*

$$\begin{aligned} & \|((\text{id}_\Omega + f_L) \circ \dots \circ (\text{id}_\Omega + f_1)) - \text{id}_\Omega\|_{C^k} \\ & \leq M_k e^{k \sum_{j=1}^L \|f_j\|_{C^k}} (\|f_1\|_{C^k} + \dots + \|f_L\|_{C^k}) \end{aligned} \quad (4.1)$$

holds for all $f_1, \dots, f_L \in \text{Diff}_\partial(\Omega) - \text{id}_\Omega$ such that $\|f_1\|_{C^k} + \dots + \|f_L\|_{C^k} \leq 1$. Notably,

$$\|((\text{id}_\Omega + f_L) \circ \dots \circ (\text{id}_\Omega + f_1)) - \text{id}_\Omega\|_{C^k} \leq M_k e^k (\|f_1\|_{C^k} + \dots + \|f_L\|_{C^k}). \quad (4.2)$$

Here $M_0 := M_1 := 1$. For $k \in \mathbb{N}$ and $j \in \{1, \dots, k\}$, let $P_{k,j}$ be the set of all partitions $P = \{I_1, \dots, I_j\}$ of $\{1, \dots, k\}$ into disjoint, non-empty subsets I_1, \dots, I_j of $\{1, \dots, k\}$. Then

$$M_k := \sum_{j=2}^k \sum_{P \in P_{k,j}} M_{|I_1|} \dots M_{|I_j|} \quad \text{for } k \geq 2. \quad (4.3)$$

If $n_1 \geq n_2 \geq \dots \geq n_j \geq 1$ are integers with $n_1 + \dots + n_j = k$, write $P(n_1, \dots, n_j) \subseteq P_j$ for the set of all partitions of $\{1, \dots, k\}$ into subsets I_1, \dots, I_j of cardinality $|I_a| = n_a$ for $a \in \{1, \dots, j\}$. Then

$$M_k = \sum_{j=2}^k \sum_{n_1 + \dots + n_j = k} |P(n_1, \dots, n_j)| M_{n_1} \dots M_{n_j},$$

using a summation over all $(n_1, \dots, n_j) \in \mathbb{N}^j$ with $n_1 \geq n_2 \geq \dots \geq n_j$ and $n_1 + \dots + n_j = k$.

For small k , the numbers M_k can easily be calculated by hand; their values are as follows:

k	1	2	3	4	5	6	7	8	9	10
M_k	1	1	4	26	236	2752	39208	660032	12818912	282137824

The constants may not be optimal. Note that, for every $k \geq 3$, the set $\{1, \dots, k\}$ admits k partitions into a singleton and a set with $k - 1$ elements. Thus $M_k \geq kM_1M_{k-1} \geq M_{k-1}$.

Remark 4.2. The numbers M_k for $k \in \mathbb{N}$ are well known in combinatorics; they arise in a classical enumeration problem known as Schröder's Fourth Problem (cf. [32,34]). A longer list can be found in the on-line encyclopedia of integer sequences (OEIS), sequence A000311.

Remark 4.3. For $m \in \mathbb{N}$, the assertions of Proposition 4.1 remain valid for all $k \in \{0, 1, \dots, m\}$ if $f_1, \dots, f_L \in \text{Diff}_\partial^{C^m}(\Omega) - \text{id}_\Omega \subseteq C_\partial^m(\Omega, \mathbb{R}^d)$ (the proof carries over).

In the following, $\mathcal{V} := \kappa(\text{Diff}_\partial(\Omega)) = \text{Diff}_\partial(\Omega) - \text{id}_\Omega$, which is an open 0-neighborhood in $C_\partial^\infty(\Omega, \mathbb{R}^d)$.

Remark 4.4. The proof will show that the condition $\|f_1\|_{C^k} + \dots + \|f_L\|_{C^k} \leq 1$ in Proposition 4.1 is unnecessary for the validity of (4.1) in the cases $k = 0$ and $k = 1$. Actually,

$$\|((\text{id}_\Omega + f_L) \circ \dots \circ (\text{id}_\Omega + f_1)) - \text{id}_\Omega\|_\infty \leq \|f_1\|_\infty + \dots + \|f_L\|_\infty \quad (4.4)$$

for all $L \in \mathbb{N}$ and $f_1, \dots, f_L \in \mathcal{V}$. We shall also see that

$$\text{Lip}(((\text{id}_\Omega + f_L) \circ \dots \circ (\text{id}_\Omega + f_1)) - \text{id}_\Omega) \quad (4.5)$$

$$\leq (1 + \text{Lip}(f_1)) \cdots (1 + \text{Lip}(f_L)) - 1 \quad (4.6)$$

$$\leq e^{\text{Lip}(f_1) + \dots + \text{Lip}(f_L)} - 1 \quad (4.7)$$

$$\leq e^{\text{Lip}(f_1) + \dots + \text{Lip}(f_L)} (\text{Lip}(f_1) + \dots + \text{Lip}(f_L)) \quad (4.8)$$

for all $L \in \mathbb{N}$ and $f_1, \dots, f_L \in \mathcal{V}$.

Remark 4.5. In the special case $\Omega = [0, 1]^d$, the conclusion of Proposition 4.1 remains valid if $\text{Diff}_\partial(\Omega)$ is replaced with $\text{Diff}(\Omega)_0$ and vector fields vanishing on the boundary are replaced with vector fields which are tangent to the boundary. The same comment applies to Remarks 4.3 and 4.4.

The following notations and facts are useful for the proof of Proposition 4.1.

4.1. If $(E, \|\cdot\|_E)$ is a normed space, we write $\overline{B}_r^E(x) := \{y \in E : \|y - x\|_E \leq r\}$ for the closed ball of radius $r > 0$ around $x \in E$. If also $(F, \|\cdot\|_F)$ is a normed space, $k \in \mathbb{N}$ and $\beta: E^k \rightarrow F$ a continuous k -linear map, we define

$$\|\beta\|_{\text{op}} := \sup\{\|\beta(x_1, \dots, x_k)\|_F : x_1, \dots, x_k \in \overline{B}_1^E(0)\}.$$

Then $\|\beta(x_1, \dots, x_k)\|_F \leq \|\beta\|_{\text{op}} \|x_1\|_E \cdots \|x_k\|_E$ for all $x_1, \dots, x_k \in E$. If Ω is a compact topological space and $g: \Omega \rightarrow (\mathcal{L}^k(E, F), \|\cdot\|_{\text{op}})$ a continuous map to the space of continuous k -linear maps, we let

$$\|g\|_{\infty, \text{op}} := \sup_{x \in \Omega} \|g(x)\|_{\text{op}} \in [0, \infty[.$$

4.2. Let $E := \mathbb{R}^d$, endowed with a fixed norm $\|\cdot\|$, and $\Omega \subseteq E$ be a compact convex subset with non-empty interior. For $f \in C^\infty(\Omega, E)$ and $k \in \mathbb{N}_0$, let $d^k f: \Omega \times E^k \rightarrow E$ be the continuous map such that, for all $x \in \Omega^\circ$ and $y_1, \dots, y_k \in E$,

$$d^k f(x, y_1, \dots, y_k) := (D_{y_k} \cdots D_{y_1} f)(x)$$

is the iterated directional derivative at x in the directions y_1, \dots, y_k . Recall that the topology on $C_\partial^\infty(\Omega, E)$ is initial with respect to the mappings

$C_{\partial}^{\infty}(\Omega, E) \rightarrow C(\Omega \times E^k, E)$, $f \mapsto d^k f$ for $k \in \mathbb{N}_0$, using the topology of compact convergence on spaces of continuous functions. Thus, the seminorms

$$f \mapsto \sup\{\|d^k f(z)\|_E : z \in K\}$$

define the locally convex vector topology on $C_{\partial}^{\infty}(\Omega, E)$, for $k \in \mathbb{N}_0$ and compact subsets $K \subseteq \Omega \times E^k$. Abbreviate $f^{(k)}(x) := d^k f(x, \cdot) \in \mathcal{L}^k(E, E)$ for $k \in \mathbb{N}$ and $x \in \Omega$. Then

$$f \mapsto \|f^{(k)}\|_{\infty, \text{op}} = \sup\{\|d^k f(z)\|_E : z \in \Omega \times \overline{B}_1^E(0)^k\}$$

is a continuous seminorm on $C_{\partial}^{\infty}(\Omega, E)$. Together with the supremum norm $\|\cdot\|_{\infty}$, the latter seminorms define the locally convex topological vector space topology of $C_{\partial}^{\infty}(\Omega, E)$. In fact, for each $k \in \mathbb{N}$ and compact subset $K \subseteq \Omega \times E^k$, we have $K \subseteq \Omega \times \overline{B}_r^E(0)^k$ for some $r > 0$, whence

$$\sup\{\|d^k f(x, z)\|_E : z \in K\} \leq r^k \|f^{(k)}\|_{\infty, \text{op}}.$$

Hence, the vector topology on $C_{\partial}^{\infty}(\Omega, E)$ is also defined by the sequence of norms $(\|\cdot\|_{C^k})_{k \in \mathbb{N}_0}$ with $\|\cdot\|_{C^0} := \|\cdot\|_{\infty}$ and

$$\|f\|_{C^k} := \max\{\|f\|_{\infty}, \|f^{(1)}\|_{\infty, \text{op}}, \dots, \|f^{(k)}\|_{\infty, \text{op}}\}. \quad (4.9)$$

The following formula actually holds for locally convex spaces X, Y , and Z .

4.3. (Faà di Bruno's Formula) Let X, Y , and Z be finite-dimensional real vector spaces, $U \subseteq X$ and $V \subseteq Y$ be locally convex subsets with dense interior, and $k \in \mathbb{N}$. Let $g: U \rightarrow Y$ and $f: V \rightarrow Z$ be C^k -maps such that $g(U) \subseteq V$. Then

$$d^k(f \circ g)(x, y_1, \dots, y_k) = \sum_{j=1}^k \sum_{P \in P_{k,j}} d^j f(d^{|I_1|} g(x, y_{I_1}), \dots, d^{|I_j|} g(x, y_{I_j}))$$

for all $x \in U$ and $y_1, \dots, y_k \in E$, where $P = \{I_1, \dots, I_j\}$ and

$$y_J := (y_{i_1}, \dots, y_{i_\ell})$$

for each subset $J \subseteq \{1, \dots, k\}$ of the form $J = \{i_1, \dots, i_\ell\}$ with $i_1 < \dots < i_\ell$ (see [15, Theorem 1.3.18 and Remark 1.4.15 (c)]; cf. also [10]).

4.4. We shall use the following fact: If $L \in \mathbb{N}$ and $x_1, \dots, x_L \geq 0$, then

$$(1 + x_1)(1 + x_2) \cdots (1 + x_L) \leq e^{x_1 + \dots + x_L} \quad (4.10)$$

as $\ln((1 + x_1) \cdots (1 + x_L)) = \sum_{j=1}^L \ln(1 + x_j) \leq \sum_{j=1}^L x_j$. We shall also use that

$$e^x - 1 \leq e^x x \text{ for all } x \geq 0. \quad (4.11)$$

In fact, $e^x - 1 = \exp(x) - \exp(0) = \exp'(\xi)x$ for some $\xi \in [0, x]$ by the Mean Value Theorem, with $\exp'(\xi) = \exp(\xi) \leq \exp(x)$ by monotonicity of \exp .

4.5. Let $(E, \|\cdot\|)$ be a normed space and $(\mathcal{L}(E), \|\cdot\|_{\text{op}})$ be the corresponding algebra of bounded operators. For all $L \in \mathbb{N}$ and $\alpha_1, \dots, \alpha_L \in \mathcal{L}(E)$, we trivially have

$$\|(\text{id}_E + \alpha_L) \circ \dots \circ (\text{id}_E + \alpha_1)\|_{\text{op}} \leq (1 + \|\alpha_L\|_{\text{op}}) \cdots (1 + \|\alpha_1\|_{\text{op}}). \quad (4.12)$$

Moreover,

$$\begin{aligned} \|((\text{id}_E + \alpha_L) \circ \dots \circ (\text{id}_E + \alpha_1)) - \text{id}_E\|_{\text{op}} \\ \leq ((1 + \|\alpha_L\|_{\text{op}}) \cdots (1 + \|\alpha_1\|_{\text{op}})) - 1 \end{aligned} \quad (4.13)$$

$$\leq e^{\|\alpha_1\|_{\text{op}} + \dots + \|\alpha_L\|_{\text{op}}} - 1 \quad (4.14)$$

$$\leq e^{\|\alpha_1\|_{\text{op}} + \dots + \|\alpha_L\|_{\text{op}}} (\|\alpha_1\|_{\text{op}} + \dots + \|\alpha_L\|_{\text{op}}). \quad (4.15)$$

In fact,

$$\beta := (\text{id}_E + \alpha_1) \cdots (\text{id}_E + \alpha_L) - \text{id}_E = \sum_{j=1}^m \sum_{|I|=j} \alpha_{i_1} \circ \dots \circ \alpha_{i_j},$$

using subsets $I = \{i_1, \dots, i_j\} \subseteq \{1, \dots, L\}$ with $i_1 < \dots < i_j$. Thus

$$\|\beta\|_{\text{op}} \leq \sum_{j=1}^L \sum_{|I|=j} \|\alpha_{i_1}\|_{\text{op}} \cdots \|\alpha_{i_j}\|_{\text{op}} = ((1 + \|\alpha_1\|_{\text{op}}) \cdots (1 + \|\alpha_L\|_{\text{op}})) - 1,$$

establishing (4.13). Using (4.10) and (4.11), the inequalities (4.14) and (4.15) follow.

Proof of Proposition 4.1. We first prove the assertion for $k \in \{0, 1\}$.

The case $k = 0$. For $f, g \in \mathcal{V}$, let

$$f * g := ((\text{id}_\Omega + f) \circ (\text{id}_\Omega + g)) - \text{id}_\Omega = g + f \circ (\text{id}_\Omega + g).$$

Then $f * g \in \mathcal{V}$ and

$$\|f * g\|_\infty \leq \|g\|_\infty + \|f\|_\infty.$$

Given $f_1, \dots, f_L \in \mathcal{V}$, we define $f_L * \dots * f_1$ recursively as $f_L * (f_{L-1} * \dots * f_1)$ and obtain

$$\|f_L * \dots * f_1\|_\infty \leq \|f_1\|_\infty + \dots + \|f_L\|_\infty, \quad (4.16)$$

by a straightforward induction. By induction on $2 \leq L \in \mathbb{N}$, also

$$((\text{id}_\Omega + f_L) \circ \dots \circ (\text{id}_\Omega + f_1)) - \text{id}_\Omega = f_L * \dots * f_1,$$

as the left-hand side of the equation equals

$$\begin{aligned} (\text{id}_\Omega + f_L)(\text{id}_\Omega + ((\text{id}_\Omega + f_{L-1}) \circ \dots \circ (\text{id}_\Omega + f_1)) - \text{id}_\Omega) - \text{id}_\Omega \\ = f_L * ((\text{id}_\Omega + f_{L-1}) \circ \dots \circ (\text{id}_\Omega + f_1)) - \text{id}_\Omega \\ = f_L * (f_{L-1} * \dots * f_1). \end{aligned}$$

Hence (4.4) is a re-writing of (4.16). Notably, (4.1) holds for $k = 0$ with $M_0 := 1$.

The case $k = 1$. Abbreviate $E := \mathbb{R}^d$. For $f_1, \dots, f_L \in \mathcal{V}$, $x_0 \in \Omega$ and $x_j := f_j(x_{j-1})$ for $j \in \{1, \dots, L-1\}$, we have

$$\begin{aligned} & (((\text{id}_\Omega + f_L) \circ \dots \circ (\text{id}_\Omega + f_1)) - \text{id}_\Omega)'(x_0) \\ &= ((\text{id}_E + f'_1(x_0)) \circ \dots \circ (\text{id}_E + f'_L(x_{L-1}))) - \text{id}_E \end{aligned}$$

with operator norm $\leq ((1 + \|f'_1(x_0)\|_{\text{op}}) \cdots (1 + \|f'_L(x_{L-1})\|_{\text{op}})) - 1$, by (4.13). Passing to the supremum in $x_0 \in \Omega$, we get

$$\|(f_L * \dots * f_1)'\|_{\infty, \text{op}} \leq ((1 + \|f'_1\|_{\infty, \text{op}}) \cdots (1 + \|f'_L\|_{\infty, \text{op}})) - 1,$$

which is bounded by $((1 + \|f_1\|_{C^1}) \cdots (1 + \|f_L\|_{C^1})) - 1$. Thus (4.6) holds, and (4.7) as well as (4.8) follow using (4.10) and (4.11). As the latter number is also an upper bound for $\|f_L * \dots * f_1\|_\infty$, we deduce that

$$\|f_L * \dots * f_1\|_{C^1} \leq ((1 + \|f_1\|_{C^1}) \cdots (1 + \|f_L\|_{C^1})) - 1$$

for all $L \in \mathbb{N}$ and $f_1, \dots, f_L \in \Omega$. By (4.10) and (4.11), the right-hand side is bounded by

$$e^{\|f_1\|_{C^1} + \dots + \|f_L\|_{C^1}} - 1 \leq e^{\|f_1\|_{C^1} + \dots + \|f_L\|_{C^1}} (\|f_1\|_{C^1} + \dots + \|f_L\|_{C^1}),$$

for all $L \in \mathbb{N}$ and $f_1, \dots, f_L \in \mathcal{V}$. Notably, (4.1) holds for $k = 1$ with $M_1 := 1$.

For $k \geq 2$, the proof is by induction. Details are postponed to Appendix A. \square

The upper bounds of proposition 4.1 are theoretical. It could therefore be of interest to see how they relate to the practically achieved C^k -norms of diffeomorphisms of the form used in section 6 and how these bounds are affected by for example the number of compositions. For simplicity, we stick to the one-dimensional case. In figure 4.1, the value

$$\frac{\|((\text{id}_\Omega + f_L) \circ \dots \circ (\text{id}_\Omega + f_1)) - \text{id}_\Omega\|_{C^k}}{\left(\sum_{\ell}^L \|f_\ell\|_{C^k}\right) e^{k \sum_{\ell}^L \|f_\ell\|_{C^k}}} \quad (4.17)$$

is shown for lower-order derivatives with varying number of layers L and number of basis functions d per layer. For $k = 1, 2$ the estimates are relatively good with values close to 60% and 15% of M_k respectively.

These estimates are largely dependent on the choice of basis functions, so the results may look different for another set of basis functions than the ones used here. Further details regarding how the values of figure 4.1 are computed can be found in appendix B.

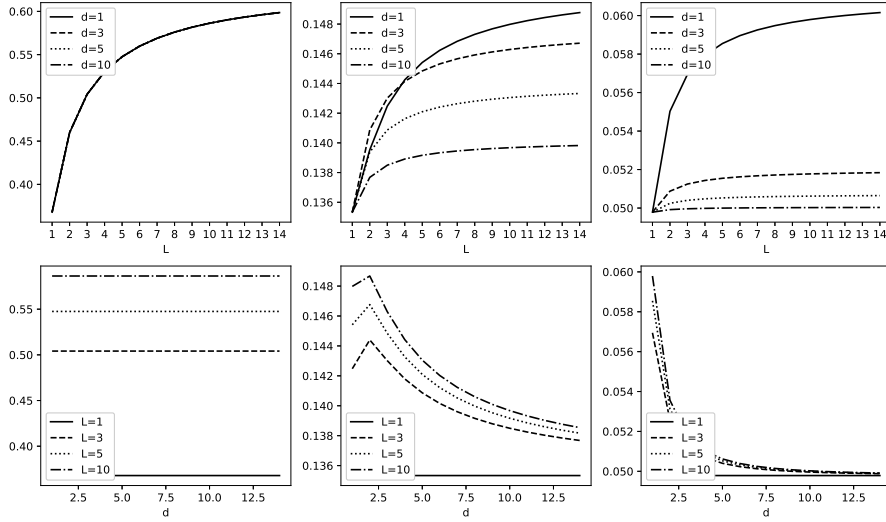


Fig. 4.1 Practical estimates of the C^k bounding coefficient in equation (4.1) for $k = 1, 2, 3$ (left to right). (Top): Varying number of layers L . (Bottom): Varying number of basis functions per layer M .

5 Implementation of the methods

This section describes the implementation of the approach outlined in section 2.4. We first consider the case of curves and then the case of surfaces. In both cases, we need three parts: a discrete problem formulation, a choice of basis functions, and a projection operator to ensure that the Lipschitz constant of each layer is bounded by one.

5.1 Curves

Problem Formulation

Denote by $W = \{\mathbf{w}^\ell\}_{\ell=1}^L \subset \mathbb{R}^d$ a set of weight vectors \mathbf{w}^ℓ , and use the notation

$$\varphi(x; W) = \varphi_L(x; \mathbf{w}^L) \circ \dots \circ \varphi_1(x; \mathbf{w}^1) \quad (5.1)$$

to highlight the dependency of φ on W , and φ_ℓ on \mathbf{w}^ℓ , respectively. Each of the layers are defined as residual-blocks

$$\varphi_\ell(x, \mathbf{w}^\ell) = x + \sum_{n=1}^M w_n^\ell f_n(x)$$

with basis function f_n as defined in (5.3). The goal is to find a set of vectors W such that the function $\varphi(\cdot, W) \in \text{Diff}^+(\Omega)$ minimizes the loss function (2.6). To approximate the shape distance, take a collection $X = \{x_k\}_{k=1}^K \subset \Omega$ of

linearly spaced points, and compute the mean squared error (MSE) between the q -maps evaluated at these points. In other words, the loss function is approximated by the function

$$E(W; X) = \frac{1}{K} \sum_{k=1}^K |q(x_k) - \sqrt{\varphi'(x_k; W)} r(\varphi(x_k; W))|^2. \quad (5.2)$$

This function should be optimized with respect to the weights W under the constraints that $\varphi(0, W) = 0$, $\varphi(1, W) = 1$, and $\varphi'(\cdot : W) > 0$. The optimization is handled by a BFGS-optimizer, with a projection step between the weight updates to ensure that the layers are invertible.³ The projection step is similar to the spectral normalization approach of [4], and entails a scaling of the weight vectors such that the Lipschitz constants of each of the residual maps $x \mapsto \sum_{n=1}^M w_n^\ell f_n(x)$ are bounded by one.

Basis Functions

The tangent space of the diffeomorphism group on an interval consists of smooth functions that vanish on the boundary of the interval. A simple finite-dimensional basis which satisfies this relation is a truncated Fourier sine series. However, the algorithm seems to perform better when the derivatives of the basis functions have similar magnitudes, and we scale the basis functions such that their derivatives are bounded by one. Therefore, we choose the basis of functions

$$f_n(x) = \frac{\sin(n\pi x)}{n\pi}, \quad n = 1, \dots, M \quad (5.3)$$

for $M \in \mathbb{N}$.

Projection Operator

To ensure the invertibility of each layer φ_ℓ , we impose that the Lipschitz constant of each of the residual maps is smaller than one. For this purpose, denote by \mathcal{W}^ε the *feasible set* of the weight vectors. To define this set explicitly, consider the upper estimate of the Lipschitz constants given by the inequalities

$$\left| \left(\sum_{n=1}^M w_n f_n(x) \right) - \left(\sum_{n=1}^M w_n f_n(y) \right) \right| \leq \underbrace{\left(\sum_{n=1}^M |w_n| L_n \right)}_{:=L} |x - y|, \quad (5.4)$$

where L_n is the Lipschitz constant for the basis function f_n . For our choice of basis functions, we have that $L_n = \sup_{x \in \Omega} |f_n'(x)| = 1$ for all $n = 1, \dots, M$.

³ This approach has performed well for the experiments in this article, but we have found limited theory on the convergence of BFGS with a projection step. A large scale application of the reparametrization algorithm might gain from changing to a constrained optimization routine, or to another projected quasi-Newton algorithm with a better theoretical foundation.

Thus $L = \sum_{n=1}^M |w_n| L_n = \|\mathbf{w}\|_1$. To ensure that the layers are diffeomorphisms, we require that L is *strictly* smaller than one. To achieve this, we define the feasible set

$$\mathcal{W}^\varepsilon = \{\mathbf{w} \in \mathbb{R}^M : \|\mathbf{w}\|_1 \leq 1 - \varepsilon\} \quad (5.5)$$

for some small ε . If the optimization algorithm makes a weight update which causes a violation of these constraints, then it needs to be projected onto the feasible set. We define this projection using a simple scaling of the vector, defined by

$$\pi: \mathbb{R}^M \rightarrow \mathcal{W}^\varepsilon, \quad \pi(\mathbf{w}) = \frac{1 - \varepsilon}{\max\{1 - \varepsilon, \|\mathbf{w}\|_1\}} \mathbf{w}. \quad (5.6)$$

5.2 Surfaces

Problem Formulation

The problem formulation for reparametrization of surfaces is analogous to the case for curves, with the natural extensions to two dimensions. We approximate the continuous loss function by

$$E(W; X) = \frac{1}{K^2} \sum_{i,j=1}^K |q(\mathbf{x}_{i,j}) - \sqrt{J_\varphi(\mathbf{x}_{i,j}; W) r(\varphi(\mathbf{x}_{i,j}; W))}|^2, \quad (5.7)$$

where $X = \{\mathbf{x}_{i,j}, i, j = 1, \dots, K\} \subset \Omega (\subset \mathbb{R}^2)$, and φ as in (5.1). The loss function should be optimized with respect to the weights W under the constraints $J_\varphi(\cdot; W) > 0$ and $\varphi(\partial M; W) = \partial M$.

Basis Functions

The tangent space of the diffeomorphism group on $\Omega = [0, 1]^2$ consists of smooth vector fields tangential to the boundary $\partial\Omega$. We construct a basis for this space componentwise, as a tensor product basis of Fourier sine series in one direction and a full Fourier basis in the other. This choice was inspired by the approach adopted in [24]. In the first component, these functions are given by the three families of functions,

$$\xi_k(x, y) = \frac{\sin(\pi k x)}{\pi k}, \quad (5.8a)$$

$$\eta_{k,l}(x, y) = \frac{\sin(\pi k x) \cos(2\pi l y)}{\pi k l}, \quad (5.8b)$$

$$\phi_{k,l}(x, y) = \frac{\sin(\pi k x) \sin(2\pi l y)}{\pi k l}, \quad (5.8c)$$

with corresponding basis functions $\tilde{\xi}_k(x, y) = \xi_k(y, x)$, $\tilde{\eta}_{k,l}(x, y) = \eta_{k,l}(y, x)$ and $\tilde{\phi}_{k,l}(x, y) = \phi_{k,l}(y, x)$ for the second component. A few example basis

vector fields are shown in figure 5.1. The basis is truncated such that the coefficients $k, l = 1, \dots, N$ for some fixed value of $N \in \mathbb{N}$. The total number of basis functions is therefore $M = 2(2N^2 + N)$. In other words, the number of basis functions per layer in the network increases quadratically with N .

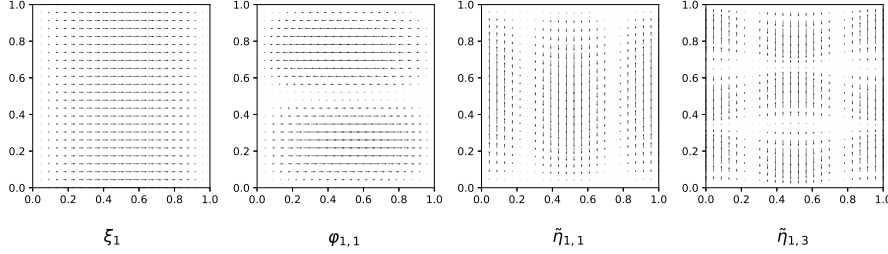


Fig. 5.1 Four examples of basis elements of $T_{\text{id}}(\text{Diff}^+(\Omega))$, as described in 5.2. The basis fields are constructed componentwise as products of trigonometric functions and point either purely in the x - or y -direction.

Projection Operator

Similarly as in section 5.1, the invertibility of each layer φ_ℓ is ensured by bounding the Lipschitz constants of the residual maps by one. The projection operator is analogous to the one-dimensional case, and is based on the approximation (5.4). However, the Lipschitz constant L_n of each of the basis vector fields is no longer simply equal to 1. Instead, we make use of the following approximation:

$$L_n = \sup_{x \in \Omega} \|Df_n(x)\|_2 \leq \sup_{x \in \Omega} \|Df_n(x)\|_F, \quad n = 1, \dots, M,$$

where Df_n denotes the Jacobian matrix of f_n , the symbol $\|\cdot\|_2$ denotes the spectral norm, and $\|\cdot\|_F$ the Frobenius norm. According to this estimate, the Lipschitz constants for each of the three types of basis vector fields are bounded by

$$L_k^\xi = 1, \quad L_{k,l}^\eta = \frac{\sqrt{k^2 + 2l^2}}{kl}, \quad L_{k,l}^\phi = \frac{\sqrt{k^2 + 2l^2}}{kl}, \quad (5.9)$$

and are found by taking the Frobenius norm of the Jacobian matrices of the basis functions in (5.8). Similar constants exist for the corresponding vector fields in the y -direction. Adopting the notations from the one-dimensional case, we define

$$\pi: \mathbb{R}^M \rightarrow \mathcal{W}^\varepsilon, \quad \pi(\mathbf{w}) = \frac{1 - \varepsilon}{\max\left\{1 - \varepsilon, \left(\sum_{n=1}^M |w_n| L_n\right)\right\}} \mathbf{w}, \quad (5.10)$$

which maps infeasible weight vectors to the feasible set.

5.3 Shape Interpolation

An interesting application of the reparametrization algorithm is to create realistic transitions between different shapes. The shape distance d_S corresponds to the length of some geodesic between the two parametric shape representations in the pre-shape space \mathcal{P} . Each of the points along this geodesic corresponds to a parametric representation of some intermediate shape, providing a natural transformation between the original curves or surfaces.

In the specific case of reparametrization of curves using the SRVT, we may compute a geodesic between their shapes. Assume for any $\tau \in (0, 1)$ that the curve $\tau\mathcal{R}(c_1)(x) + (1 - \tau)\mathcal{R}(c_2)(x) \neq 0, \forall x \in \Omega$. Then the map

$$\tau \mapsto \mathcal{R}^{-1}(\tau\mathcal{R}(c_1) + (1 - \tau)\mathcal{R}(c_2)) \quad (5.11)$$

with

$$\mathcal{R}^{-1}(q)(x) = \int_0^x q(\xi) |q(\xi)| d\xi \quad (5.12)$$

defines a geodesic between the two curves c_1 and c_2 .

The surface transforms, however, are not invertible. Therefore, we adopt an alternative approach which seems to work well in many cases. Define the map

$$\gamma : \mathcal{P} \times \mathcal{P} \times [0, 1] \rightarrow C^\infty(\Omega, \mathbb{R}^n), \quad \gamma(f_1, f_2, \tau) = \tau f_1 + (1 - \tau) f_2.$$

Note that for $f_1, f_2 \in \mathcal{P}$, the map $\tau \mapsto \gamma(f_1, f_2, \tau)$ defines a convex linear combination of two elements in $C^\infty(\Omega, \mathbb{R}^n)$ connecting the two surfaces. However, when applied directly to the parametrized curves without performing the reparametrization, the corresponding path of this interpolation in shape-space is typically far from length-minimizing. To improve upon this shape transformation, we start by reparametrizing one of the shapes by finding $\varphi^* \in \arg \min_{\varphi \in \text{Diff}(\Omega)} d_{\mathcal{P}}([f_1], [f_2 \circ \varphi])$ and then apply the shape interpolation according to

$$\tau \mapsto \gamma(f_1, f_2 \circ \varphi^*, \tau). \quad (5.13)$$

Figure 5.2 illustrates how the different interpolation approaches compare to each other both in pre-shape space and in shape space. Figure 5.3 shows the different approaches applied to curves and figure 6.4 presents an example of the approach (5.13) applied to images. Applied directly to the parametrized curves as in the left panel of fig. 5.3, the intermediate curves of the transition between the two shapes does not follow what we intuitively consider a shortest path between the curves. Instead of only changing the parts of the curves which differs between the two shapes, the entire curve is moved in the intermediate steps. This is in contrast to both the middle- and the right panel of fig. 5.3 where the upper half-circle is stationary, and only the lower half of the circle is moving between the shapes.

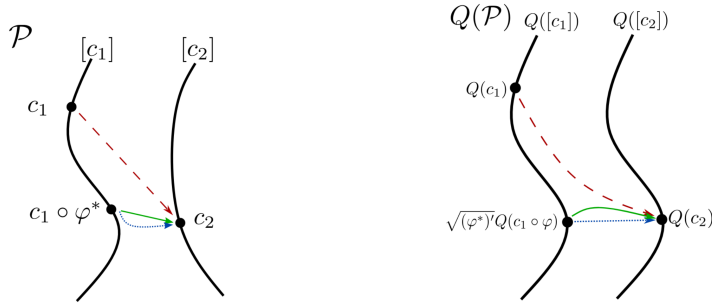


Fig. 5.2 Different interpolation methods between two shapes and how these curves relate to different shape representations. The dashed red curve corresponds to the direct linear interpolation between the two original parametric shapes. The solid green curve corresponds to a geodesic, whereas the dotted blue line represents a linear interpolation after reparametrization given in (5.13).

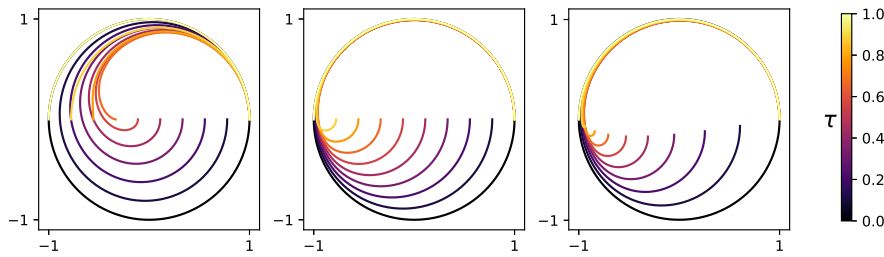


Fig. 5.3 Comparison of the methods described in section 5.3 for interpolation between a half-circle and a circle. (Left) Direct linear interpolation, corresponding to the dashed red curve in 5.3. (Middle) Linear interpolation after reparametrization, corresponding to the dotted blue curve. (Right) Geodesic, corresponding to the solid green curve.

5.4 Notes on Implementation

Since the structure of the diffeomorphisms that we are searching for is similar to that of a residual neural network, the weight optimization is analogous to the process of training a neural network. This is typically done by an iterative line search method such as gradient descent rather than Newton’s method or other Quasi-Newton algorithms. However, since we are dealing with a deterministic problem, we have chosen to use the BFGS-algorithm (see e.g. [29]). The network is implemented using the PyTorch machine learning framework in Python[31] which takes care of the computation of gradients and updates the weight vectors. By building our algorithms upon the PyTorch-framework, the algorithms easily achieve high performance, scalability and extensibility.

6 Numerical Experiments

6.1 Curves

We start by comparing two different parametric curves representing the same shape. A simple way to find two such curves is by first selecting a curve c , and reparametrize the curve with some diffeomorphism φ . By applying the reparametrization algorithm to the curves $c_1 = c \circ \varphi$ and $c_2 = c$, we expect to recover a diffeomorphism $\psi \approx \varphi$ such that the distance $d_{\mathcal{P}}(c_1, c_2 \circ \psi) \approx 0$. As a first example, consider the parametric curve and diffeomorphism given by

$$c(t) = [\cos(2\pi t), \sin(4\pi t)],$$

$$\varphi(t) = \frac{\log(20t + 1)}{2 \log(21)} + \frac{1 + \tanh(20(t - 0.5))}{4 \tanh(10)}. \quad (6.1)$$

The curves c and $c \circ \varphi$ are shown in figure 6.1. The dots along the curves represent linearly spaced points on the domain $\Omega = [0, 1]$ for the curve parameter t . Figure 6.1 presents the results of applying the reparametrization algorithm to these curves using a network with $L = 10$ layers and $M = 10$ basis functions per layer. The resulting reparametrization ψ is visually indistinguishable from the true diffeomorphism φ , and the components of the reparametrized curve $c \circ \psi$ exactly match the components of the components of $c \circ \varphi$.

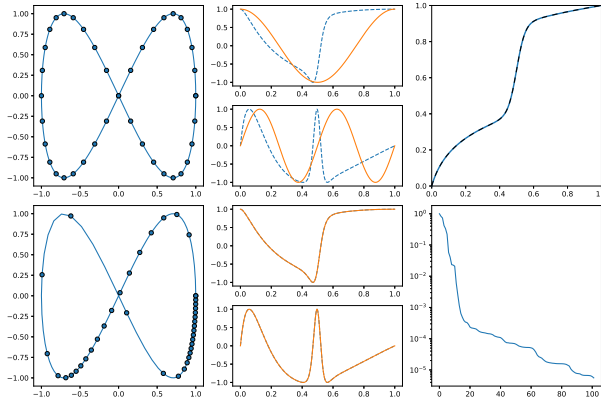


Fig. 6.1 (Left) The curve c to be reparametrized on top, and the target curve $c \circ \varphi$ below, defined in (6.1). (Middle) The component functions of the two curves $c \circ \varphi$ (dotted blue) and c (solid orange) before, and after reparametrization. (Top right) The true reparametrization φ (dotted black) compared to the reparametrization ψ found by the algorithm. (Bottom right): The value of the loss function plotted against the iteration number of the weight updates, given relative to the initial error. Each iteration corresponds to one iteration of the BFGS algorithm including line-search.

6.2 Surfaces

Similarly as in section 6.1, the performance of the reparametrization algorithm is investigated by comparing surfaces representing the same shape. The results are presented in figures 6.2, 6.3. The coloring on the surfaces represents the local area scaling factor a_f for the given surface. Once again, the algorithm does indeed find a diffeomorphism $\psi \approx \varphi$. Moreover, based on figure 6.6, the convergence behavior of the reparametrization algorithm under an increasing number of basis functions and network layers is similar to the behavior already observed for curves.

The previous examples are using artificial surfaces defined by explicit parametric functions. In most practical applications, however, objects are represented by discrete observations. For example, images are typically stored as matrices representing image intensities. Such data must be transformed to parametrized surfaces before reparametrization. For this purpose, we made use of a function⁴ from the PyTorch library which implements a bicubic convolution algorithm, [21]. Since we are working with three-dimensional surfaces, it seems natural to represent single-channel images by their graphs.

The fact that the optimal reparametrization is unknown when comparing surfaces representing different shapes, implies that it is difficult to properly verify the correctness of the algorithm. However, the shape interpolation method of section 5.3 provides a good alternative to validate the results; an optimal reparametrization of one surface relative to the other, should provide a smoother interpolation between the two. Figure 6.4 illustrates the interpolation between two images before and after reparametrization. Before reparametrization, the intermediate images clearly show elements from each of the images simultaneously, as one fades away and the other is revealed. After reparametrization the interpolation represents a smoother transition, with one of the surfaces being “warped” into the other.

⁴ torch.nn.functional.grid_sample

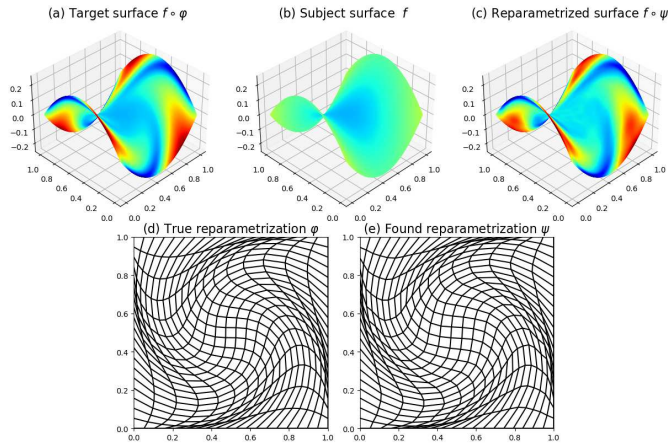


Fig. 6.2 First example of reparametrization of a surface when compared to another parametric surface representing the same shape. The colors on the surfaces represent the area scaling factor of the parametrization. The target surface $f \circ \varphi$ (a) have been created by reparametrizing the subject surface f (b) by some prescribed diffeomorphism φ (d). After applying the reparametrization algorithm to compare the two surfaces, the resulting reparametrization is shown in (e) and the reparametrized surface shown in (c).

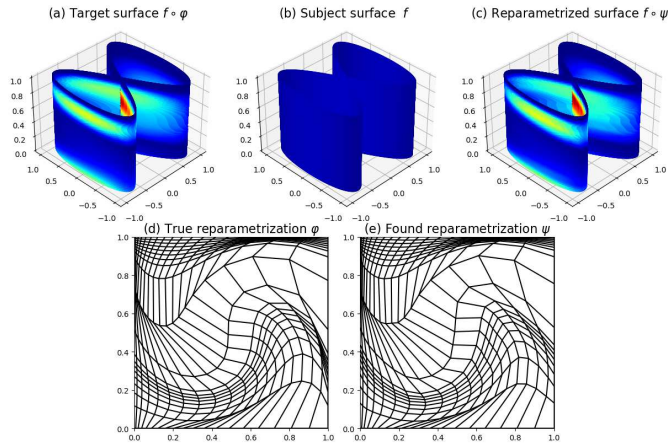


Fig. 6.3 Second example of reparametrization of two surfaces representing the same shape such as in figure 6.2, this time including an immersion and a more complex diffeomorphism involving both rotations and stretching.

6.3 Comparing Network Sizes

Figure 6.5 presents the results of the reparametrization algorithm for a varying number of layers and basis functions per layer in the network. The results represent the value of the loss function (5.2) after reparametrization for the curves from the example in figure 6.1. It seems that by keeping the number of layers low while increasing the number of basis functions, or vice versa, results

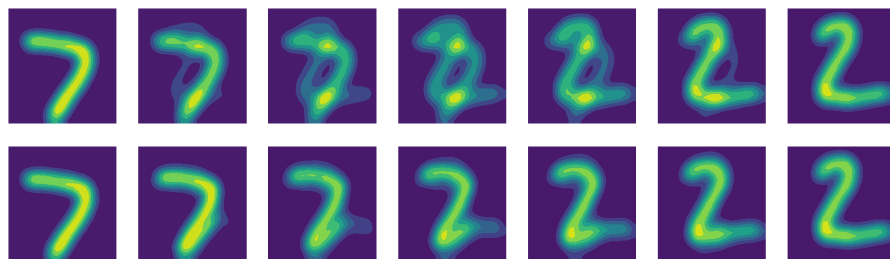


Fig. 6.4 Linear interpolation between two images from the MNIST digits dataset, before and after reparametrization of the surfaces. The first row corresponds to direct linear interpolation between the graphs, whereas the second row corresponds to linear interpolation after reparametrization as given in (5.13).

in a relatively poor match between the functions. A combination of network depth and width is required to reduce the error to the smallest values.

Figure 6.6 presents similar results based on the reparametrization of the surfaces in figure 6.2. Note that N represents the “largest frequency” of the basis functions, and not the number of basis functions. This number is much higher as pointed out in section 5.2. In other words, a deep network will be able to achieve similar performance as a wide network, using much fewer coefficients.

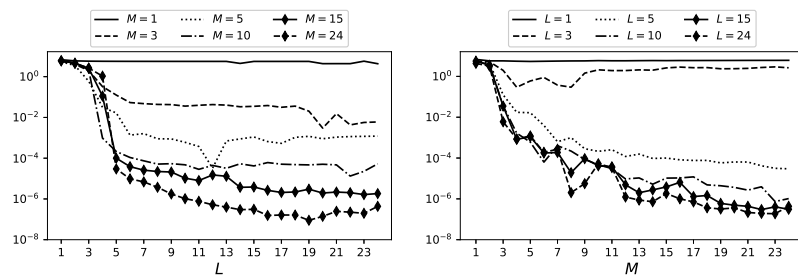


Fig. 6.5 Final error after reparametrization of the curves in figure 6.1, for varying number of layers L and basis functions per layer M .

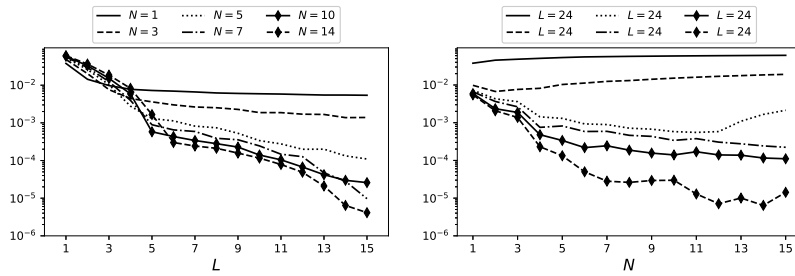


Fig. 6.6 Final error after reparametrization of the surfaces in figure 6.2 for varying number of layers L and the “largest frequency” N of the basis functions. The number of basis functions per layer M is in this case related to the frequency by $M = 2(2N^2 + N)$.

7 Discussion

Based on fig. 6.5 there seems to be a lower limit for how close to the optimal reparametrization one can get using this approach, as opposed to what one would expect based on the universal approximation properties presented in section 3. The main reason behind this behavior seems to be the stopping criteria for the BFGS-optimizer. There are three criteria playing a role in the stopping behaviour: The termination tolerance on the gradient norm, the termination tolerance on the change of function value, or the maximal number of iterations in the optimizer. By significantly lowering the termination tolerances and increasing the maximal number of iterations, it is possible to further reduce the errors for the larger networks. This reduction, however, comes at a high computational cost relative to small improvements to the registration.

Due to the relationship of the algorithm with the gradient-descent algorithm, we have chosen to keep the structure and activation/basis functions from the original problem, rather than doing an extensive investigation into alternative network structures. Because of the constraints of the diffeomorphism group, it does take some effort to switch network structures and activation functions we have considered this to be out of the scope of this article. However, it is most certainly an interesting topic for future research.

One important difference between a typical deep learning problem and the algorithm presented here, is that since we are registering a single pair of shapes, overfitting is desirable in our case — at least under the assumption that the shapes are without noise. This assumption is justified since the noise from the original data will usually be handled by the method one uses to create a parametric curve/surface. This is typically done by interpolation or other regression methods with the possibility to add a regularizer or other preprocessing steps to reduce noise. An example is the bicubic interpolation method used for the MNIST-images in fig. 6.4. For future research, however, it would be interesting if one could incorporate the construction of the parametrization from a point collection into the network structure itself. In such a case,

overfitting would indeed pose a problem, and typical regularization techniques would need to be applied.

8 Conclusion

In this paper we proposed a method for the solution of optimization problems on the group of diffeomorphisms. The method is inspired by deep neural networks and implemented in PyTorch. An analysis of the universal approximation properties and a priori estimates of the norms of the obtained approximations are presented. A number of numerical results illustrate the merit of the proposed approach in applications of shape analysis.

A Proof of Proposition 4.1

In this appendix, we provide the details postponed in the proof of Proposition 4.1. Let us recall the statement for the reader's convenience:

A.1 (Statement of Proposition 4.1). For all $k \in \mathbb{N}_0$ and all $L \in \mathbb{N}$,

$$\begin{aligned} & \|((\text{id}_\Omega + f_L) \circ \cdots \circ (\text{id}_\Omega + f_1)) - \text{id}_\Omega\|_{C^k} \\ & \leq M_k e^k \sum_{j=1}^L \|f_j\|_{C^k} (\|f_1\|_{C^k} + \cdots + \|f_L\|_{C^k}) \end{aligned}$$

holds for all $f_1, \dots, f_L \in \text{Diff}_\partial(\Omega) - \text{id}_\Omega$ such that $\|f_1\|_{C^k} + \cdots + \|f_L\|_{C^k} \leq 1$.

Recall that in Section 4 we have already established the statement for all $L \in \mathbb{N}$ and $k \in \{0, 1\}$. Thus all we need to prove is the statement for all $k \geq 2$.

Proof of Proposition 4.1 in the case $k \geq 2$. The proof is by induction. Assume that $k \geq 2$ and assume that the asserted estimates already hold for $k - 1$ in place of k . For $f, g \in \mathcal{V}$, applying Faà di Bruno's formula to the first term in $f * g = (\text{id}_\Omega + f) \circ (\text{id}_\Omega + g) - \text{id}_\Omega$, we get for all $x \in \Omega$ and $y_1, \dots, y_k \in \overline{B}_1^E(0)$

$$\begin{aligned} & d^k(f * g)(x, y_1, \dots, y_k) \\ & = \sum_{j=2}^k \sum_{P \in P_{k,j}} d^j f(h(x), d^{|I_1|} h(x, y_{I_1}), \dots, d^{|I_j|} h(x, y_{I_j})) \end{aligned} \quad (\text{A.1})$$

$$+ (\text{id}_E + f'(x))(d^k g(x, y_1, \dots, y_k)), \quad (\text{A.2})$$

with $h := \text{id}_\Omega + g$. The norm of the summand in (A.2) is bounded by

$$(1 + \|f'\|_{\infty, \text{op}}) \|g^{(k)}\|_{\infty, \text{op}} \leq e^{\|f\|_{C^k}} \|g\|_{C^k}; \quad (\text{A.3})$$

this is $\leq M_k e^{k(\|f\|_{C^k} + \|g\|_{C^k})} \|g\|_{C^k}$. The norm of a summand in (A.1) is

$$\leq \|f^{(j)}\|_{\infty, \text{op}} \|h^{(|I_1|)}\|_{\infty, \text{op}} \cdots \|h^{(|I_j|)}\|_{\infty, \text{op}} \leq \|f\|_{C^k} \|h^{(|I_1|)}\|_{\infty, \text{op}} \cdots \|h^{(|I_j|)}\|_{\infty, \text{op}}.$$

For $a \in \{1, \dots, j\}$, possible cases are that $|I_a| = 1$; then

$$\|h^{(|I_a|)}\|_{\infty, \text{op}} = \|\text{id}_E + g'\|_{\infty, \text{op}} \leq 1 + \|g'\|_{\infty, \text{op}} \leq e^{\|g\|_{C^k}} = e^{|I_a|} \|g\|_{C^k}.$$

If $|I_a| \geq 2$, then

$$\|h^{(|I_a|)}\|_{\infty, \text{op}} = \|g^{(|I_a|)}\|_{\infty, \text{op}} \leq \|g\|_{C^k} \leq e^{\|g\|_{C^k}} \leq e^{|I_a|} \|g\|_{C^k}.$$

As

$$e^{|I_1| \|g\|_{C^k}} \dots e^{|I_j| \|g\|_{C^k}} = e^{k \|g\|_{C^k}} \leq M_{|I_1|} \dots M_{|I_j|} e^{k(\|f\|_{C^k} + \|g\|_{C^k})},$$

we get we get

$$\begin{aligned} \|(f * g)^{(k)}\|_{\infty, \text{op}} &\leq \left(\sum_{j=2}^k \sum_{P \in P_{k,j}} M_{|I_1|} \dots M_{|I_j|} \right) \|f\|_{C^k} e^{k(\|f\|_{C^k} + \|g\|_{C^k})} \\ &\quad + M_k e^{k(\|f\|_{C^k} + \|g\|_{C^k})} \|g\|_{C^k} \\ &\leq M_k e^{k(\|f\|_{C^k} + \|g\|_{C^k})} (\|f\|_{C^k} + \|g\|_{C^k}). \end{aligned}$$

Thus $\|f * g\|_{C^k} \leq M_k e^{k(\|f\|_{C^k} + \|g\|_{C^k})} (\|f\|_{C^k} + \|g\|_{C^k})$, whence (4.1) holds for $L = 2$. If we are given $f_1, \dots, f_L \in \mathcal{V}$ with $L \geq 3$ and

$$\|f_1\|_{C^k} + \dots + \|f_L\|_{C^k} \leq 1, \quad (\text{A.4})$$

let $f := f_L$ and $g := f_{L-1} * \dots * f_1$. Using (A.3) and the inductive hypothesis (the case of $L - 1$ factors), we see that the right-hand side of (A.3) is less or equal

$$\begin{aligned} e^{\|f_1\|_{C^k}} \|g\|_{C^k} &\leq e^{\|f_L\|_{C^k}} M_k e^{k(\|f_{L-1}\|_{C^k} + \dots + \|f_1\|_{C^k})} (\|f_{L-1}\|_{C^k} + \dots + \|f_1\|_{C^k}) \\ &\leq M_k e^{k \sum_{j=1}^L \|f_j\|_{C^k}} (\|f_{L-1}\|_{C^k} + \dots + \|f_1\|_{C^k}). \end{aligned}$$

With $h := \text{id}_\Omega + g = (\text{id}_\Omega + f_{L-1}) \circ \dots \circ (\text{id}_\Omega + f_1)$, the norm of a summand in (A.1) is bounded by

$$\|f_L\|_{C^k} \|h^{(I_1)}\|_{\infty, \text{op}} \dots \|h^{(I_j)}\|_{\infty, \text{op}}.$$

For $a \in \{1, \dots, j\}$, possible cases are that $|I_a| = 1$; then

$$\begin{aligned} \|h^{(I_a)}\|_{\infty, \text{op}} &= \|((\text{id}_\Omega + f_{L-1}) \circ \dots \circ (\text{id}_\Omega + f_1))'\|_{\infty, \text{op}} \\ &\leq (1 + \|f_{L-1}\|_{\infty, \text{op}}) \dots (1 + \|f_1\|_{\infty, \text{op}}) \\ &\leq (1 + \|f_{L-1}\|_{C^k}) \dots (1 + \|f_1\|_{C^k}) \leq e^{|I_a|(\|f_{L-1}\|_{C^k} + \dots + \|f_1\|_{C^k})} \\ &\leq M_{|I_a|} e^{|I_a|(\|f_1\|_{C^k} + \dots + \|f_L\|_{C^k})}. \end{aligned}$$

If $|I_a| \geq 2$, then

$$\begin{aligned} \|h^{(I_a)}\|_{\infty, \text{op}} &= \|g^{(I_a)}\|_{\infty, \text{op}} \leq \|g\|_{C^{|I_a|}} \\ &\leq M_{|I_a|} e^{|I_a|(\|f_{L-1}\|_{C^k} + \dots + \|f_1\|_{C^k})} (\|f_{L-1}\|_{C^k} + \dots + \|f_1\|_{C^k}) \\ &\leq M_{|I_a|} e^{|I_a|(\|f_1\|_{C^k} + \dots + \|f_L\|_{C^k})}, \end{aligned}$$

using (A.4) for the final estimate. As

$$e^{|I_1|(\|f_1\|_{C^k} + \dots + \|f_L\|_{C^k})} \dots e^{|I_j|(\|f_1\|_{C^k} + \dots + \|f_L\|_{C^k})} = e^{k(\|f_1\|_{C^k} + \dots + \|f_L\|_{C^k})},$$

we get

$$\begin{aligned} \|(f_L * \dots * f_1)^{(k)}\|_{\infty, \text{op}} &\leq \left(\sum_{j=2}^k \sum_{P \in P_{k,j}} M_{|I_1|} \dots M_{|I_j|} \right) \|f_1\|_{C^k} e^{k(\|f_1\|_{C^k} + \dots + \|f_L\|_{C^k})} \\ &\quad + M_k e^{k(\|f_1\|_{C^k} + \dots + \|f_L\|_{C^k})} (\|f_2\|_{C^k} + \dots + \|f_L\|_{C^k}) \\ &\leq M_k e^{k \sum_{j=1}^L \|f_j\|_{C^k}} (\|f_1\|_{C^k} + \dots + \|f_L\|_{C^k}), \end{aligned}$$

whence $\|f_L * \dots * f_1\|_{C^k} \leq M_k e^{k \sum_{j=1}^L \|f_j\|_{C^k}} (\|f_1\|_{C^k} + \dots + \|f_L\|_{C^k})$. This completes the proof. \square

Remark A.1. Proposition 4.1 was inspired by a recent result from the theory of evolution equations on infinite-dimensional Lie groups (regularity theory). Consider a Lie group G modeled on a locally convex space E , with neutral element e . Let TG be the tangent bundle, $\mathfrak{g} := T_e G$ and $G \times TG \rightarrow TG$, $(g, v) \mapsto g.v := TL_g(v)$ be the natural left action of G on TG , which uses the tangent map TL_g of the left translation $L_g: G \rightarrow G$, $x \mapsto gx$. The Lie group G is called C^0 -regular if, for each continuous path $\gamma: [0, 1] \rightarrow \mathfrak{g}$, there exists a C^1 -map $\eta: [0, 1] \rightarrow G$ with $\eta(0) = e$ and

$$\eta'(t) = \eta(t).\gamma(t) \text{ for all } t \in [0, 1];$$

and moreover the time 1-map $C([0, 1], \mathfrak{g}) \rightarrow G$, $\gamma \mapsto \eta(1)$ is smooth (endowing the domain with the topology of uniform convergence); see [12, 28]. As shown by Hanusch [18], every C^0 -regular Lie group is *locally μ -convex* in the sense of [14]. Thus, for each open identity neighborhood $U \subseteq G$ and C^∞ -diffeomorphism $\phi: U \rightarrow V \subseteq E$ with $\phi(e) = 0$, and each continuous seminorm $q: E \rightarrow [0, \infty[$, there exists a continuous seminorm $p: E \rightarrow [0, \infty[$ such that

$$g_L \cdots g_2 g_1 \in U$$

and

$$q(\phi(g_L \cdots g_1)) \leq p(\phi(g_L)) + \cdots + p(\phi(g_1))$$

for all $L \in \mathbb{N}$ and $g_1, \dots, g_L \in U$ such that $p(\phi(g_L)) + \cdots + p(\phi(g_1)) \leq 1$. As shown in [14], the Lie group $\text{Diff}_\partial(\Omega)$ is C^0 -regular. Applying the preceding fact to the C^∞ -diffeomorphism $\kappa: \text{id}_\Omega + \mathcal{V} \rightarrow \mathcal{V}$, $g \mapsto g - \text{id}_\Omega$ and the seminorm $q := \|\cdot\|_{C^k}$ with a fixed $k \in \mathbb{N}_0$, we infer the existence of a continuous seminorm p on $C_\partial^\infty(\Omega, \mathbb{R}^d)$ such that

$$\|((\text{id}_\Omega + f_L) \circ \cdots \circ (\text{id}_\Omega + f_1)) - \text{id}_\Omega\|_{C^k} \leq p(f_1) + \cdots + p(f_L)$$

for all $L \in \mathbb{N}$ and $f_1, \dots, f_L \in \mathcal{V}$ with $p(f_1) + \cdots + p(f_L) \leq 1$. We may assume that p is a multiple $r\|\cdot\|_{C^\ell}$ of $\|\cdot\|_{C^\ell}$ for some $\ell \geq k$, as these seminorms define the topology on $C_\partial^\infty(\Omega, \mathbb{R}^d)$. The point of Proposition 4.1 is that we can always choose $\ell = k$, and that it provides an explicit choice for the constant $r > 0$.

Previously, μ -regularity has been discussed in quantitative form only for Banach-Lie groups, where estimates for the Baker-Campbell-Hausdorff multiplication $*$ on the Banach-Lie algebra \mathfrak{g} enable norms $\|x_1 * \cdots * x_m\|$ of products of elements $x_1, \dots, x_m \in \mathfrak{g}$ to be estimated (see [12]).

B Practical upper bounds for C^k -norm

This appendix describes the methods that were used for the values in figure 4.1. The goal is to estimate how the practically achieved C^k -norms compare to the theoretical results of proposition 4.1. For simplicity, we stick to the one-dimensional case. Since both sides of the inequality 4.1 relies on the chosen vector fields f_ℓ , we start by reorganizing the inequality according to

$$\frac{\|((\text{id}_\Omega + f_L) \circ \cdots \circ (\text{id}_\Omega + f_1)) - \text{id}_\Omega\|_{C^k}}{\left(\sum_\ell^L \|f_\ell\|_{C^k}\right) e^{k \sum_\ell^L \|f_\ell\|_{C^k}}} \leq M_k. \quad (\text{B.1})$$

Estimating C^k -norms

Let $X = \{x_i\}_{i=1}^N \in \Omega$ be a collection of points tightly distributed throughout the domain Ω . Then we estimate the C^k -norm of a function $f \in \Omega$ by $\|f\|_{C^k} = \|D^k f\|_{\text{op}, \infty} = \max_{x \in \Omega} \|D^k f(x)\|_{\text{op}} \approx \max_{x_i \in X} \|D^k f(x_i)\|_{\text{op}}$. In the one-dimensional case, this reduces to

$$\max_{x_i \in X} \left| \frac{\partial^k f}{\partial x^k}(x_i) \right| \quad (\text{B.2})$$

where X is typically chosen as an equidistant grid of points $X = \{i/N \mid i = 0, \dots, N\}$.

Choosing vector fields

Consider vector fields of the form

$$f_\ell(x) = \sum_{j=1}^M w_{\ell,j} \varphi_j(x), \quad (\text{B.3})$$

for basis functions as given in (5.3) and some weight vector $\mathbf{w}_\ell \in \mathbb{R}^M$. Since different weight vectors might yield different results for the relative norm estimates, we use two different strategies for initializing the weight vectors:

1. Deterministic vector of ones: $w_{\ell,j} = 1$, $j = 1, \dots, M$, $\ell = 1, \dots, L$, and
2. Normal random sampling: $w_{\ell,j} \sim \mathcal{N}(\mu = 0, \sigma^2 = 1)$.

The values of (B.2) presented in figure 4.1 are chosen as the largest attained value from strategy one and 500 runs of strategy 2.

Ensuring the assumptions hold

The weight vectors cannot be chosen freely, but are assumed to adhere to a unit norm-sum constraint. To ensure that this condition holds, we will typically start out with an initial set of weight vectors, which will be scaled onto the feasible set. Let $W = \{\mathbf{w}_\ell\}_{\ell=1}^L$ be a collection of weight vectors, from which we create a set of vector fields f_ℓ , $\ell = 1, \dots, L$. Moreover, assume that for this set of vector fields

$$\sum_{\ell=1}^L \|f_\ell\|_{C^k} > 1. \quad (\text{B.4})$$

The scaling will be performed in a way that is similar to the methods of section 5.1; Find a scalar $\alpha \in (0, 1]$ such that $\alpha \sum_{\ell=1}^L \|f_\ell\|_{C^k} \leq 1$, i.e.

$$\alpha = \min \left\{ 1, \frac{1}{\sum_{\ell=1}^L \|f_\ell\|_{C^k}} \right\}.$$

and then choose $\tilde{W} = \{\alpha \mathbf{w}_\ell\}_{\ell=1}^L$.

Competing Interests

The authors declare no competing interests related to the contents of the article.

References

1. Agrachev, A., Caponigro, M.: Controllability on the group of diffeomorphisms. *Annales de l'Institut Henri Poincaré C, Analyse non linéaire* **26**(6), 2503–2509 (2009). DOI 10.1016/j.anihpc.2009.07.003
2. Agrachev, A., Sarychev, A.: Control on the manifolds of mappings with a view to the deep learning. *Journal of Dynamical and Control Systems* (2021). DOI 10.1007/s10883-021-09561-2
3. Bauer, M., Bruveris, M., Marsland, S., Michor, P.W.: Constructing reparameterization invariant metrics on spaces of plane curves. *Differential Geometry and its Applications* **34**, 139–165 (2014). DOI 10.1016/j.difgeo.2014.04.008

4. Behrmann, J., Grathwohl, W., Chen, R.T.Q., Duvenaud, D., Jacobsen, J.H.: Invertible residual networks. In: K. Chaudhuri, R. Salakhutdinov (eds.) *Proceedings of the 36th International Conference on Machine Learning, *Proceedings of Machine Learning Research**, vol. 97, pp. 573–582. PMLR, Long Beach, California, USA (2019). URL <https://proceedings.mlr.press/v97/behrmann19a.html>
5. Behrmann, J., Vicol, P., Wang, K.C., Grosse, R., Jacobsen, J.H.: Understanding and mitigating exploding inverses in invertible neural networks. In: A. Banerjee, K. Fukumizu (eds.) *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics, *Proceedings of Machine Learning Research**, vol. 130, pp. 1792–1800. PMLR, Virtual (2021). URL <https://proceedings.mlr.press/v130/behrmann21a.html>
6. Benning, M., Celledoni, E., Ehrhardt, M.J., Owren, B., Schönlieb, C.B.: Deep learning as optimal control problems: models and numerical methods. *Journal of Computational Dynamics* **6**(2), 171–198 (2019). DOI 10.3934/jcd.2019009. URL <http://arxiv.org/abs/1904.05657>
7. Celledoni, E., Ehrhardt, M.J., Etmann, C., Mclachlan, R.I., Owren, B., Schönlieb, C.B., Sherry, F.: Structure-preserving deep learning. *European Journal of Applied Mathematics* **32**(5), 888–936 (2021). DOI 10.1017/s0956792521000139
8. Celledoni, E., Eidnes, S., Eslitzbichler, M., Schmeding, A.: Shape analysis on Lie groups and homogeneous spaces. In: F. Nielsen, F. Barbaresco (eds.) *Geometric Science of Information*, pp. 49–56. Springer International Publishing, Cham (2017). DOI 10.1007/978-3-319-68445-1_6
9. Celledoni, E., Eslitzbichler, M., Schmeding, A.: Shape analysis on Lie groups with applications in computer animation. *Journal of Geometric Mechanics* **8**(3), 273–304 (2016). DOI 10.3934/jgm.2016008
10. Clark, D.E., Houssineau, J.: Faa di bruno’s formula for gateaux differentials and interacting stochastic population processes (2012). DOI 10.48550/ARXIV.1202.0264
11. Cybenko, G.: Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems* **2**(4), 303–314 (1989). DOI 10.1007/bf02551274
12. Glöckner, H.: Regularity properties of infinite-dimensional lie groups, and semiregularity (2012). DOI 10.48550/ARXIV.1208.0715
13. Glöckner, H.: Diffeomorphism groups of convex polytopes (2022). DOI 10.48550/ARXIV.2203.09285
14. Glöckner, H., Neeb, K.H.: Diffeomorphism groups of compact convex sets. *Indagationes Mathematicae* **28**(4), 760–783 (2017). DOI 10.1016/j.indag.2017.04.004
15. Glöckner, H., Neeb, K.H.: *Infinite-Dimensional Lie Groups* (2022). Book in preparation.
16. Haber, E., Ruthotto, L.: Stable architectures for deep neural networks. *Inverse Problems* **34**(1), 014,004 (2017). DOI 10.1088/1361-6420/aa9a90
17. Hamilton, R.S.: The inverse function theorem of Nash and Moser. *Bulletin of the American Mathematical Society* **7**(1), 65–222 (1982). DOI 10.1090/s0273-0979-1982-15004-2
18. Hanusch, M.: Regularity of Lie groups. *Communications in Analysis and Geometry* **30**(1), 53–152 (2022). DOI 10.4310/cag.2022.v30.n1.a2. URL <https://doi.org/10.4310/cag.2022.v30.n1.a2>
19. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. *IEEE Conference on Computer Vision and Pattern Recognition*. (1), 770–778 (2016)
20. Jermyn, I.H., Kurtek, S., Klassen, E., Srivastava, A.: Elastic shape matching of parameterized surfaces using square root normal fields. In: A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, C. Schmid (eds.) *Computer Vision – ECCV 2012*, pp. 804–817. Springer Berlin Heidelberg, Berlin, Heidelberg (2012). DOI 10.1007/978-3-642-33715-4_58
21. Keys, R.: Cubic convolution interpolation for digital image processing. *IEEE Transactions on Acoustics, Speech, and Signal Processing* **29**(6), 1153–1160 (1981). DOI 10.1109/tassp.1981.1163711
22. Klassen, E., Michor, P.W.: Closed surfaces with different shapes that are indistinguishable by the srnf (2019). DOI 10.48550/ARXIV.1910.10804
23. Kurtek, S., Klassen, E., Ding, Z., Srivastava, A.: A novel riemannian framework for shape analysis of 3d objects. In: *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1625–1632 (2010). DOI 10.1109/CVPR.2010.5539778

24. Kurtek, S., Klassen, E., Gore, J.C., Ding, Z., Srivastava, A.: Elastic geodesic paths in shape space of parameterized surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **34**(9), 1717–1730 (2012). DOI 10.1109/tpami.2011.233
25. Mani, M., Kurtek, S., Barillot, C., Srivastava, A.: A Comprehensive Riemannian Framework for the Analysis of White Matter Fiber Tracts. In: IEEE (ed.) *IEEE-ISBI*, pp. 1101–1104. IEEE, Rotterdam, Netherlands (2010). DOI 10.1109/ISBI.2010.5490185. URL <https://www.hal.inserm.fr/inserm-00723805>
26. Michor, P.W.: *Manifolds of Differentiable Mappings*. Shiva Pub., Orpington (1980)
27. Milnor, J.: Remarks on infinite-dimensional Lie groups. In: *Relativity, groups and topology*. 2 (1984)
28. Neeb, K.H.: Towards a Lie theory of locally convex groups. *Japanese Journal of Mathematics* **1**(2), 291–468 (2006). DOI 10.1007/s11537-006-0606-y
29. Nocedal, J., Wright, S.: *Numerical Optimization (Springer Series in Operations Research and Financial Engineering)*. Springer, New York (2006). DOI <https://doi.org/10.1007/978-0-387-40065-5>
30. Omori, H.: *Infinite Dimensional Lie Transformations Groups*. Springer-Verlag, Berlin (1974). DOI 10.1007/bfb0063400
31. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: Pytorch: An imperative style, high-performance deep learning library. In: H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, R. Garnett (eds.) *Advances in Neural Information Processing Systems*, vol. 32. Curran Associates, Inc., Vancouver, BC, Canada (2019). URL <https://proceedings.neurips.cc/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf>
32. Riordan, J.: The blossoming of Schröder's fourth problem. *Acta Mathematica* **137**(0), 1–16 (1976). DOI 10.1007/bf02392410
33. Riseth, J.N.: Gradient-based optimization in shape analysis for reparametrization of parametric curves and surfaces. Master's thesis, NTNU (2021). URL <https://hdl.handle.net/11250/2778386>
34. Schröder, E.: Vier combinatorische Probleme. *Z. Math. Phys* **15**, 361–376 (1870)
35. Srivastava, A., Klassen, E., Joshi, S.H., Jermyn, I.: Shape analysis of elastic curves in euclidean spaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **33**(7), 1415–1428 (2011). DOI 10.1109/tpami.2010.184
36. Su, Z., Bauer, M., Preston, S.C., Laga, H., Klassen, E.: Shape analysis of surfaces using general elastic metrics. *Journal of Mathematical Imaging and Vision* **62**(8), 1087–1106 (2020). DOI 10.1007/s10851-020-00959-4
37. Tabak, E.G., Vanden-Eijnden, E.: Density estimation by dual ascent of the log-likelihood. *Communications in Mathematical Sciences* **8**(1), 217–233 (2010)
38. Ulyanov, D., Vedaldi, A., Lempitsky, V.: Deep image prior. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 9446–9454 (2018)
39. Weinan, E.: A proposal on machine learning via dynamical systems. *Communications in Mathematics and Statistics* **5**(1), 1–11 (2017)