Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

An operator-splitting approach for variational optimal control formulations for diffeomorphic shape matching

Andreas Mang, Jiwen He *, Robert Azencott

University of Houston, 3551 Cullen Blvd, Houston, TX 77204-3008, USA

ARTICLE INFO

Article history:
Received 28 February 2023
Received in revised form 15 July 2023
Accepted 16 August 2023
Available online 30 August 2023

Dedicated to the memory of Roland Glowinski

Keywords:
Diffeomorphic shape matching
Optimal control
Numerical optimization
Operator splitting
Shape classification

ABSTRACT

We present formulations and numerical algorithms for solving diffeomorphic shape matching problems. We formulate shape matching as a variational problem governed by a dynamical system that models the flow of diffeomorphism $f_t \in \mathrm{diff}(\mathbb{R}^3)$. We overview our contributions in this area, and present an improved, matrix-free implementation of an operator splitting strategy for diffeomorphic shape matching. We showcase results for diffeomorphic shape matching of real clinical cardiac data in $\mathbb{R}^3$ to assess the performance of our methodology.

1. Introduction

In (bio)medical imaging applications, the automatic matching of k-dimensional deformable shapes across subjects or multi-temporal data of individual patients is a critical step to aid clinical diagnosis [1–8]. From a mathematical point of view, this matching problem constitutes an *inverse problem* [3]: Given two (or more) shapes $s_i \in \mathcal{S}$, $i = 0, 1$, in $\mathbb{R}^d$ representing an object/anatomy of interest, defined in some shape space $\mathcal{S}$, we seek a *plausible* spatial transformation $y \in \mathcal{Y}_{\mathrm{ad}}$, $\mathcal{Y}_{\mathrm{ad}} \subseteq \{\phi \mid \phi : \mathbb{R}^d \to \mathbb{R}^d\}$ with $d \in \{2, 3\}$ that establishes a point-wise correspondence between these objects [1,2]. Based on the problem specific notion of plausibility of an admissible transformation y, various continuum mechanical models emerged to restrict the space of admissible transformations $\mathcal{Y}_{\mathrm{ad}}$. In a variational setting, this prior knowledge on admissible maps y is typically prescribed through regularization functionals or constraints. Examples include models of elasticity [9–11] or incompressibility [12–16].

The preferred model typically depends on the application, of which there are many in medical imaging. Excellent reference works giving an overview of developments in this field are [1–7]. First, one may simply be interested in relating local information observed in two views of the same object. This can be helpful in assessing, for example, changes in texture or morphology that occurred over time due to the progression of a disease or treatment [8,17,18]. Moreover, one can integrate data acquired by multiple sensors (different imaging modalities) into a common reference frame (between imaging sessions, the patient/organs move). Likewise, one can compensate for organ motion associated with breathing or the heart cycle [19–23]. Registration has also been used for the construction of atlases [8,24–28] and atlas-based segmenta-

* Corresponding author.
E-mail addresses: andreas@math.uh.edu (A. Mang), jiwenhe@math.uh.edu (J. He), razencott@uh.edu (R. Azencott).

https://doi.org/10.1016/j.jcp.2023.112463

tion [29–31]. Aside from these examples of integrating data into a common frame of reference, one can use these tools to study changes in the (morphological) appearance of organs. This observation has lead to the notion of *computational anatomy* [19,32–38]. The results presented in this work fall into this category. Here, one uses the computed maps y (typically, diffeomorphisms) to derive features that might be useful to classify or characterize patients (e.g., diseased versus healthy) [18,39–45]. These features include geodesic distances, strain values, or local volume change associated with the computed map y (we specify this mathematically more rigorously below). One can either compare patients to one another [46], or, likewise, individual patients to a standard reference template [24,25,27] and then—based on the derived features—try to classify them. Similarly, one can monitor a patient over time and use the computed maps between time frames to derive features that can be used to assess the progression or existence of a pathology. We consider both applications in the present work.

We limit $\mathcal{Y}_{\text{ad}}$ to infinite dimensional groups of diffeomorphisms $\mathcal{G} \subseteq \text{diff}(\mathbb{R}^3)$ in $\mathbb{R}^3$. This approach provides a rich mathematical framework for studying and quantifying shape variability in $\mathbb{R}^3$ through the lens of geodesics in the group of diffeomorphisms $\mathcal{G}$. In the context of biomedical imaging, this area of research is—as mentioned above—typically referred to as computational anatomy [34–38]. The underlying mathematical framework for diffeomorphic shape matching is referred to as *large deformation diffeomorphic metric mapping* (**LDDMM**) [6,47–52] and builds upon the seminal works [53–55]. In [53–55] it was shown that in an incompressible fluid, if $f_t(x)$, $f_t(x) := f(t,x)$, is the position at time t of a particle originating from initial location x, then the map $t \to f_t(x)$ defines a geodesic in $\mathcal{G}$ for the metric defined by the kinetic energy of the fluid (see (2)). This observation has been used in numerous works to design variational optimal control formulations that allow us to match two shapes s_0 and s_1 by computing the "shortest" geodesic in $\mathcal{G}$ that connects them, i.e., maps s_0 to s_1.

1.1. Contributions

We provide an exposition of our work in the area of diffeomorphic shape matching, including the design of numerical algorithms [56–62] and their application to clinical problems [46,63–67]. In addition, we present an improved numerical implementation of the operator-splitting algorithm originally proposed in [61,62]. The work in [61,62] uses direct methods [68,69] to solve the optimality systems, i.e., matrices that appear in our optimality systems are formed and stored explicitly. Our new implementation is matrix-free. That is, we either implement matrix-vector products instead of forming matrices and/or exploit the block-structure of the matrix operators to reduce the memory footprint. Moreover, we consider a Schur complement (reduced-space) method [70] and use iterative algorithms to solve the individual subproblems that appear in our optimality system. Overall, we obtain a speedup of approximately 2× compared to our prior work [61,62]. In addition, the matrix-free implementation reduces the memory footprint of our method, which allows us to solve larger problems compared to our prior work.

1.2. Related work

We present numerical algorithms for solving optimal control problems for diffeomorphic shape matching and their application to cardiac imaging. We refer the reader to [1,2,5,6] for a general introduction into numerical methods, applications, and formulations for image registration and diffeomorphic shape matching. There exists a plurality of mathematical and numerical frameworks to study shape variability; examples include [71–83]. The work presented in this manuscript builds upon the rich mathematical framework developed in [6,47–51,53–55,71,84,85]. In this framework, we define the shape space $\mathcal{S}$ as orbits of given template shapes under the action of the group of diffeomorphisms $\mathcal{G}$. Our past work for the matching of smooth surfaces has been described in [56–60,62]. Likewise to [6,47–51], we consider a variational approach for solving for the diffeomorphism $y \in \mathcal{G}$ that maps one shape $s_0 \in \mathcal{S}$ to another shape $s_1 \in \mathcal{S}$ such that $y \cdot s_0 \approx s_1$. These shapes s_i, $i = 0, 1$, are typically represented as curves, lines, or surfaces in $\mathbb{R}^d$ with $d \in \{2, 3\}$ [57,86].

Our work in this area was initiated by a short exposition of the mathematical problem formulation [56]. This joint paper of Azencott and Glowinski represents the foundation for a series of papers of our group, spanning more than one decade [46,56–60,62,65–67]. In [57] Azencott, Glowinski, He, and co-authors build up on [56] and develop a first, efficient gradient descent algorithm for the considered optimal control formulation [56]. They applied their method to datasets of mitral valves, initiating a long-term collaboration between the Department of Mathematics at the University of Houston and the Houston Methodist Hospital. This collaboration has resulted in several joint publications over the years [46,62–67]. Improvements to the gradient descent (adjoint-based) methods and associated numerical strategies originally proposed in [57] led to several doctoral research projects [58–61]. This work culminated in the development of an operator-splitting strategy to solve the associated control problem [62]—a numerical technique that is equivalent to the *Douglas–Rachford* splitting method [87–92]. This approach can be traced back to the seminal works of Glowinski and Marrocco [93] and Gabay and Mercier [94]. The present work describes our current numerical implementation of this approach, and showcases its application to real world problems in biomedical imaging.

Related work on adjoint-based, variational approaches for diffeomorphic registration of images by our group is described in [15,16,95–102]. As opposed to the work considered here, these approaches directly operate on image intensities rather than shape representations derived from these data. Other work on numerical algorithms for the solution of variational shape matching problems (for images as well as surface representations) of other groups are, e.g., described in [51,103–110]. The majority of existing works consider an *optimize-then-discretize* approach to solve the variational problem [15,16,51,95,96,107,109–113]. Aside from our work [57,62,96], *discretize-then-optimize* approaches for LDDMM can be found in [106,114]. The problem formulation as well as our discretization naturally leads to a reproducing kernel Hilbert space (**RKHS**) structure [115]. Exploiting an RKHS representation of smooth functions in the context of diffeomorphic shape matching (and associated problems) is a common strategy considered in numerous works [38,49,110,116–118]. As for the numerical solution of the variational optimization problem, many existing works consider (first order) gradient descent-type optimization algorithms; see, e.g., [51,103,107,109,119]. Higher order (quasi-)Newton algorithms have, e.g., been developed in [15,16,98,105,112]. Related work on operator-splitting algorithms for LDDMM (and related problems) can be found in [120,121]. Other recent works that do not explicitly derive optimality conditions based on variational principles but rely on automatic differentiation can be found in [79,118,122–124]. Lastly, we note that the success of machine learning in various scientific disciplines has led to several recent works that attempt to solve the inverse problem of diffeomorphic registration based on machine learning techniques [79,125–134].

1.3. Outline

We present our mathematical formulation and our numerical approaches in §2. The parameterization of diffeomorphic matching in terms of a time dependent velocity field v is presented in §2.1. The associated ODE constitutes the state equation for our nonlinear control problem. The variational problem formulation is described in §2.2. We present the numerical discretization in §2.3. We describe our approach for numerical optimization in §2.4. We list quantitative measures of shape variability derived from diffeomorphic shape matching in §2.5. We report numerical experiments in §3 and conclude with §4.

2. Mathematical treatment and numerical methods

In the following sections, we describe the overall mathematical formulation and different numerical strategies to approach the problem. For easier access to the material, we provide an overview of our notation in Table 1.

Diffeomorphic shape matching can be formulated as follows: Given two k-dimensional shapes $s_0 \in \mathcal{S}$ and $s_1 \in \mathcal{S}$ in $\mathbb{R}^3$ with $k \in \{1,2,3\}$ we seek an $\mathbb{R}^3$-diffeomorphism $y \in \text{diff}(\mathbb{R}^3)$ such that $y \cdot s_0 = y(s_0) = s_1$. This problem is ill-posed; it requires regularization for a numerical treatment [135]. In the following, we describe the mathematical formulation and different numerical strategies to solve the variational optimization problem presented below. One can either match biomedical imaging data by diffeomorphic image registration operating directly on image intensities (see, e.g., [15,51,95,96,98,107,109,112]), or one can match shape representations extracted from these data (see Fig. 1). Here, we focus on the latter approach. In the present work, these shapes are represented as "surfaces." The term "surface" denotes a compact smooth 2-dimensional manifold with boundary, smoothly embedded in $\mathbb{R}^3$. The boundary of such a surface is a smooth 1-dimensional manifold smoothly embedded in $\mathbb{R}^3$, and—in all the cases considered here—is a connected finite union of smooth curves. We note that the general framework presented here applies (and has been applied) to curves, lines, or boundaries of solids embedded in $\mathbb{R}^3$. We parametrize the considered smooth surfaces with boundaries as finite sets of points in $\mathbb{R}^3$ (see §2.3).

2.1. Flows of diffeomorphisms

Let $k := (k_1, k_2, k_3) \in \mathbb{N}^3$ denote a multi-index and ∂^k denote the differential operator of order $|k| = \sum_{i=1}^3 k_i$. Moreover, let $q \in \mathbb{N}$, $1 \le p \le \infty$, and $u : \mathbb{R}^3 \supseteq \Omega \to \mathbb{R}$. Let

$$C^q(\Omega) := \left\{ u : \Omega \to \mathbb{R} : \partial^k u \text{ is continuous for } |k| \le q \right\}$$

denote the space of q-times continuously differentiable functions on $\Omega \subseteq \mathbb{R}^3$. Moreover, let

$$W^{q,p}(\Omega) := \left\{ u \in L^p(\Omega) : \partial^k u \in L^p(\Omega) \text{ for } 0 \le |k| \le q \right\}$$

denote the Sobolev space with norm

$$\|u\|_{q,p} := \begin{cases} \sum_{0 \le |k| \le q} \left(\|\partial^k u\|_p^p \right)^{1/p} & \text{if } 1 \le p < \infty, \\ \max_{0 \le |k| \le q} \|\partial^k u\|_\infty & \text{if } p = \infty, \end{cases}$$

where $\|\cdot\|_\infty$ denotes the standard supremum norm. Using these definitions, we denote by $C_0^q(\Omega)^3 \subset C^q(\Omega)^3$ with $q \in \mathbb{N}$ the completion of the space of vector fields of class C^q who along with their derivatives of order less than or equal to q converge to zero at infinity. The space $C_0^q(\Omega)^3$ is a Banach space for the norm $\|u\|_{q,\infty}$. Similarly, we define the Sobolev space $W_0^{q,p}(\Omega)^3$ as a space that consists of elements with compact support on $\Omega \subseteq \mathbb{R}^3$.

We introduce a pseudo-time variable $t \in [0,1]$, a suitable Hilbert space $\mathcal{H}$ of smooth vector fields in $\mathbb{R}^3$ (e.g., a RKHS; see §2.3.1), and parameterize diffeomorphisms using smooth vector fields $v \in \mathcal{V}$, $\mathcal{V} := L^2([0,1], \mathcal{H})$, $t \mapsto v_t := v(t,\cdot)$, $v_t \in \mathcal{H}$.

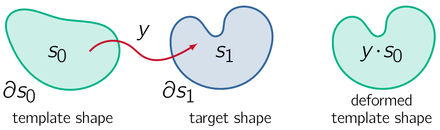

Fig. 1. Illustration of the diffeomorphic shape matching problem. We display two shapes $s_0, s_1 \in \mathcal{S}$ with smooth boundaries ∂s_0, ∂s_1. We seek a diffeomorphic map y that, if applied to s_0, makes the deformed template shape $y \cdot s_0$ similar to the target shape s_1, i.e., $y \cdot s_0 = y(s_0) = s_1$.

Table 1
Notation and symbols.

Variable/Acronym	Meaning
$\mathcal{S}$	shape space
$\mathcal{H}$	Hilbert space of $\mathbb{R}^3$ vector fields
$\mathcal{H}_\sigma^d$	RKHS of $\mathbb{R}^d$ vector fields parametrized by $\sigma > 0$ (we drop d for $d = 1$)
$\text{diff}(\mathbb{R}^3)$	space of diffeomorphic mappings from $\mathbb{R}^3$ to $\mathbb{R}^3$
$\mathcal{V}$	space of L^2-integrable (in time) smooth $\mathbb{R}^3$ vector fields
$\mathcal{G}$	group of diffeomorphisms; $\mathcal{G} \subseteq \text{diff}(\mathbb{R}^3)$
$\text{id}_{\mathbb{R}^3}$	identity map in $\mathbb{R}^3$; $\text{id}_{\mathbb{R}^3}(x) = x$ for any $x \in \mathbb{R}^3$
$s_0 \in \mathcal{S}$	template shape (shape to be deformed)
$s_1 \in \mathcal{S}$	target/fixed shape
$x_0 = (x_0^i)_{i=1}^{m_0} \in \mathbb{R}^{3m_0}$	vector of mesh points to represent s_0
$x_1 = (x_1^i)_{i=1}^{m_1} \in \mathbb{R}^{3m_1}$	vector of mesh points to represent s_1
$m_i \in \mathbb{N}$	number of points for the parameterization of s_i, $i = 0, 1$
$n \in \mathbb{N}$	number of cells to discretize time interval $[0, 1]$
$v : [0, 1] \times \mathbb{R}^3 \to \mathbb{R}^3$	time-dependent velocity field; $v_t := v(t, \cdot)$
$f : [0, 1] \times \mathbb{R}^3 \to \mathbb{R}^3$	diffeomorphic flow; $f_t := f(t, \cdot)$
$y := f_1 \;\; y : \mathbb{R}^3 \to \mathbb{R}^3$	diffeomorphic map at final time $t = 1$
$t \in [0, 1]$	pseudo-time variable
$a \in \mathbb{R}^{3m_0(p+1)}$	discrete control variable (control vector)
$x \in \mathbb{R}^{3m_0(p+1)}$	discrete state variable (state vector)
$u \in \mathbb{R}^{3m_0(p+1)}$	discrete dual variable corresponding to the state vector x
$w \in \mathbb{R}^{3m_0(p+1)}$	discrete dual variable corresponding to the control vector w
$\text{dist} : \mathcal{S} \times \mathcal{S} \to \mathbb{R}$	shape similarity measure
$\text{kin} : \mathcal{V} \to \mathbb{R}$	kinetic energy/regularization functional
$I_n = \text{diag}(1, \dots, 1) \in \mathbb{R}^{n,n}$	$n \times n$ identity matrix
$e_n = (1, \dots, 1) \in \mathbb{R}^n$	1-vector of size $n \times 1$
$\otimes$	Kronecker product (see (13))
$\odot$	Hadamard product (see (21))
ADMM	alternating direction method of multipliers
KKT	Karush–Kuhn–Tucker (conditions)
LDDMM	large deformation diffeomorphic metric mapping
RKHS	reproducing kernel Hilbert space
PCG	preconditioned conjugate gradient method

With this, we model the flow of $\mathbb{R}^3$-diffeomorphisms $f_t := f(t, \cdot)$ as the solution of the ordinary differential equation (the flow equation)

$$\begin{aligned} \partial_t f_t &= v_t(f_t) \quad \text{in } (0, 1], \\ f_0 &= \text{id}_{\mathbb{R}^3}, \end{aligned} \tag{1}$$

where $\text{id}_{\mathbb{R}^3} : \mathbb{R}^3 \to \mathbb{R}^3$, $\text{id}_{\mathbb{R}^3}(x) = x$, is the identity transformation in $\mathbb{R}^3$ and the vector field v_t tends to zero as $x \to \infty$. More precisely, we assume that v_t is an element of $C_0^q(\Omega)^3 \supseteq \mathcal{H}$, as defined above, for any $t \in [0, 1]$. This assumption, along with suitable regularity requirements in time guarantees that (1) admits a unique solution, and that the flow f_t is a diffeomorphism of Ω [6]. We refer to [6,50,57,85] for a more rigorous discussion. We have already specified above that v is L^2-integrable in time. However, we note that L^1-integrability is another option; see, e.g., [6]. The smoothness of $v_t \in \mathcal{H}$ is typically determined using a Sobolev norm $\| \cdot \|_{q,p}$ as defined above, where q defines the differentiability class and p the integrability order. This choice is motivated by the fact that the above result is true for vector fields v belonging to Banach or Hilbert spaces that are continuously embedded in $C_0^1(\Omega)^3$. A common and natural choice for p is again $p = 2$. The choice of q depends on the dimension d of the ambient space $\Omega \subseteq \mathbb{R}^d$, where $d \in \{2, 3\}$ in most applications. Based on the Sobolev embedding theorem (see, e.g., [136]), we can conclude that for $p = 2$ and $q > {}^d\!/\!_2 + 1$ the embedding $W_0^{q,2}(\Omega) \hookrightarrow C^1(\bar{\Omega})$ is compact. Since this embedding holds for all components of v_t, we have that $v_t \in \mathcal{H} = W^{q,2}(\Omega)^3$ with $q > {}^5\!/\!_2$ for $d = 3$ is an admissible space that yields a diffeomorphic flow f_t of smoothness class $1 \leq r < q - {}^3\!/\!_2$, $r \in \mathbb{N}$.

The set of all admissible flows $y := f_1$ at time $t = 1$ is a subgroup

$$\mathcal{G} := \{f_1 : \int_0^1 \|v_t\|_{\mathcal{H}} \,\mathrm{d}t < \infty\}$$

of C^r-diffeomorphisms in $\mathbb{R}^3$. This subgroup can be equipped with a right-invariant metric defined as the minimal path length of all geodesics joining two elements $\phi, \psi \in \mathcal{G}$. We refer, e.g., to [36,48,71,137] for a precise definition and mathematically rigorous discussion. The geodesic distance between $\mathrm{id}_{\mathbb{R}^3}$ and a mapping ϕ corresponds to the square root of the kinetic energy

$$\mathrm{kin}(v) := \|v\|^2_{L^2([0,1],\mathcal{H})} = \int_0^1 \|v_t\|^2_{\mathcal{H}} \,\mathrm{d}t \tag{2}$$

subject to the constraint that ϕ is equal to the solution of (1) at time $t = 1$ for the energy minimizing velocity v. We denote this geodesic distance by $\rho_{\mathcal{G}}(\mathrm{id}_{\mathbb{R}^3}, \phi)$. We have,

$$\rho_{\mathcal{G}}(\mathrm{id}_{\mathbb{R}^3}, \phi)^2 := \inf_{v \in \mathcal{V}} \left\{ \int_0^1 \|v_t\|^2_{\mathcal{H}} \,\mathrm{d}t : v \in \mathcal{V},\ \phi = f_1,\ \partial_t f_t = v_t(f_t),\ f_0 = \mathrm{id}_{\mathbb{R}^3} \right\}.$$

The geodesic distance between two diffeomorphic maps ϕ and ψ is given by $\rho_{\mathcal{G}}(\mathrm{id}_{\mathbb{R}^3}, \phi \circ \psi^{-1})^2$. Similarly, we can measure the distance between two shapes $s_0, s_1 \in \mathcal{S}$ in terms of the kinetic energy associated with the energy minimizing v_t that gives rise to the diffeomorphic flow f_t that maps the shape s_0 to the shape s_1. That is, we measure the distance between two shapes s_0 and s_1 as the smallest-cost deformation that maps s_0 to s_1. Using the geodesic distance $\rho_{\mathcal{G}}$ introduced above we obtain

$$\rho_{\mathcal{S}}(s_0, s_1) := \inf_{\phi \in \mathcal{G}} \left\{ \rho_{\mathcal{G}}(\mathrm{id}_{\mathbb{R}^3}, \phi) : s_1 = \phi \cdot s_0 \right\}. \tag{3}$$

Consequently, to compare two shapes s_0 and s_1, we have to find the energy minimizing velocity v for kin in (2) that parameterizes the map that maps s_0 to s_1. This is exactly the inverse problem we seek to solve in the present work. More precisely, we seek $y \in \mathcal{G}$ such that $s_1 = y \cdot s_0$, where $y := f_1$ is the end point at $t = 1$ of the geodesic f_t, $t \in [0, 1]$ satisfying (1).

2.2. Variational problem formulation

We consider diffeomorphic shape matching as a variational, optimal control problem. We relax the hard constraint $y \cdot s_0 = s_1$ by introducing a distance functional $\mathrm{dist} : \mathcal{S} \times \mathcal{S} \to \mathbb{R}$ that quantifies the proximity between $y \cdot s_0$ and s_1. We model the diffeomorphism y to be generated by some time dependent flow of smooth $\mathbb{R}^3$-vector fields v_t, as prescribed by the dynamical system in (1). We arrive at the nonlinear optimal control problem [51,57,85]

$$\begin{aligned} &\underset{(f_t) \in \mathcal{F},\ v \in \mathcal{V}}{\text{minimize}} \ \ \mathrm{dist}(f_1 \cdot s_0, s_1) + \mathrm{reg}(v) \\ &\text{subject to} \ \ \partial_t f_t = v_t(f_t) \quad \text{in } (0, 1], \\ &\qquad\qquad\quad f_0 = \mathrm{id}_{\mathbb{R}^3}. \end{aligned} \tag{4}$$

We specify the distance measure in (4) in §2.3.3. The regularization functional $\mathrm{reg} : \mathcal{V} \to \mathbb{R}$ corresponds to the kinetic energy in (2); more details can be found in §2.3.2.

Remark 1. We note that the formulation above has been extended to the matching of finite sequences of shapes [56,62,108]. For simplicity, we omit this case here.

2.3. Numerical discretization

We consider a *discretize-then-optimize* approach. Other works using a discretize-then-optimize approach for LDDMM are described in [96,106,114]. For a discussion of tradeoffs between *discretize-then-optimize* and *optimize-then-discretize* approaches we refer to [138].

In our work, we model the surfaces s_0 (template shape) and s_1 (target shape) by a mesh of points $x_0 = (x_0^i)_{i=1}^{m_0} \in \mathbb{R}^{3m_0}$ and $x_1 = (x_1^i)_{i=1}^{m_1} \in \mathbb{R}^{3m_1}$ in $\mathbb{R}^3$, respectively. We subdivide the time horizon $[0, 1]$ into n equispaced cells of size $h = 1/(n+1)$ on a nodal grid. The associated discrete time points are denoted by $t^j := (j-1)h$, $j = 1, \ldots, n+1$. We denote the state and control vectors of our problem by $x(t^j) = (x^i(t^j))_{i=1}^{m_0} \in \mathbb{R}^{3m_0}$ and $a(t^j) = (a^i(t^j))_{i=1}^{m_0} \in \mathbb{R}^{3m_0}$ with $j = 1, \dots, n+1$, respectively. The control $a^i(t^j) \in \mathbb{R}^3$ are coefficients associated with the ith state vector $x^i(t^j) \in \mathbb{R}^3$ at time t^j; they parameterize the continuous velocity field v (see §2.3.1 for details). We rearrange the vectors $x(t^j)$ and $a(t^j)$ for $j = 1, \dots, n+1$ in lexicographical orderings

$$x := (x(t^1), \dots, x(t^{n+1})) \in \mathbb{R}^{3m_0(n+1)} \quad \text{and} \quad a := (a(t^1), \dots, a(t^{n+1})) \in \mathbb{R}^{3m_0(n+1)}, \tag{5}$$

respectively, to be able to present the discretized version of the optimal control problem in (4) in a compact format.

Remark 2. We note that our implementation allows for matching shape representations discretized with a varying number $m_i \in \mathbb{N}$ of points $(x_i^j)_{j=1}^{m_i}$, $i \in \{0, 1\}$, despite the fact that the numerical experiments included in this study are performed for meshes x_i with an identical number of discretization points m_i.

2.3.1. Reproducing kernel Hilbert space representation

RKHS are a key element of diffeomorphic shape matching [38,49,57,110,116–118]. As we have stated above, we assume that v_t are smooth vector fields on $\mathbb{R}^3$. To guarantee this we have introduced an adequate Hilbert space $\mathcal{H}$, as outlined in §2.1. Assuming that the Hilbert space $\mathcal{H}$ of $\mathbb{R}^3$ vector fields is continuously embedded into $C_0^q(\Omega)^3$, $\Omega \subseteq \mathbb{R}^3$, for some $q \geq 1$, there exists a constant $c > 0$ such that

$$\|v_t\|_{q,\infty} \leq c\|v_t\|_{\mathcal{H}}$$

for some $v_t \in \mathcal{H}$, where $\|\cdot\|_{\mathcal{H}}$ denotes the Hilbert norm on $\mathcal{H}$ with associated inner product $\langle\cdot,\cdot\rangle_{\mathcal{H}}$. From this it follows that $\mathcal{H}$ is an RKHS. Since $\mathcal{H}$ is a Cartesian product space for vector fields in $\mathbb{R}^3$, the construction of the associated RKHS is slightly more involved than in the scalar case. In particular, we have to assume that the kernel $\mathrm{ker}: \mathbb{R}^3 \times \mathbb{R}^3 \to \mathbb{R}^{3,3}$ is matrix valued. Based on the Riesz representation theorem we know that there exists a matrix valued kernel function such that for all $z, a \in \mathbb{R}^3$ the vector field $\mathrm{ker}(\cdot, z)a$ belongs to $\mathcal{H}$ and $\langle \mathrm{ker}(\cdot, z)a, v_t\rangle_{\mathcal{H}} = a^\mathsf{T} v_t(z)$ for $v_t \in \mathcal{H}$. Using this construction, it is possible to reduce the minimization in (4) from an infinite-dimensional, nonparametric variational problem to a parametric one. This can be accomplished using an RKHS argument similar to the so called kernel trick in kernel methods [115,139,140]. Since the distance between the deformed shape $f_1 \cdot s_0$ and s_1 only depends on f at time $t = 1$ and we parameterize the template shape s_0 as a set of a few select points $\{x_0^i\}_{i=1}^{m_0}$ (see §2.3), it suffices to compute trajectories for these select points x^i, $i = 1, \dots, m_0$, that satisfy the flow equation (1), i.e., $\partial_t x^i = v_t(x^i)$. It follows that for an optimal velocity v we have that the squared norm $\|v_t\|_{\mathcal{H}}$ is minimal over all $\|u\|_{\mathcal{H}}^2$ with $u(x^i) = v_t(x^i)$ for all $t \in [0, 1]$. Therefore, we have that v_t is of the form [57,139]

$$v_t(z) = \sum_{i=1}^{m_0} \mathrm{ker}(x^i(t), z)\, a^i(t) \quad \text{for all } z \in \mathbb{R}^3$$

with $\mathrm{ker}: \mathbb{R}^3 \times \mathbb{R}^3 \to \mathbb{R}^{3,3}$ as defined above. With this, we reduced the search for $v \in L^2([0,1], \mathcal{H})$ to a set of time dependent, unknown coefficients $a(t) = (a^i(t))_{i=1}^{m_0}$ in $\mathbb{R}^3$.

Next, we specify more concretely the implementation details. We simplify the RKHS construction by modeling the kernel as a scalar function $\mathrm{ker}_{\sigma_v}: \mathbb{R}^3 \times \mathbb{R}^3 \to \mathbb{R}$ parameterized by $\sigma_v > 0$. The matrix representation introduced above is, equivalently, given by $\mathrm{ker}(\cdot,\cdot) = \mathrm{ker}_{\sigma_v}(\cdot,\cdot) I_3$, $I_3 := \mathrm{diag}(1,1,1) \in \mathbb{R}^{3,3}$. With this, we obtain

$$v_t(z) = \sum_{i=1}^{m_0} \mathrm{ker}_{\sigma_v}(x^i(t), z)\, a^i(t) \quad \text{for all } z \in \mathbb{R}^3, \tag{6}$$

where the coefficients $a(t) = (a^i(t))_{i=1}^{m_0} \in \mathbb{R}^{3m_0}$, $m_0 \in \mathbb{N}$, are associated with the state vector $x(t) = (x^i(t))_{i=1}^{m_0} \in \mathbb{R}^{3m_0}$. Another key feature of this construction is that we can specify the kernel ker instead of defining the space $\mathcal{H}$ [6,115].
The choice of the kernel ker_{σ_v} is crucial, since it uniquely determines the associated RKHS. We consider a Gaussian kernel defined as

$$\mathrm{ker}_{\sigma_v}(u, u') = \frac{1}{\left(\sqrt{2\pi}\sigma_v\right)^3} \exp\left(-\frac{1}{2}\|u - u'\|^2/\sigma_v^2\right) \tag{7}$$

for any $u, u' \in \mathbb{R}^3$. This kernel is symmetric, positive definite and yields a metric that is invariant under rotation and translation. (We note that other kernels have been considered [116].) The Hilbert space $\mathcal{H}_{\sigma_v}^3$ of smooth vector fields in $\mathbb{R}^3$ is defined as

$$\mathcal{H}_{\sigma_v}^3 := \overline{\left\{w : w(\cdot) = \textstyle\sum_{i=1}^{m_0} \mathrm{ker}_{\sigma_v}(x^i, \cdot)a^i,\ a^i \in \mathbb{R}^3,\ x^i \in \mathbb{R}^3,\ m \in \mathbb{N}\right\}}. \tag{8}$$

We note that the Sobolev embedding hypothesis stated earlier in this work is satisfied if we model the space $\mathcal{H}$ using the self-reproducing Hilbert space $\mathcal{H}_{\sigma_v}^3$ for any $q > 5/2$.

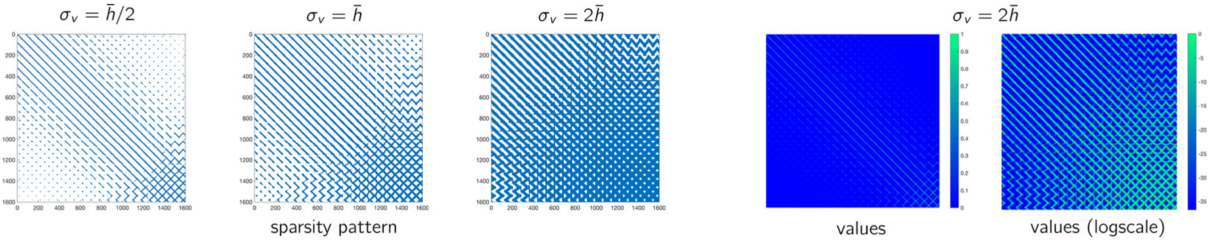

Fig. 2. Left block: Spy plot for the $m_0 \times m_0$ kernel matrix K in (14) for different choices of the bandwidth σ_v as a function of the mean edge length $\bar{h}$. The plots have been generated for an exemplary mitral valve shape. The matrices are of size 1600 × 1600. The mean edge length $\bar{h}$ is 1.2. We set $\sigma_v = s\bar{h}$ with $s \in \{1/4, 1/2, 1, 2, 4\}$. We show the sparsity pattern for $s = 1/2$ (left), $s = 1$ (middle) and $s = 2$ (right). The number of nonzero entries (more precisely, entries with a value below 1×10^{-16}) is 42 966 (1.68%; $s = 1/4$; not shown), 157 988 (6.17%; $s = 1/2$), and 523 740 (percentage: 20.46%; $s = 1$), 1 470 372 (57.44%; $s = 2$), and 2 559 418 (99.97%; $s = 4$; not shown), respectively. Right block: Visualization of the entries of the kernel matrix K for $\sigma_v = 2\bar{h}$ (left: values for the entries $(K_{ij})_{ij}$; right: logarithmic scale). (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

2.3.2. Discretized kinetic energy

The norm on $\mathcal{H}^3_{\sigma_v}$ in (8) is defined by the square root of

$$\|\sum_{l=1}^{m_0} \mathrm{ker}_{\sigma_v}(\cdot, x^l)\, a^l\|^2_{\mathcal{H}^3_{\sigma_v}} = \sum_{k=1}^{m_0}\sum_{l=1}^{m_0} \mathrm{ker}_{\sigma_v}(x^k, x^l)\, \langle a^k, a^l\rangle_{\mathbb{R}^3}, \tag{9}$$

where $\langle \cdot, \cdot \rangle_{\mathbb{R}^3}$ denotes the standard inner product in $\mathbb{R}^3$, and $a^l \in \mathbb{R}^3$ are coefficient vectors associated with the mesh points $x^l \in \mathbb{R}^3$. We use a left Riemann summation [141] to discretize the integration in time that appears in (2). With this, the discrete version of the kinetic energy in (2) is given by

$$\mathrm{kin}^h(a) := h \sum_{j=1}^{n} \mathrm{hilb}(t^j), \tag{10}$$

where $h = {}^1/_{(n+1)}$ denotes the size of the cells of the temporal mesh $[t^1, \dots, t^{n+1}] \in \mathbb{R}^{n+1}$ and

$$\mathrm{hilb}(t^j) := \sum_{k=1}^{m_0}\sum_{l=1}^{m_0} \mathrm{ker}_{\sigma_v}(x^k(t^j), x^l(t^j))\, \langle a^i(t^j), a^l(t^j)\rangle_{\mathbb{R}^3}.$$

Under the assumption that the controls a are ordered according to (5), we can rewrite (10) as

$$\mathrm{kin}^h(a) := h a^\mathsf{T} B a, \tag{11}$$

where B is a block-diagonal matrix of the form

$$B := B[x] = \mathrm{diag}\left(B^1[x], \dots, B^{n+1}[x]\right) \in \mathbb{R}^{3m_0(n+1), 3m_0(n+1)}, \quad B^j[x] = I_3 \otimes K[x(t^j)] \in \mathbb{R}^{3m_0, 3m_0}, \tag{12}$$

where $\otimes$ denotes the Kronecker product, i.e.,

$$A \otimes B = \begin{pmatrix} a_{11}B & \dots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \dots & a_{mn}B \end{pmatrix} \in \mathbb{R}^{mp,nq} \tag{13}$$

for any $A \in \mathbb{R}^{n,m}$, $B \in \mathbb{R}^{p,q}$, $I_3 = \mathrm{diag}(1, 1, 1) \in \mathbb{R}^{3,3}$, and $K[x(t^j)]$ are Gram matrices of the form

$$K[x(t^j)] = \left(K_{lk}(x(t^j))\right)_{l,k=1}^{m_0,m_0} \in \mathbb{R}^{m_0,m_0}, \quad K_{lk}(x(t^j)) := \mathrm{ker}_{\sigma_v}(x^l(t^j), x^k(t^j)) \in \mathbb{R}, \tag{14}$$

with $x^l(t^j) \in \mathbb{R}^3$ and $x^k(t^j) \in \mathbb{R}^3$ denoting the lth and kth entry of the state vector x at time point t^j and $\mathrm{ker}_{\sigma_v} : \mathbb{R}^3 \times \mathbb{R}^3 \to \mathbb{R}$ as in (7). We illustrate the matrix $B[x]$ for a particular shape x for different choices of the bandwidth σ_v in Fig. 2.

Remark 3. We note that the kernel matrices K in (14) that appear in the regularization operator B in (12) are functions of the state vector $x(t^j)$. This significantly complicates the optimization problem since we need to differentiate not only with respect to the control variable a but also the state variable x (and the dual variables) to derive the optimality conditions. To avoid this complication, we fix the kernel matrices that enter the regularizer in (11) by replacing $K[x(t^j)]$ by $K[x(t^0)] = K[x_0]$ for all $j = 1, \dots, n$. This approximation significantly simplifies the optimization problem: The regularization operator B does not change during optimization since it depends on the fixed (static) shape representation x_0 and not on the state variable $x(t^j)$ that is updated during the course of the optimization. We end up with a quadratic term that does not need to be differentiated with respect to the state vector x. We have seen that this approximation works well in practice, even if the shapes to be registered, i.e., the shapes parameterized by x_0 and x_1, differ significantly. (If the shapes are different, we expect the state vector x to change significantly. Consequently, the kernel entries in B are expected to change as we move from x_0 to x_1.) We note that the norm in (11) still depends on the dynamics of our problem, since the coefficients a in (11) change in time.

2.3.3. Discrete distance measures between shapes

Before we discuss our framework, we note that several geometric distance measures have been proposed, based on different shape representations. They include representations as measures [49,57] (see also below), currents [142], or varifolds [86,143–145]. We refer to these papers for more details.

Let $\mathcal{S}$ be the set of all compact, smooth 3D-surfaces properly embedded in $\mathbb{R}^3$ with piecewise smooth boundaries. Let $\mathcal{B}$ denote the space of bounded Borel measures on $\mathbb{R}^3$ endowed with the Hilbert norm $\|\cdot\|_{\mathcal{H}_\sigma}$ associated with the scalar inner product

$$\langle \mu, \tilde{\mu}\rangle_{\mathcal{H}_\sigma} := \int_{\mathbb{R}^3}\int_{\mathbb{R}^3} \operatorname{ker}_\sigma(x,\tilde{x})\,\mathrm{d}\mu(x)\,\mathrm{d}\tilde{\mu}(\tilde{x}) \tag{15}$$

for any $\mu, \tilde{\mu} \in \mathcal{B}$, where $\operatorname{ker}_\sigma : \mathbb{R}^3 \times \mathbb{R}^3 \to \mathbb{R}_{>0}$ denotes any smooth, symmetric, translation invariant and bounded positive definite kernel parameterized by $\sigma > 0$. (We are going to consider Gaussian kernels as defined in (7); hence the dependence on the parameter σ.) We can identify each submanifold s regularly embedded in $\mathbb{R}^3$ with Borel measure $\mu \in \mathcal{B}$ induced on s by the Lebesgue measure of $\mathbb{R}^3$. The corresponding squared distance between two bounded Borel subsets s and $\tilde{s}$ with measures μ and $\tilde{\mu}$, respectively, is then defined by

$$\operatorname{dist}_{\mathcal{H}_\sigma}(s,\tilde{s}) = \|\mu - \tilde{\mu}\|^2_{\mathcal{H}_\sigma}. \tag{16}$$

As we have outlined in §2.3, we identify these surfaces as a mesh of points $x_i \in \mathbb{R}^{3m_i}$, $i = 0, 1$, respectively. In this setting, it is natural to represent the shapes $s_i \in \mathcal{S}$ as bounded positive Radon measures on $\mathbb{R}^3$, namely as weighted sums of Dirac measures $\delta(x_i^k)$, $k = 1, \dots, m_i$, $i = 0, 1$, associated with the point sets $x_i \in \mathbb{R}^m$ [49,56,57]. In particular,

$$\mu_0 = \sum_{k=1}^{m_0} p_k\,\delta(x_0^k), \quad \mu_1 = \sum_{k'=1}^{m_1} q_{k'}\,\delta(x_1^{k'}), \tag{17}$$

with weights $p_k, q_{k'} \in \mathbb{R}$. With this, we arrive at the discrete representation of the distance functional in (4) given by

$$\operatorname{dist}^h(x_0, x_1) = \frac{\alpha}{2}\|\mu_0 - \mu_1\|^2_{\mathcal{H}_{\sigma_s}}. \tag{18}$$

As stated above, the set of bounded Borel measures is endowed with the Hilbert scalar product in (15). The associated squared Hilbert norm $\|\mu_0 - \mu_1\|^2_{\mathcal{H}_{\sigma_s}}$ in (18) represents the Borel distance between the shapes s_0 and s_1, associated with the Gaussian kernels $\operatorname{ker}_{\sigma_s}$ defined in (7) with an adequate choice for the parameter $\sigma_s > 0$. With the definition of the inner product in (15) and the distance in (16) we obtain

$$\|\mu_0 - \mu_1\|^2_{\mathcal{H}_{\sigma_s}} = \langle \mu_0 - \mu_1, \mu_0 - \mu_1\rangle_{\mathcal{H}_{\sigma_s}} = \langle \mu_0, \mu_0\rangle_{\mathcal{H}_{\sigma_s}} - 2\langle \mu_0, \mu_1\rangle_{\mathcal{H}_{\sigma_s}} + \langle \mu_1, \mu_1\rangle_{\mathcal{H}_{\sigma_s}},$$

where the inner products $\langle\cdot,\cdot\rangle_{\mathcal{H}_{\sigma_s}}$ are, based on the representation of μ_i, $i = 0, 1$, in (17), given by

$$\langle \mu_0, \mu_0\rangle_{\mathcal{H}_{\sigma_s}} = \int_{\mathbb{R}^3}\int_{\mathbb{R}^3} \operatorname{ker}_{\sigma_s}(x,x')\,\mathrm{d}\mu_0(x)\,\mathrm{d}\mu_1(x') = \sum_{k=1}^{m_0}\sum_{k'=1}^{m_0} p_k p_{k'} \operatorname{ker}_{\sigma_s}(x_0^k, x_0^{k'}), \tag{19a}$$

$$\langle \mu_0, \mu_1\rangle_{\mathcal{H}_{\sigma_s}} = \int_{\mathbb{R}^3}\int_{\mathbb{R}^3} \operatorname{ker}_{\sigma_s}(x,x')\,\mathrm{d}\mu_0(x)\,\mathrm{d}\mu_1(x') = \sum_{k=1}^{m_0}\sum_{k'=1}^{m_1} p_k q_{k'} \operatorname{ker}_{\sigma_s}(x_0^k, x_1^{k'}), \tag{19b}$$

$$\langle \mu_1, \mu_1\rangle_{\mathcal{H}_{\sigma_s}} = \int_{\mathbb{R}^3}\int_{\mathbb{R}^3} \operatorname{ker}_{\sigma_s}(x,x')\,\mathrm{d}\mu_1(x)\,\mathrm{d}\mu_1(x') = \sum_{k=1}^{m_1}\sum_{k'=1}^{m_1} q_k q_{k'} \operatorname{ker}_{\sigma_s}(x_1^k, x_1^{k'}), \tag{19c}$$

respectively.

The parameter $\alpha > 0$ in (18) is a weight that controls the contribution of the distance in (18) versus the contribution of the kinetic energy in (10). We discuss policies for selecting α in §2.4.4.

Remark 4. We note that $\sigma_v \neq \sigma_s$, in general. That is, σ_v is chosen to control the smoothness of the velocity v whereas σ_s controls the contribution of all points $x_j^l \in \mathbb{R}^3$ to the distance for a particular point $x_i^k \in \mathbb{R}^3$ with $i, j \in \{0, 1\}$ for all $k, l = 1, \dots, m$. We specify heuristic choices for the parameters σ_s and σ_v in §2.4.4.

Likewise to the kinetic energy in (11), we consider a matrix-vector representation to compactly represent the distance in (18). We have

$$\mathrm{dist}^h(x_0, x_1) = \frac{\alpha}{2}\left(\phi(x_0, x_0) - 2\phi(x_0, x_1) + \phi(x_1, x_1)\right), \tag{20}$$

with

$$\phi(x_i, x_j) = \frac{1}{m_i m_j}\sum_{k=1}^{m_i}\sum_{l=1}^{m_j} \mathrm{ker}_{\sigma_s}(x_i^k, x_j^l) = \frac{1}{m_i m_j} e_{m_i m_j}^{\mathsf{T}} \exp\left(-\frac{1}{2\sigma_s} S(d(x_i, x_j) \odot d(x_i, x_j))\right),$$

where $S := e_3^{\mathsf{T}} \otimes I_{m_i m_j}$, $S \in \mathbb{R}^{m_i m_j, 3 m_i m_j}$, $e_3 = (1, 1, 1) \in \mathbb{R}^3$, $I_{m_i m_j} := \mathrm{diag}(1, \dots, 1) \in \mathbb{R}^{m_i m_j, m_i m_j}$, $\exp : \mathbb{R}^{m_i m_j} \to \mathbb{R}^{m_i m_j}$ is the pointwise exponential, $\otimes$ is the Kronecker product (see (13)), $\odot$ is the Hadamard product defined as the elementwise/entrywise product of two matrices/vectors, i.e.,

$$(A \odot B)_{kl} = (A)_{kl}(B)_{kl}, \quad k, l = 1, \dots, n, \tag{21}$$

for any $A, B \in \mathbb{R}^{n,n}$, and all-to-all distance vector $d(x_i, x_j) := P_1 x_i - P_2 x_j \in \mathbb{R}^{3 m_i m_j}$ with permutation matrices $P_1 := I_3 \otimes I_{m_i} \otimes e_{m_j}$, $P_1 \in \mathbb{R}^{3 m_i m_j, 3 m_i}$, $P_2 := I_3 \otimes e_{m_i} \otimes I_{m_j}$, $P_2 \in \mathbb{R}^{3 m_i m_j, 3 m_j}$, respectively. The all-to-all distance vector d introduced above is a vector whose entries represent the difference between any point $\{x_i^l\}_{l=1}^{m_i}$ of x_i to any point $\{x_j^k\}_{k=1}^{m_j}$ of x_j with $i, j \in \{0, 1\}$.

Comparing the expression for ϕ in (20) to (19) reveals that we select uniform weights $p_k = 1/m_0$ and $q_{k'} = 1/m_1$ for all $k = 1, \dots, m_0$, $k' = 1, \dots, m_1$, respectively, for the weighted sums in (17). We use uniform weights for simplicity. Based on our past experience and the results reported in this work, this choice yields good performance in terms of convergence and shape matching accuracy. However, we note that it is certainly possible to use weights that are adaptive and depend on the parameterization of the shape. It remains subject to future work to explore if such an adaptive scheme is beneficial for the problems considered in our work. A simple modification we have implemented, that does not quite depend on the parameterization, is to increase the contribution of the points located on the boundary of the curved surface to the matching term (the boundaries of the mitral valve leaflets in our case; see Fig. 4 for an illustration). That is, we increase α in (20) for points located on the boundary. We observed in our past work [62] that this leads to a more accurate matching close to the boundary. We omit these details for simplicity of presentation.

Remark 5. The permutation matrices P_l, $l = 1, 2$, are introduced to provide a compact representation of the computation of the all-to-all difference between the points of x_i, x_j, $i, j = 0, 1$, across all spatial dimensions, respectively. We note that the Kronecker product notation is only used for presentation purposes. In computation, tensor operations are used to avoid explicit formation of these large, sparse matrices.

2.3.4. Numerical time integration

We use an explicit Euler method to integrate (1) forward in time [56,57]. Using the RKHS representation in (6) to model v_t we have

$$x^l(t^{j+1}) = x^l(t^j) + h\sum_{k=1}^{m_0} \mathrm{ker}_{\sigma_v}(x^k(t^j), x^l(t^j))\, a^k(t^j), \quad l = 1, \dots, m_0, \tag{22}$$

for the lth entry $x^l(t^{j+1}) \in \mathbb{R}^3$ of the state vector $x = (x(t^1), \dots, x(t^{n+1})) \in \mathbb{R}^{3 m_0 (n+1)}$ at time t^{j+1}, with uniform time step size $h = 1/(n+1)$. Using the lexicographical ordering in (5), we can represent (22) as

$$\begin{pmatrix} G^x & G^a \end{pmatrix} \begin{pmatrix} x \\ a \end{pmatrix} = q$$

with state-control vector $(x, a) \in \mathbb{R}^{6 m_0 (n+1)}$. The matrices G^x and $G^a := G^a[x]$ are given by the lower block-bidiagonal and lower block-diagonal matrices

$$G^a[x] = -h \begin{pmatrix} 0 & \cdots & 0 \\ B^1[x] & \ddots & \vdots \\ \vdots & \ddots & 0 \\ 0 & \cdots & B^{n+1}[x] \end{pmatrix} \in \mathbb{R}^{3m_0(n+1),3m_0(n+1)} \quad \text{and}$$
$$G^x = \begin{pmatrix} I_{3m_0} & \cdots & 0 \\ -I_{3m_0} & \ddots & \vdots \\ \vdots & \ddots & I_{3m_0} \\ 0 & \cdots & -I_{3m_0} \end{pmatrix} \in \mathbb{R}^{3m_0(n+1),3m_0(n+1)}, \tag{23}$$

$I_{3m_0} = \text{diag}(1, \dots, 1) \in \mathbb{R}^{3m_0,3m_0}$, respectively. The right hand side $q \in \mathbb{R}^{3m_0(n+1)}$ is given by

$$q = \begin{pmatrix} x_0 \\ 0 \end{pmatrix} \tag{24}$$

and $B^j[x]$ is as defined in (12).

Remark 6. Notice that the matrix G^x is a sparse matrix whereas the lower block-diagonal entries of G^a are, in general, dense (as a function of σ_v; see Fig. 2). In our implementation, the matrices G^x and G^a (and the associated adjoint operators) are not assembled and/or stored; we implement matrix-vector products using the block matrices, instead. However, we assemble and store the matrices $K[x(t^j)]$ which imposes memory restrictions on our current implementation.

Remark 7. The explicit Euler scheme considered in this work is only first order accurate in time. Our methodology is modular in the sense that we can replace this scheme with any other (high-order) numerical time integration scheme (at the expense of computational complexity and runtime). In past work on a different problem formulation we observed that using higher-order time integration schemes for computing the diffeomorphic flow can be beneficial [96]. In particular, we observed that a fourth order Runge–Kutta method for numerical time integration is more expressive when it comes to modeling complex diffeomorphic flows [96]. In other past work, we consider second order accurate time integration schemes [15,95,97,98]. We did not observe issues in the present application with our first order scheme. Exploring high-order schemes for numerical time integration remains subject to future work.

2.3.5. *Discretized optimal control problem*

If we put all building blocks described above together, we arrive the following discrete version of (4):

$$\begin{aligned} &\underset{a,\,x}{\text{minimize}} \;\; \text{dist}^h(Qx, x_1) + \text{kin}^h(a) \\ &\text{subject to} \;\; \begin{pmatrix} G^x & G^a \end{pmatrix} \begin{pmatrix} x \\ a \end{pmatrix} = q. \end{aligned} \tag{25}$$

Here, $(x, a) \in \mathbb{R}^{6m_0(n+1)}$ denotes the entire state-control trajectory, with x and a as in (5), discretized kinetic energy kin^h as in (10), discretized distance dist^h as in (20), matrix-operators G^a and G^x as in (23), and right-hand side q as in (24), respectively. Notice that we apply an observation operator Q to the state variable x, such that we compute the proximity between the solution of the state equation (equality constraint in (25)), i.e., the deformed shape at time t^{n+1} for a given control a, to the reference shape x_1. More precisely, $x(t^{n+1}) = Qx$, $Q = (0_{3m_0} \dots 0_{3m_0} \, I_{3m_0}) \in \mathbb{R}^{3m_0,3m_0(n+1)}$ with zero matrices $0_{3m_0} = \text{diag}(0, \dots, 0) \in \mathbb{R}^{3m_0,3m_0}$ and identity matrix $I_{3m_0} \in \mathbb{R}^{3m_0,3m_0}$, respectively.

2.4. *Numerical optimization*

In what follows, we describe different methods to solve (25) numerically.

2.4.1. *Adjoint-based line search methods*

In our past work, our group has developed several adjoint based first and second order numerical optimization algorithms [57–60]. These approaches can be summarized by the iterative scheme [70,146]

$$a^{(k+1)} = a^{(k)} - t^{(k)} s^{(k)}, \quad k = 1, 2, \dots, \tag{26}$$

where $k \in \mathbb{N}$ denotes the iteration index, $a^{(k)} \in \mathbb{R}^{3m_0}$ is the iterate, and $s^{(k)} := -B^{(k)} g^{(k)}$ represents the search direction with $B^{(k)} \in \mathbb{R}^{3m_0,3m_0}$, $m_0 \in \mathbb{N}$, reduced gradient $g^{a,(k)} \in \mathbb{R}^{3m_0}$ (with respect to the control variable a), and line search parameter $t^{(k)} > 0$. We note that we consider a reduced space method in (26). That is, we only iterate on the reduced space of the control variable a. In *full space* or *all-at-once methods*, all unknowns of the problem (i.e., the state variable(s), the adjoint (or dual) variable(s), and the control variable(s)) are updated simultaneously. This affects the structure and size of the vectors and matrices involved in the computations associated with (26), respectively. We refer, e.g., to [147–152] for more details. The choice of $B^{(k)}$ determines the type of algorithm [70,146]. If we set $B^{(k)}$ to $I_{3m_0} := \text{diag}(1, \dots, 1) \in \mathbb{R}^{3m_0,3m_0}$, we obtain a gradient descent algorithm. If we select $B^{(k)} = (H^{(k)})^{-1}$, where $H^{(k)} \in \mathbb{R}^{3m_0,3m_0}$ is the Hessian matrix, we arrive at a Newton algorithm. Other options for the matrix $B^{(k)}$ include approximations to the Hessian, resulting in different flavors of quasi-Newton methods. These methods vary in their rate of convergence, ranging from linear (gradient descent), to super-linear (quasi-Newton methods), to quadratic (Newton method) [70,146]. In the context of LDDMM (and related formulations), quasi–Newton algorithms have been considered in [15,16,98,112] (specifically, adjoint-based Gauss–Newton) and [105,118] (BFGS; adjoint-based or automatic differentiation), respectively.

We omit additional details for the sake of brevity and refer to our past work [57–60]. We provide bits and pieces of the building blocks for these algorithms in the following sections, including derivative information of the variational optimization problem (25). We note that we demonstrated in [62] that our (original) implementation of the splitting approach discussed below outperforms the Newton algorithm developed in our prior work [60] in terms of runtime. Our improved implementation presented in this work is about 2× faster and requires less memory, allowing for the solution of bigger problems than those considered in [62] due to a matrix-free implementation.

2.4.2. Operator splitting

In [62], we propose the use of an operator splitting strategy to solve the nonlinear control problem in (4). The considered approach is typically referred to as *alternating direction method of multipliers* (**ADMM**) [153–155]. It is equivalent to the *Douglas–Rachford splitting method* [87–92], which was initially developed to numerically solve PDEs [156]. In its modern form, this algorithm was introduced by Glowinski and Marrocco [93] and Gabay and Mercier [94]. Recent work of Glowinski and colleagues in the area of operator splitting includes [90,157–161]. We refer to [90] for a general exposition of operator splitting techniques and to [87,153,155,162] for a lucid discussion of operator splitting approaches in the context of optimization.

Problem formulation To concisely present the splitting algorithm developed in [62], we introduce the set

$$C := \left\{ \begin{pmatrix} x \\ a \end{pmatrix} \in \mathbb{R}^{6m_0(n+1)} : \begin{bmatrix} G^x & G^a \end{bmatrix} \begin{pmatrix} x \\ a \end{pmatrix} = q \right\} \tag{27}$$

of state-control pairs $(x, a) \in \mathbb{R}^{6m_0(n+1)}$ that satisfy the dynamical system (constraint) in (25). Moreover, let $\text{ind}_C : \mathbb{R}^{3m_0(n+1)} \times \mathbb{R}^{3m_0(n+1)} \to \{0, \infty\}$ denote the indicator function for the set C in (27), i.e.,

$$\text{ind}_C(x, a) := \begin{cases} 0 & \text{if } (x, a) \in C, \\ \infty & \text{otherwise.} \end{cases}$$

In addition, with slight abuse of notation, we define

$$\text{dist}^h(x, a) := \text{dist}^h(Qx, x_1) \quad \text{and} \quad \text{kin}^h(x, a) := \text{kin}^h(a)$$

with discretized distance dist^h and discretized kinetic energy kin^h as in (25), respectively. Using this notation allows us to better illustrate the derivation of the splitting strategy. We can rewrite (25) to obtain

$$\begin{aligned} &\underset{a,x}{\text{minimize}} \;\; \text{dist}^h(x, a) + \text{kin}^h(x, a) \\ &\text{subject to} \;\; \begin{pmatrix} G^x & G^a \end{pmatrix} \begin{pmatrix} x \\ a \end{pmatrix} = q \end{aligned} \tag{28}$$

with variables $(x, a) \in \mathbb{R}^{6m_0(n+1)}$. With the indicator function ind_C introduced above, we can represent the equality constrained optimization problem in (25) as an unconstrained optimization problem of the form

$$\underset{x,a}{\text{minimize}} \;\; \text{ind}_C(x, a) + \text{kin}^h(x, a) + \text{dist}^h(x, a). \tag{29}$$

To derive our splitting approach, we consider the *consensus form* [153,162] of (29). The fundamental idea of the consensus form of operator splitting is to solve a coupled, multi-objective optimization problem (that is difficult to solve in its original form) by splitting it into multiple (easier) subproblems to be solved separately. These subproblems are then tied together by a constraint that enforces convergence to a single, consensus solution at the end of optimization. For the particular case of (29), we split the problem into two separate parts by introducing a second pair of state-control variables $(\tilde{x}, \tilde{a}) \in \mathbb{R}^{6m_0(n+1)}$. To ensure convergence to one solution, we introduce an equality constraint that enforces the state-control pairs (x, a) and $(\tilde{x}, \tilde{a})$ to be equal—the *consensus* or *consistency constraint*. We obtain

$$
\begin{aligned}
&\underset{x,a}{\text{minimize}} \quad \Big(\operatorname{ind}_C(x,a) + \operatorname{kin}^h(x,a)\Big) + \operatorname{dist}^h(\tilde{x},\tilde{a})\\
&\text{subject to} \quad \begin{pmatrix} x\\ a\end{pmatrix} = \begin{pmatrix}\tilde{x}\\ \tilde{a}\end{pmatrix}.
\end{aligned}
\tag{30}
$$

As indicated in (30), we have split the objective in (28) into two separate parts, with different variables. The first part contains the indicator function for the set C (i.e., the model for the discretized ODE constraint) and the kinetic energy. The second part encodes the mismatch term. The equality constraint states that these terms are "in consensus." Using this formulation, we can derive an iterative scheme as follows. We use the method of Lagrange multipliers to solve (30). We introduce the dual variables $(\nu,\omega) \in \mathbb{R}^{6m_0(n+1)}$ for the equality constraint $(x,a)-(\tilde{x},\tilde{a}) = 0$ in (30). With this, the augmented Lagrangian for the problem in (30) is given by

$$
\mathcal{L}_\rho(x,a,\tilde{x},\tilde{a},\nu,\omega) = \Big(\operatorname{ind}_C(x,a) + \operatorname{kin}^h(x,a)\Big) + \operatorname{dist}^h(\tilde{x},\tilde{a}) + \begin{pmatrix}\nu\\ \omega\end{pmatrix}^{\mathsf{T}}\left(\begin{pmatrix}x\\ a\end{pmatrix} - \begin{pmatrix}\tilde{x}\\ \tilde{a}\end{pmatrix}\right) + \frac{\rho}{2}\left\|\begin{pmatrix}\tilde{x}\\ \tilde{a}\end{pmatrix} - \begin{pmatrix}x\\ a\end{pmatrix}\right\|_2^2,
$$

where $\rho > 0$ is a parameter. Let $(\tilde{x}^{(1)},\tilde{a}^{(1)}) \in \mathbb{R}^{6m_0(n+1)}$ and $(\nu^{(1)},\omega^{(1)}) \in \mathbb{R}^{6m_0(n+1)}$ denote our initial guess for the state-control pair $(\tilde{x},\tilde{a}) \in \mathbb{R}^{6m_0(n+1)}$ and the associated dual variables $(\nu,\omega) \in \mathbb{R}^{6m_0(n+1)}$ at iteration $k=1$, respectively. We can now minimize the Lagrangian $\mathcal{L}_\rho$ over each individual pair of primal variables, while the value for the other pair of primal variables as well as the dual variables remain fixed. We obtain

$$
\begin{aligned}
\begin{pmatrix}x^{(k+1)}\\ a^{(k+1)}\end{pmatrix} &= \underset{x,a}{\operatorname{argmin}}\, \mathcal{L}_\rho(x,a,\tilde{x}^{(k)},\tilde{a}^{(k)},\nu^{(k)},\omega^{(k)})\\
\begin{pmatrix}\tilde{x}^{(k+1)}\\ \tilde{a}^{(k+1)}\end{pmatrix} &= \underset{\tilde{x},\tilde{a}}{\operatorname{argmin}}\, \mathcal{L}_\rho(x^{(k+1)},a^{(k+1)},\tilde{x},\tilde{a},\nu^{(k)},\omega^{(k)})\\
\begin{pmatrix}\nu^{(k+1)}\\ \omega^{(k+1)}\end{pmatrix} &= \begin{pmatrix}\nu^{(k)}\\ \omega^{(k)}\end{pmatrix} + \rho\left(\begin{pmatrix}\tilde{x}^{(k+1)}\\ \tilde{a}^{(k+1)}\end{pmatrix} - \begin{pmatrix}x^{(k+1)}\\ a^{(k+1)}\end{pmatrix}\right).
\end{aligned}
$$

We can observe that the dual variables of this problem correspond to a scaled running sum of the consensus error [153]. We combine the linear and quadratic terms in the augmented Lagrangian to derive the proximal version of the scheme stated above. Dropping constant terms from the augmented Lagrangian $\mathcal{L}_\rho$ in each step, we obtain

$$
\begin{aligned}
\begin{pmatrix}x^{(k+1)}\\ a^{(k+1)}\end{pmatrix} &= \underset{x,a}{\operatorname{argmin}}\left(\operatorname{ind}_C(x,a) + \operatorname{kin}^h(x,a) + \begin{pmatrix}\nu^{(k)}\\ \omega^{(k)}\end{pmatrix}^{\mathsf{T}}\begin{pmatrix}x\\ a\end{pmatrix} + \frac{\rho}{2}\left\|\begin{pmatrix}\tilde{x}^{(k)}\\ \tilde{a}^{(k)}\end{pmatrix} - \begin{pmatrix}x\\ a\end{pmatrix}\right\|_2^2\right),\\
\begin{pmatrix}\tilde{x}^{(k+1)}\\ \tilde{a}^{(k+1)}\end{pmatrix} &= \underset{\tilde{x},\tilde{a}}{\operatorname{argmin}}\left(\operatorname{dist}^h(\tilde{x},\tilde{a}) - \begin{pmatrix}\nu^{(k)}\\ \omega^{(k)}\end{pmatrix}^{\mathsf{T}}\begin{pmatrix}\tilde{x}\\ \tilde{a}\end{pmatrix} + \frac{\rho}{2}\left\|\begin{pmatrix}\tilde{x}\\ \tilde{a}\end{pmatrix} - \begin{pmatrix}x^{(k+1)}\\ a^{(k+1)}\end{pmatrix}\right\|_2^2\right),\\
\begin{pmatrix}\nu^{(k+1)}\\ \omega^{(k+1)}\end{pmatrix} &= \begin{pmatrix}\nu^{(k)}\\ \omega^{(k)}\end{pmatrix} + \rho\left(\begin{pmatrix}\tilde{x}^{(k+1)}\\ \tilde{a}^{(k+1)}\end{pmatrix} - \begin{pmatrix}x^{(k+1)}\\ a^{(k+1)}\end{pmatrix}\right).
\end{aligned}
$$

We rescale the dual variables to obtain $(u^{(k)}, w^{(k)}) := ((1/\rho)\nu^{(k)}, (1/\rho)\omega^{(k)})$, and pull the linear terms into the quadratic terms. Then, the iterative scheme for successively updating the primal and dual variables (x,a), $(\tilde{x},\tilde{a})$, (u,w), is given by

$$
\begin{pmatrix}x^{(k+1)}\\ a^{(k+1)}\end{pmatrix} = \underset{x,a}{\operatorname{argmin}}\left(\operatorname{ind}_C(x,a) + \operatorname{kin}^h(x,a) + \frac{\rho}{2}\left\|\begin{pmatrix}x\\ a\end{pmatrix} - \begin{pmatrix}\tilde{x}^{(k)}\\ \tilde{a}^{(k)}\end{pmatrix} - \begin{pmatrix}u^{(k)}\\ w^{(k)}\end{pmatrix}\right\|_2^2\right), \tag{31a}
$$

$$
\begin{pmatrix}\tilde{x}^{(k+1)}\\ \tilde{a}^{(k+1)}\end{pmatrix} = \underset{\tilde{x},\tilde{a}}{\operatorname{argmin}}\left(\operatorname{dist}^h(\tilde{x},\tilde{a}) + \frac{\rho}{2}\left\|\begin{pmatrix}\tilde{x}\\ \tilde{a}\end{pmatrix} - \begin{pmatrix}x^{(k+1)}\\ a^{(k+1)}\end{pmatrix} + \begin{pmatrix}u^{(k)}\\ w^{(k)}\end{pmatrix}\right\|_2^2\right), \tag{31b}
$$

$$
\begin{pmatrix}u^{(k+1)}\\ w^{(k+1)}\end{pmatrix} = \begin{pmatrix}u^{(k)}\\ w^{(k)}\end{pmatrix} + \begin{pmatrix}\tilde{x}^{(k+1)}\\ \tilde{a}^{(k+1)}\end{pmatrix} - \begin{pmatrix}x^{(k+1)}\\ a^{(k+1)}\end{pmatrix}, \tag{31c}
$$

with iteration index $k = 1, 2, \ldots$. We can represent (31) more concisely in its proximal form as

$$
\begin{aligned}
\begin{pmatrix}x^{(k+1)}\\ a^{(k+1)}\end{pmatrix} &= \operatorname{prox}_{\gamma(\operatorname{ind}_C + \operatorname{kin}^h)}\left(\tilde{x}^{(k)} + u^{(k)}, \tilde{a}^{(k)} + w^{(k)}\right),\\
\begin{pmatrix}\tilde{x}^{(k+1)}\\ \tilde{a}^{(k+1)}\end{pmatrix} &= \operatorname{prox}_{\gamma \operatorname{dist}^h}\left(x^{(k+1)} - u^{(k)}, a^{(k+1)} - w^{(k)}\right),\\
\begin{pmatrix}u^{(k+1)}\\ w^{(k+1)}\end{pmatrix} &= \begin{pmatrix}u^{(k)}\\ w^{(k)}\end{pmatrix} + \begin{pmatrix}\tilde{x}^{(k+1)}\\ \tilde{a}^{(k+1)}\end{pmatrix} - \begin{pmatrix}x^{(k+1)}\\ a^{(k+1)}\end{pmatrix},
\end{aligned}
$$

where $\gamma := 1/\rho$. Compared to the optimization problem in (29), we now have to tackle two optimization problems. We see below that they are significantly simpler than the original, coupled problem in (29). The first optimization problem for (x, a) in (31a) solely involves the kinetic energy kin^h in (10) (along with the indicator function for the set of state-control pairs C in (27)). We refer to this subproblem as the *kinetic energy subproblem*. The second optimization problem for $(\tilde{x}, \tilde{a})$ in (31b) solely involves the kernel distance dist^h in (20); it minimizes the distance between the deformed shape $x(t^{n+1}) = Qx$ (i.e., the solution of the state equation for a candidate a) and the reference shape x_1. We refer to this optimization problem as the *distance subproblem*. The update in (31c) involves the residual of the primal variables (x, a) and $(\tilde{x}, \tilde{a})$ at iteration $k + 1$. As we converge to the solution of the original problem, (x, a) and $(\tilde{x}, \tilde{a})$ converges to the same vectors and, consequently, we expect the dual variables (u, w) to vanish at optimality. Evaluating (31c) is prosaic, but solutions to (31a) and (31b) are nontrivial. We discuss our approaches for each of these subproblems next.

Kinetic energy subproblem At each iteration of our splitting algorithm, we solve (31a) by setting up and solving the linear system associated with the first order KKT conditions. To this end, we form the Lagrangian for the quadratic, equality constrained optimization problem associated with the implicit, unconstrained formulation in (31a): Given current estimates $(u^{(k)}, w^{(k)})$ and $(\tilde{x}^{(k)}, \tilde{a}^{(k)})$ find (x, a) such that

$$\begin{aligned} &\underset{x,\,a}{\text{minimize}} \quad h a^\mathsf{T} B a + \frac{\rho}{2} \left\| \begin{pmatrix} x \\ a \end{pmatrix} - \begin{pmatrix} \tilde{x}^{(k)} \\ \tilde{a}^{(k)} \end{pmatrix} - \begin{pmatrix} u^{(k)} \\ w^{(k)} \end{pmatrix} \right\|_2^2 \\ &\text{subject to} \quad \begin{pmatrix} G^x & G^a \end{pmatrix} \begin{pmatrix} x \\ a \end{pmatrix} = q. \end{aligned} \tag{32}$$

As stated above, we use the method of Lagrange multipliers to handle the equality constraint. We have

$$\mathcal{L}(x, a, \nu) = h a^\mathsf{T} B a + \frac{\rho}{2} \left\| \begin{pmatrix} x \\ a \end{pmatrix} - \begin{pmatrix} \tilde{x}^{(k)} \\ \tilde{a}^{(k)} \end{pmatrix} - \begin{pmatrix} u^{(k)} \\ w^{(k)} \end{pmatrix} \right\|_2^2 - \nu^\mathsf{T}(G^a a + G^x x - q),$$

where $\nu \in \mathbb{R}^{\tilde{n}}$, $\tilde{n} := 3m_0(n+1)$ represents the vector of Lagrangian multipliers. Differentiating the Lagrangian with respect to each block of variables, we can derive the first order KKT optimality conditions for (32). These state that the optimal variables $(a^{\text{opt}}, x^{\text{opt}}, \nu^{\text{opt}})$ satisfies the block linear system

$$\begin{pmatrix} B + \rho I_{\tilde{n}} & 0 & -(G^a)^\mathsf{T} \\ 0 & \rho I_{\tilde{n}} & -(G^x)^\mathsf{T} \\ G^a & G^x & 0 \end{pmatrix} \begin{pmatrix} a^{\text{opt}} \\ x^{\text{opt}} \\ \nu^{\text{opt}} \end{pmatrix} = \begin{pmatrix} \tilde{a}^{(k)} + u^{(k)} \\ \tilde{x}^{(k)} + w^{(k)} \\ q \end{pmatrix}.$$

To solve this KKT system numerically, we perform the change of variables $(a^{\text{opt}}, x^{\text{opt}}) = (a + p^a, x + p^x)$. We obtain the linear system

$$\begin{pmatrix} B + \rho I_{\tilde{n}} & 0 & (G^a)^\mathsf{T} \\ 0 & \rho I_{\tilde{n}} & (G^x)^\mathsf{T} \\ G^a & G^x & 0 \end{pmatrix} \begin{pmatrix} -p^a \\ -p^x \\ \nu^{\text{opt}} \end{pmatrix} = \begin{pmatrix} g^a \\ g^x \\ c \end{pmatrix} \tag{33}$$

with $g^a = Ba + \rho(a - \tilde{a}^{(k)} - u^{(k)})$, $g^a \in \mathbb{R}^{\tilde{n}}$, $g^x = \rho(x - \tilde{x}^{(k)} - \nu^{(k)})$, $g^x \in \mathbb{R}^{\tilde{n}}$, and $c = G^a a + G^x x - q$, $c \in \mathbb{R}^{\tilde{n}}$. In exact arithmetic, the solution to this linear system gives the step (p^a, p^x) that sums with our current approximation to the solution (a, x) to get the solution $(a^{\text{opt}}, x^{\text{opt}})$ that satisfies the first order optimality conditions. Here, the vectors g^a and g^x correspond to the gradient of (32) with respect to a and x, respectively, and c corresponds to the linear constraint. This system is (in exact arithmetic) symmetric positive semi-definite. (See Fig. 3.)

There are several options to solve the linear system (33). The matrix on the left hand side of (33) is highly structured and sparse. Additionally, for the fixed kernel matrices we use, this matrix does not change during the optimization. These traits make direct factorization approaches [68,69] feasible for smaller problems; we considered a symmetric indefinite factorization [70, p. 454f.] (i.e., an LDLT factorization [163,164]) in our prior work [62]. Since we fix the block matrices, this factorization needs to be computed only once (at the beginning of the algorithm). This trick provides significant computational gains. On the downside, the factorization is computational demanding for fine discretizations and quickly results in excessive memory pressure since we need to form and store (33) and its factorization.

Another option is the Schur–complement method—a projection based approach [70, p.]. This method projects the problem onto the space of Lagrangian multipliers, resulting in the linear system

$$G M^{-1} G^\mathsf{T} \nu = G M^{-1} g - c \tag{34}$$

with

$$G := \begin{pmatrix} G^a & G^x \end{pmatrix} \in \mathbb{R}^{\tilde{n}, 2\tilde{n}}, \qquad M := \begin{pmatrix} B + \rho I_{\tilde{n}} & 0 \\ 0 & \rho I_{\tilde{n}} \end{pmatrix} \in \mathbb{R}^{2\tilde{n}, 2\tilde{n}}, \qquad g := \begin{pmatrix} g^a \\ g^x \end{pmatrix} \in \mathbb{R}^{2\tilde{n}}.$$

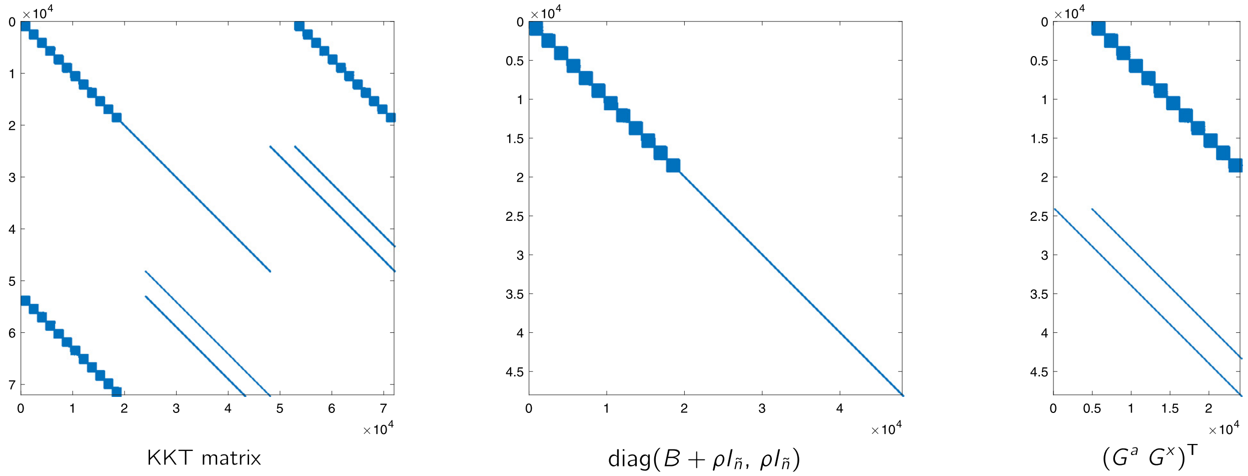

Fig. 3. Illustration of the KKT matrix in (33). We show the matrix on the left and the individual blocks on the middle and right. The matrices are constructed for two representative shapes from our mitral valve databases with $m_0 = m_1 = 1\,600$.

We solve this equation iteratively using a preconditioned conjugate gradient method (**PCG**). After solving for ν, the step updates are recovered by evaluating

$$\begin{pmatrix} p^a \\ p^x \end{pmatrix} = M^{-1}\left(G^\mathsf{T}\nu - g\right).$$

Remark 8. We make several remarks about the Schur–complement method we use here. First, we reiterate that the block matrices used above do not need to be computed explicitly but can be accessed via function calls to matrix-vector multiplication with a vector. This extends to M and its inverse, both of which are block diagonal. We compute matrix-vector products with the inverse of this matrix in a block-wise fashion. This requires the inversion of the $2(n+1)$ diagonal blocks. The first $n+1$ of these blocks are of the form $hB^j + \rho I_{3m_0}$ for $j = 1, \dots, n+1$ with B^j as in (12). These matrices are symmetric positive definite, and for small problems their Cholesky factors can be used to compute and store their inverses across all operator splitting iterations. For larger problems, explicit inversion may prove infeasible. This is a limitation of the current implementation, and the inversion of these blocks for larger problems represents future work. The remaining $n+1$ blocks are identity matrices scaled by the factor ρ, so their inversion is trivial.

We also note that solving (34) to a low tolerance using an iterative method results in steps (p^a, p^x) that do not yield the exact solution $(a^{\text{opt}}, x^{\text{opt}})$ satisfying the optimality conditions. In practice, the solution only represent an update resulting in a new iterate, $(a^{(k+1)}, x^{(k+1)}) = (a^{(k)} + p^a, x^{(k)} + p^x)$. Balancing accuracy and computational load, we set the tolerance for the PCG to 1×10^{-4} with an upper bound of 100 iterations.

Lastly, we note that our (modified) PCG implementation allows us to deal with optimization problems that tackle systems with matrices that are not positive definite. That is, we have added a condition that checks for negative curvature. We refer to, e.g., [70, p. 168f.] for details.

Preconditioner for reduced space KKT system We have designed a block preconditioner for (34). First, we note that

$$GM^{-1}G^\mathsf{T} = G^a(B + \rho I_{\tilde n})^{-1}(G^a)^\mathsf{T} + \rho^{-1}G^x(G^x)^\mathsf{T}.$$

Using block matrix operations, we then see that the matrix has block tridiagonal form with

$$GM^{-1}G^\mathsf{T} = \frac{1}{\rho}\begin{pmatrix} I_{3m_0} & -I_{3m_0} & & & & \\ -I_{3m_0} & L^1 & -I_{3m_0} & & & \\ & -I_{3m_0} & L^2 & -I_{3m_0} & & \\ & & \ddots & \ddots & \ddots & \\ & & & -I_{3m_0} & L^{n-1} & -I_{3m_0} \\ & & & & -I_{3m_0} & L^n \end{pmatrix} \in \mathbb{R}^{3m_0(n+1),3m_0(n+1)},$$

where $L^j = h^2 B^j(B^j + \rho I_{3m_0})^{-1}(B^j)^\mathsf{T} + 2\rho I_{3m_0}$ for $j = 1, \dots, n$. For the preconditioner, we drop the off diagonal identity blocks and invert to get

$$P^{-1} = \operatorname{diag}\left(I_{3m_0}, L^1, \dots, L^n\right)^{-1} \in \mathbb{R}^{3m_0(n+1),3m_0(n+1)}$$

as a left preconditioner. We note that P^{-1} is block diagonal, so the computation of the inverse requires only the inverses of the component blocks. These can be computed offline using a Cholesky factorization and reused throughout the optimization. Furthermore, the matrix does not need to be computed explicitly and can be accessed as a function call to a matrix-vector product.

Distance subproblem Next, we develop a numerical method to solve subproblem (31b). The associated unconstrained optimization problem is as follows: Given current estimates $(x^{(k+1)}, a^{(k+1)})$, $(u^{(k)}, w^{(k)})$ find $(\tilde{x}, \tilde{a})$ such that

$$\underset{\tilde{x},\tilde{a}}{\text{minimize}} \;\; \text{dist}^h(\tilde{x}, \tilde{a}) + \frac{\rho}{2} \left\| \begin{pmatrix} \tilde{x} \\ \tilde{a} \end{pmatrix} - \begin{pmatrix} x^{(k+1)} \\ a^{(k+1)} \end{pmatrix} + \begin{pmatrix} u^{(k)} \\ w^{(k)} \end{pmatrix} \right\|_2^2 . \tag{35}$$

Notice that the only variable appearing in the distance functional is $x^{n+1} = x(t^{n+1}) \in \mathbb{R}^{3m_0}$, i.e., the "transported" shape/state variable at the terminal time point $t^{n+1} = 1$. The remaining variables are included only in the mismatch term involving a 2-norm. Consequently, we can compute the minimizer for these variables explicitly as

$$\tilde{a}^{(k+1)} = a^{(k+1)} - w^{(k)} \quad \text{and} \quad \tilde{x}^{(k+1)}(t^j) = x^{(k+1)}(t^j) - u^{(k)}(t^j) \quad \text{for } j = 1, \dots, n.$$

The remaining problem for $\tilde{x}$ at time t^{n+1} with $z := \tilde{x}^{n+1}$ is

$$\underset{z}{\text{minimize}} \;\; \text{dist}^h(z, x_1) + \frac{\rho}{2} \| z - x^{(k+1)}(t^{n+1}) + v^{(k)}(t^{n+1}) \|_2^2 \tag{36}$$

We solve the unconstrained optimization problem in (36) at each outer iteration using a matrix-free Newton–Krylov method [70]. This requires first order (gradient) and second order (curvature) derivative information. The derivatives of the squared-norm are straightforward to obtain and therefore omitted. The first and second derivative of the distance as defined in (20) with respect to an arbitrary vector $z \in \mathbb{R}^{3m_0}$ is formally given by

$$\mathrm{d}_z \text{dist}^h(z, x_1) = \frac{\alpha}{2} \mathrm{d}_z \left(\phi(z, z) - 2\phi(z, x_1) + \phi(x_1, x_1) \right) = \frac{\alpha}{2} \left(\mathrm{d}_z \phi(z, z) - 2\mathrm{d}_z \phi(z, x_1) \right)$$

and

$$\mathrm{d}_{zz} \text{dist}^h(z, x_1) = \frac{\alpha}{2} \left(\mathrm{d}_{zz} \phi(z, z) - 2\mathrm{d}_{zz} \phi(z, x_1) \right),$$

respectively. Starting with the second term of the sum, the first derivative of ϕ with respect to the vector $z \in \mathbb{R}^{3m_0}$ is given by

$$\mathrm{d}_z \phi(z, x_1) = -\frac{1}{m_0 m_1 \sigma_s^2} P_1^\mathsf{T} D^\mathsf{T} S^\mathsf{T} \exp\left(-\frac{1}{2\sigma_s^2} S(d(z, x_1) \odot d(z, x_1)) \right) \in \mathbb{R}^{3m_0},$$

where $D := \text{diag}(d(z, x_1))$ denotes a diagonal matrix with the entries of the vector $d(z, x_1)$ on the main diagonal. The second derivative is given by the $3m_0 \times 3m_0$ matrix

$$\begin{aligned} \mathrm{d}_{zz} \phi(z, x_1) = {} & \frac{1}{m_0 m_1 \sigma_s^4} P_1^\mathsf{T} D^\mathsf{T} S^\mathsf{T} \text{diag}\left(\exp\left(-\frac{1}{2\sigma_s^2} S\left(d(z, x_1) \odot d(z, x_1) \right) \right) \right) S D P_1 \\ & - \frac{1}{m_0 m_1 \sigma_s^2} P_1^\mathsf{T} \text{diag}\left(S^\mathsf{T} \exp\left(-\frac{1}{2\sigma_s^2} S(d(z, x_1) \odot d(z, x_1)) \right) \right) P_1, \end{aligned}$$

The derivatives for $\phi(z, z)$ are computed similarly with the exception that the all-to-all distance vector d (see §2.3.3 for the definition), becomes $d(z, z) = P_1 z - P_2 z = (P_1 - P_2) z$. Thus, the first and second derivative for $\phi(z, z)$ are given by the vector

$$\mathrm{d}_z \phi(z, z) = -\frac{1}{m_0^2 \sigma_s^2} (P_1 - P_2)^\mathsf{T} D^\mathsf{T} S^\mathsf{T} \exp\left(-\frac{1}{2\sigma_s^2} S(d(z, z) \odot d(z, z)) \right) \in \mathbb{R}^{3m_0}$$

and the $3m_0 \times 3m_0$ matrix

$$\begin{aligned} \mathrm{d}_{zz} \phi(z, z) = {} & \frac{1}{m_0^2 \sigma_s^4} (P_1 - P_2)^\mathsf{T} D^\mathsf{T} S^\mathsf{T} \text{diag}\left(\exp\left(-\frac{1}{2\sigma_s^2} S(d(z, z) \odot d(z, z)) \right) \right) S D \,(P_1 - P_2) \\ & - \frac{1}{m_0^2 \sigma_s^2} (P_1 - P_2)^\mathsf{T} \text{diag}\left(S^\mathsf{T} \exp\left(-\frac{1}{2\sigma_s^2} S(d(z, z) \odot d(z, z)) \right) \right) (P_1 - P_2) \end{aligned}$$

with $D := \text{diag}(d(z, z))$, respectively.

Remark 9. It is inefficient to store the Hessian pieces $\mathrm{d}_{zz}\phi(z,z)$ and $\mathrm{d}_{zz}\phi(z,x_1)$ explicitly. Instead, they are passed as function handles, evaluating their product with a given vector z.

The Newton step $s^{(k)} \in \mathbb{R}^{3m_0}$ at outer iteration k is given by

$$H^{(k)}s^{(k)} = -g^{(k)}, \quad \tilde{x}^{(k+1)}(t^{n+1}) = \tilde{x}^{(k)}(t^{n+1}) + t^{(k)}s^{(k)}, \tag{37}$$

where $H := \mathrm{d}_{zz}\phi(z,z) + \mathrm{d}_{zz}\phi(z,x_1) \in \mathbb{R}^{3m_0,3m_0}$, $g := \mathrm{d}_z\phi(z,z) + \mathrm{d}_z\phi(z,x_1) \in \mathbb{R}^{3m_0}$ are the Hessian and gradient of the objective function with $\mathrm{d}_{zz}\phi(z,z)$, $\mathrm{d}_{zz}\phi(z,x_1)$, $\mathrm{d}_z\phi(z,z)$ and $\mathrm{d}_z\phi(z,x_1)$ as defined above, and $t^{(k)} > 0$ is a line search parameter. We use a backtracking line search subject to the Armijo condition [70,146]. The solution to this linear system at iteration k is the search direction $s^{(k)}$ for $\tilde{x}$ at time $t^{n+1} = 1$. We solve the linear system $H^{(k)}s^{(k)} = -g^{(k)}$ in (37) iteratively using a matrix-free PCG method. We use a *quadratic forcing sequence* to compute the tolerance ϵ_{PCG} for the PCG method. The term *forcing sequence* refers to an adaptive strategy for selecting the tolerance of iterative methods used to solve for the search direction $s^{(k)}$ in (37) that is commonly used in large scale optimization problems. Instead of solving (37) at every iteration k to high accuracy (i.e., to machine precision, as we would do when considering direct methods), we choose a tolerance that is proportional to the norm of the gradient $g^{(k)}$. This way, we gradually increase the accuracy for the computation of the search direction $s^{(k)}$. Since the overall rate of convergence may not benefit from an accurate estimate of $s^{(k)}$ far away from an optimal point, this allows us to significantly reduce the computational burden. Assuming (optimal) quadratic convergence of our Newton–Krylov solver, we set ϵ_{PCG} to $\epsilon_{\mathrm{PCG}} = \min\left(\|g^{(k)}\|_2/\|g^{(0)}\|_2, \tau_{\mathrm{PCG}}\right)$, with $\tau_{\mathrm{PCG}} = 1/4$. If we were to assume superlinear convergence we can slightly relax this tolerance by replacing $\|g^{(k)}\|_2/\|g^{(0)}\|_2$ with its square root. We note that other options for selecting τ_{PCG} exist. We refer, e.g., to [70,165] for additional details. Notice that the normalization with respect to the initial gradient $\|g^{(0)}\|_2$ at iteration $k = 0$ is not the standard textbook approach. Based on our past experience with solving large scale inverse problems, we prefer this choice since it allows us to avoid issues with the scaling of the gradient g.

2.4.3. Stopping criteria

We now discuss the stopping criteria for the operator-splitting iteration introduced in (31). We separate this discussion into two parts: a discussion of the algorithm convergence of the operator-splitting iteration and a discussion of the convergence of the mapped template shape to the reference shape.

We monitor the convergence of the splitting algorithm using the primal and dual residuals r_{prim} and r_{dual}, which are given by

$$r^{(k)}_{\mathrm{prim}} = \begin{pmatrix} a^{(k)} \\ x^{(k)} \end{pmatrix} - \begin{pmatrix} \tilde{a}^{(k)} \\ \tilde{x}^{(k)} \end{pmatrix} \quad \text{and} \quad r^{(k)}_{\mathrm{dual}} = \rho\left(\begin{pmatrix} \tilde{a}^{(k)} \\ \tilde{x}^{(k)} \end{pmatrix} - \begin{pmatrix} \tilde{a}^{(k-1)} \\ \tilde{x}^{(k-1)} \end{pmatrix}\right), \tag{38}$$

respectively. Here, the primal residual monitors the convergence of variables to a consensus solution, while the dual residual monitors the convergence to a fixed solution across iterations. Both residuals can be shown to converge to zero [155]. Thus, a suitable criteria for convergence of the splitting method in (31) is to stop when the norms of r_{prim} and r_{dual} falls below some tolerance ϵ_l, $l \in \{\mathrm{prim}, \mathrm{dual}\}$, respectively. Guidelines for selecting ϵ_{prim} and ϵ_{dual} are provided in [153,155].

While these criteria assess the convergence of the optimization algorithm, we are ultimately interested in matching the template shape x_0 to the reference shape x_1. Consequently, we add an additional convergence criteria that evaluates the discrepancy between the deformed template shape x at t^{n+1} and the reference shape x_1. We consider the Hausdorff distance [57,62]. For two generic shapes $x = (x^1, \dots, x^{m_0}) \in \mathbb{R}^{3m_0}$ and $z = (z^1, \dots, z^{m_1}) \in \mathbb{R}^{3m_1}$, the Hausdorff distance is defined as

$$\mathrm{dist}_H(x,z) = \max\{d_H(x,z), d_H(z,x)\}, \tag{39}$$

where $d_H(x,z) = \max_{i\in\{1,\dots,m_0\}}\left\{\min_{j\in\{1,\dots,m_1\}} \|x^i - z^j\|_2\right\}$. In practice, we use the 95th percentile of the Hausdorff distance to guard against outliers. From a mathematical point of view, this censoring is not required. However, we consider this confidence interval to account for the fact that in practical applications we cannot expect the data to be represented accurately, due to measurement errors, noise, and a limited resolution as well the modeling errors in their representation. Additional sources of errors stem from our problem formulation along with the numerical discretization. Consequently, we do not expect the matching to be perfect everywhere. Adding a 95% confidence interval allows us to account for these inaccuracies. We terminate the algorithm if this censored Hausdorff distance falls below a user defined threshold $\epsilon_{\mathrm{haus}} > 0$. Moreover, we monitor the update of the Hausdorff distance across five consecutive iterations. Lastly, to safeguard against excessive runtimes, we terminate the optimization if the number of iterations exceeds a user defined upper bound n_{iter}.

In summary, the stopping conditions are

$$(C1) \qquad \mathrm{dist}_H(Qx^{(k)}, x_1) < \epsilon_{\mathrm{haus}}, \tag{40a}$$

$$(C2) \qquad \textstyle\sum_{j=0}^{4} |\delta_{\mathrm{haus}}(j)| < \epsilon_{\mathrm{haus}}/10^3, \tag{40b}$$

$$(C3) \qquad \|r^{(k)}_{\mathrm{prim}}\|_2 < \epsilon_{\mathrm{prim}}, \tag{40c}$$

Table 2
Algorithmic parameters. We select these parameters empirically. We describe some of the policies we have developed to select them in §2.4.4.

Symbol	Meaning	Value
n	number of time steps (time-discretization)	5
α	parameter controlling the contribution of the distance (20)	1
σ_v	bandwidth of Gaussian kernel for parameterization of v (see (6)	see (42)
σ_s	bandwidth of Gaussian kernel for kernel distance (20)	see (43)
ρ	operator-splitting parameter (see (31))	1
stopping conditions		
ϵ_{prim}	stopping tolerance for primal residual	1×10^{-3}
ϵ_{dual}	stopping tolerance for dual residual	1×10^{-3}
ϵ_{haus}	tolerance for censored Hausdorff distance in (39)	see (44)
n_{iter}	maximum number of iterations	100

$$(C4) \qquad \|r^{(k)}_{\text{dual}}\|_2 < \epsilon_{\text{dual}}, \tag{40d}$$

$$(C5) \qquad k > n_{\text{iter}}, \tag{40e}$$

with $\delta_{\text{haus}}(j) := \text{dist}_H(Qx^{(k-j)}, x_1) - \text{dist}_H(Qx^{(k-(j+1))}, x_1)$. We terminate the algorithm if

$$(C1) \vee (C2) \vee (C3) \vee (C4) \vee (C5). \tag{41}$$

We overview the entire algorithm in Algorithm 1.

Algorithm 1 Pseudo algorithm for operator splitting approach. We summarize the values and heuristic/empirical policies to select algorithmic and problem specific parameters in Table 2.

```
Input: $x_0, x_1, a^{(0)} = 0$
Parameters: $n_{\text{iter}}$, $\epsilon_l > 0$, $l \in \{\text{prim}, \text{dual}, \text{haus}\}$, $\rho > 0$, $\alpha > 0$, $\sigma_v, \sigma_s > 0$, $n \in \mathbb{N}$
$k \leftarrow 0$
while $\neg$ stop do
    $\left(x^{(k+1)}, a^{(k+1)}\right) \leftarrow$ given $\left(u^{(k)}, w^{(k)}\right)$ and $\left(\tilde{x}^{(k)}, \tilde{a}^{(k)}\right)$ solve kinetic energy subproblem (32)
    $\left(\tilde{x}^{(k+1)}, \tilde{a}^{(k+1)}\right) \leftarrow$ given $\left(x^{(k+1)}, a^{(k+1)}\right)$, $\left(u^{(k)}, w^{(k)}\right)$ solve distance subproblem (35)
    $\begin{pmatrix} u^{(k+1)} \\ w^{(k+1)} \end{pmatrix} \leftarrow \begin{pmatrix} u^{(k)} \\ w^{(k)} \end{pmatrix} + \begin{pmatrix} \tilde{x}^{(k+1)} \\ \tilde{a}^{(k+1)} \end{pmatrix} - \begin{pmatrix} x^{(k+1)} \\ a^{(k+1)} \end{pmatrix}$
    $\left(r^{(k+1)}_{\text{prim}}, r^{(k+1)}_{\text{dual}}\right) \leftarrow$ evaluate (38)
    $k \leftarrow k+1$
    stop $\leftarrow$ evaluate (41)
end while
Output: $a^{(k)}$
```

Remark 10. We can use various measures to assess the proximity between geometric structures. In the present case, we are–in essence–interested in matching point clouds. To measure the proximity between points in $\mathbb{R}^3$, the Hausdorff distance is a natural choice since it has an intuitive geometric interpretation. This is not the case for the kernel distance we consider for optimization. Due to its geometric interpretation, we prefer using the Hausdorff distance as a stopping criterion. Conversely, while the use of the Hausdorff distance to measure the proximity of the deformed shape $y \cdot s_0$ and the target shape s_1 may seem natural, it poses significant challenges in the context of variational optimization (from a theoretical and a practical point of view); it is not smooth with respect to small perturbations of the compared shapes. The kernel distance is. As such, we prefer it for optimization. A more detailed discussion can, e.g., be found in [57].

2.4.4. Parameter choices

We have to select several parameters. We summarize these parameters in Table 2. This section includes a brief description of each of those parameters and details how they were selected for our numerical experiments.

Number of time steps The number of cells $n \in \mathbb{N}$ used to discretize the time interval $[0, 1]$ into $n+1$ time steps has to be selected as a trade-off between accuracy requirements and computational complexity. In our implementation, it affects the discretization of the kinetic energy and the ODE constraint in (4). As stated in §2.3.2, we compute the integral with respect to time using a left Riemann summation [141]. As stated in §2.3.4, we discretize the ODE using the forward Euler method. It follows that the error associated with the time integration is of order $h = 1/(n+1)$. Consequently, a finer discretization in time increases the numerical accuracy in a linear way. Increasing the number of time steps increases the computational cost of the problem. Thus, although the kinetic energy and the discretization of the ODE both benefit from a finer discretization, this is tempered by the need to keep a computationally tractable size for n. If we register two arbitrary shapes (e.g., shapes representing organs of two distinct individuals), we have to select an adequate number of time steps. For the experiments in this paper, we set $n = 5$. We found empirically that $n = 5$ yields a good trade off between computational complexity and accuracy requirements in the considered application. Another scenario we consider in the present work is the diffeomorphic matching of a time series of smooth shapes s_i, $i = 0, \dots, n_f - 1$. In this case, we, e.g., monitor (for an individual patient) the motion of an organ of interest in time. (In our case, the motion of the leaflets of mitral valves during the cardiac cycle.) Consequently, the data consists of a fixed number of frames n_f. The task is to reconstruct the motion between the individual frames (or, e.g., from frame at pseudo-time $t = 0$ (first frame) to frame at pseudo-time $t = 1$ (last frame)). In this case, we also have to select an adequate number of time steps for the discretization. We select to set n equal to the number of frames n_f.

Weight for distance measure The problem also requires the selection of the parameter $\alpha > 0$. This parameter balances the two terms of the objective functional in (4): the kinetic energy enforces smoothness of the velocity and the distance term which enforces the matching between the deformed template shape and the reference shape. Selecting this parameter is a much researched topic in the field of inverse problems, and common selection methods include generalized cross-validation, L-curve, Morozov's discrepancy principle, or the unbiased predictive risk estimator method; see e.g. [166–169]. In [15,57] we have developed parameter-continuation schemes to select appropriate values for the hyperparameters that balance the kinetic energy (regularization) and the distance measure. The basic idea is to solve a family of optimization problems parameterized by the weight α, starting with small values for α. Consequently, in a first step we essentially optimize the kinetic energy (i.e., we basically solve a convex, quadratic optimization problem). We can expect quick convergence. Subsequently, we increase α to improve the matching. We use the solution from the former step as an initialization ("warm start"). For simplicity of our numerical study presented here, we do not consider this scheme.

Standard deviation for kernel matrices The parameterization of the velocity as elements of an RKHS in (6) as well as the kernel distance in (20) require the selection of the scale parameters/bandwidths $\sigma_v > 0$ and $\sigma_s > 0$, respectively. If the scale parameter σ_v increases, the smoothness of the optimal diffeomorphic deformation increases. While smooth deformations are desirable, too large values for σ_v may lead to a poor matching quality. Conversely, σ_s controls how many points in a local neighborhood $\mathcal{N}(x_0^i) := \{z \in \mathbb{R}^3 : \|x_0^i - y\|_2^2 \le r, r > 0, x_0^i \in x_0\}$, $i \in \{1, \dots, m_0\}$, of radius $r > 0$ of a point $x_0^i \in x_0$ influence the movement of point x_0^i of the template shape x_0. If σ_s is chosen too small, the Gaussian kernels are too narrow; few points contribute locally to the distance functional in (20) and, thus, a point x_0^i may not move towards points in the target shape x_1. In contrast, choosing σ_s too large means the Gaussian kernels are too wide. This causes individual points x_0^i to be affected by too many points. This increases the difficulty of optimization, and may result in a poor matching quality, as well. Consequently, policies to control the selection of these parameters are important; we select policies that take into account the geometry of the problem. In particular, they consider the average distance between points in the point cloud (or mesh size) and strike a balance between local and global matching of shapes.

We consider criteria similar to the ones proposed in [57]. Recall that the shapes s_i, $i = 0, 1$, are parameterized as a set of points/landmarks $x_i \in \mathbb{R}^{3m_i}$. For these sets of points x_i, we can compute a triangulation to approximate the surface of s_i. We select σ_v to be proportional to the mean edge length $\bar{h}_0$ of the Delaunay triangulation of the template shape x_0. That is, let $\mathcal{E}(x_0^i)$ denote a set of all points x_0^j, $j \neq i$, of x_0 that share an edge with x^i. Then,

$$\bar{h}_0 := \operatorname{mean}\left(\bigcup_{i=1}^{m_0} \left\{ \|x_0^j - x_0^i\|_2^2 : x_0^j \in \mathcal{E}(x_0^i), j = 1, \dots, m_0 \right\}\right).$$

We have

$$\sigma_v = \tau_v \, 2^{-1/2} \, \bar{h}_0, \tag{42}$$

where $\tau_v > 0$ is a user selected parameter.

The bandwidth of the distance is selected to be proportional to the Hausdorff distance between the shapes x_0 and x_1 according to

$$\sigma_s = \max\left(\bar{h}_1, \tau_s \operatorname{dist}_H(x_0, x_1)/2\right), \tag{43}$$

where $1 \le \tau_s \le 2$ is a user defined parameter, dist_H is the Hausdorff distance (39) and $\bar{h}_1$ is the average mesh size of the target shape x_1. We note that we observed numerical issues if σ_s is at the order of or below the average mesh size of the target shape (the computed search direction is no longer a descent direction and the convergence of the solver deteriorates). This is the reason for the lower bound in (43). We conduct numerical experiments to illustrate the behavior of our methodology for different choices of τ_v and τ_s, respectively.

Remark 11. One possibility to make the selection of the bandwidth completely automatic is to use a parameter continuation in σ_v and σ_s (or τ_v and τ_s, respectively; i.e., a multiscale approach). The question on how to combine this multiscale strategy with a parameter continuation in α remains subject to future work.

Stopping tolerances As can be seen in condition (C1) of (40), the stopping tolerance for the censored Hausdorff distance is selected proportional to the average mesh size $\bar{h}_1$ of the discretized target shape x_1. We use

$$\epsilon_{\text{haus}} = \tau_{\text{haus}}\bar{h}_1, \quad \tau_{\text{haus}} > 0. \tag{44}$$

We found $\tau_{\text{haus}} = 1/2$ yields a good trade-off between accuracy and runtime requirements. The condition (C2) monitors the change of the Hausdorff distance across five iterations. That is, if we do not make significant progress in five consecutive iterations we terminate the solver. To avoid additional parameters, we select the distance updates as a fraction of the tolerance ϵ_{haus} used for the Hausdorff distance.

Since we are mostly interested in driving the mismatch between the data to zero, the remaining tolerances are treated as safeguards to avoid excessive computations. We select them as stated in Table 2.

2.5. Quantitative measures of shape variability

After computing the optimal diffeomorphic map $f^{\text{opt}} = (f_t^{\text{opt}})$ (state variable) that approximately minimizes (4) and by that matching two surfaces $s_0, s_1 \in \mathcal{S}$, the terminal $\mathbb{R}^3$-diffeomorphism $y := f_1^{\text{opt}} = f^{\text{opt}}(t = 1, \cdot)$ is a smooth invertible map from s_0 onto a smooth surface $y \cdot s_0 = y(s_0) \approx s_1$. In a Riemannian setting, we can—according to (3)—compute the distance between two arbitrary shapes s_i, $i = 0, 1$, by evaluating the kinetic energy (2) for a minimizer $v^{\text{opt}} = (v_t^{\text{opt}})$ of (2) that parameterizes the diffeomorphic flow f_t that maps s_0 to s_1. We refer to [36,48,71,137] for a rigorous mathematical treatment. Intuitively, this corresponds to the minimal energy required to map s_0 to s_1. Clearly, even in exact arithmetic we can only hope to compute an approximation to this distance, since the considered control problem is high-dimensional after discretization, nonconvex, and nonlinear. Moreover, the computation of the distance is also affected by the implementation choices we made. Aside from numerical tolerances or numerical accuracy of the carried out computations, any changes of the parameterization of the curved shapes, any reparametrization of these shapes, or any changes in the RKHS structure for the parameterization of the velocity yields different results. Moreover, computing geodesic distances between two shapes—even if we assume that the shapes s_0, s_1, are in the orbit associated with the group action of the diffeomorphisms generated here—assumes an exact matching, i.e., $y \cdot s_0 = s_1$. Given the issues described above as well as perturbations introduced by discretization and measurement errors, this assumption does not hold in practice. We can only hope for an approximate matching $y \cdot s_0 \approx s_1$. Lastly, the minimal kinetic energy involves averages of squared velocities, and hence only provides a global dissimilarity between s_0 and s_1. Strain analysis of f, which we now outline as in [46,62], generates the spatial distribution of local distortions between s_0 and s_1.

Let $x \in s_0$ denote any fixed point and let $\tilde{x} = y(x)$, where $y = f_1$ again denotes the terminal diffeomorphism at $t = 1$. (For simplicity, we limit this exposition to the terminal diffeomorphism; the computation naturally translates to all time points t of the flow f_t). Moreover, let $\mathcal{T}_x$ and $\mathcal{T}_{\tilde{x}}$ denote the tangent spaces of the initial shape s_0 and the deformed shape $\tilde{s}_0$ at x and $\tilde{x}$, respectively, endowed with local surface metrics. Since $y \in \mathcal{G}$ is a smooth bijection, the differential of y is an invertible linear map $J_x : \mathcal{T}_x \to \mathcal{T}_{\tilde{x}}$, $J_x \in \mathbb{R}^{2,2}$. Let $u \in \mathcal{T}_x$ denote any tangent vector with $\|u\| = 1$. With this, we can define the directional strain at x associated with the map y in the direction of u as the dilation/contraction factor $q_{\text{dir}}(x, u) = \|J_x u\|$. Let $J_x^* : \mathcal{T}_{\tilde{x}} \to \mathcal{T}_x$ denote the adjoint operator of J_x. Moreover, let $\lambda_x^i > 0$, $i = 1, 2$, with $\lambda_x^1 \leq \lambda_x^2$ denote the eigenvalues of $J_x^* J_x \succ 0$, with $\sqrt{\lambda_x^i}$, $i = 1, 2$, equal to the minimal and maximal directional strains at x, respectively. To avoid anisotropies associated with the definition of q_{dir}, we consider the isotropic strain q_{iso} defined as

$$q_{\text{iso}}(x) = \left(\prod_{i=1}^{2} \lambda_x^i\right)^{1/2} = \sqrt{|\det(J_x)|},$$

instead. It corresponds to the local isotropic change of length for the tangent vector. We note that q_{iso} has a simple geometric interpretation. For fixed $x \in s_0$, and any open patch $p_x \subset s_0$ around x, define the ratio of surface areas

$$\text{area}_{\text{rel}}(p_x) = \frac{\text{area}(y(p_x))}{\text{area}(p_x)}.$$

Then, area_{rel} tends to the square of q_{iso} at x when the diameter of p_x tends to 0. This observation allows us to compute strain directly from the triangulation after discretization of s_0. Since q_{iso} is a dimensionless quantity of average dilation/contraction around x, we convert it into an “isotropic strain intensity” defined as

$$p_{\text{iso}}(x) = |q_{\text{iso}}(x) - 1|.$$

In [62] we have used measures derived from the kinetic energy and strain intensity to study differences between datasets of normal and diseased patients. In [46], we have extended this idea by exploiting these quantitative measures of shape variability as features for the automatic classification of shapes and shape deformations based on random forests. Here, we merely consider isotropic strain as a visual tool to evaluate complex diffeomorphic deformations.

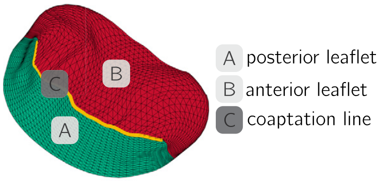

Fig. 4. Main anatomical structures for a representative mitral valve dataset. We show a Delaunay triangulation of the mitral valve surface. We highlight the main anatomical regions: the posterior leaflet (teal color), the anterior leaflet (red color), and the coaptation line (gold). Each leaflet is discretized by a grid of 800 points, resulting in a total of 1,600 points per 3D surface. We show the Delaunay triangulation of this mesh. (Figure modified from [46].)

3. Numerical experiments

In the following, we report results for the numerical software developed by our group for solving the diffeomorphic shape matching problem formulated in (4). The application is cardiac imaging, and, in particular, surface representations of mitral valves of individual patients extracted from image sequences.

3.1. Data

For a total of 150 patients, transesophageal echocardiography provides dynamic 3D views of their mitral valves at rates of roughly 25 frames per heart cycle with potential mitral valve complications. Computerized segmentation of these 3D images extracts twenty-five 3D snapshots of the mitral valve surface. For time frame t, the extracted surface displays the anterior and posterior leaflets of the mitral valve as smooth surfaces discretized by a grid of 800 points per leaflet. Consequently, each 3D surface is discretized by 1 600 points per frame. These data were prepared and annotated by our longtime collaborators at the DeBakey Heart and Vascular Center of the Houston Methodist Hospital. During each heart cycle, the leaflets close the mitral valve at mid-systole ($t = t_{\text{ms}}$) and open the mitral valve at end-systole ($t = t_{\text{es}}$). The two leaflets define a deformable connected 3D-surface bounded by a flexible ring (the "annulus"). At $t = t_{\text{ms}}$, the leaflets are in full contact along the *coaptation line* in order to tightly close the mitral valve. The leaflets open the mitral valve progressively until end-systole, then remain fully open during diastole, and start closing again at beginning of systole. When the mitral valve is open, the coaptation line is split into two curved boundary segments sharing the same endpoints. We show the main anatomical regions of these data in Fig. 4.

In previous studies [57,62,66,67,170], diffeomorphic shape matching of the mitral valve leaflets between $t = t_{\text{ms}}$ and $t = t_{\text{es}}$ (a problem referred to as "multi-frame registration" below) was developed and systematically implemented in order to first reconstruct the smooth deformation $t \to f_t$ of the mitral valve leaflets between mid systole and end systole. We then computed the intensities and spatial distribution of the tissue strain induced by mitral valve deformation between mid systole and end systole (see §2.5). This research project aimed to provide cardiologists with patient specific displays of mitral valve leaflet strain intensities as a potential aid to evaluate/compare mitral valve clinical cases [66,67,170]. In a recent followup paper, we develop diffeomorphic deformation techniques for automatic classification of soft smooth shapes [46]. We have used the described database of 3D mitral valve snapshots as a benchmark to implement and test methodology for automatic classification of smooth surfaces; more precisely, to discriminate patients diagnosed with *mitral valve regurgitation* from *normal* cases. In the present work, we consider these 3D mitral valve snapshots to study the numerical performance of our improved numerical solver for the operator splitting problem described above.

3.2. Bandwidth selection

We perform experiments to study the effect of the parameters τ_s and τ_v for the bandwidth σ_s of the kernel distance in (20) and σ_v for the parameterization of the velocity in (6), respectively. We observe that these parameters have a significant effect on the performance of our method. To obtain comparable results, we execute the solver for a fixed number of $n_{\text{iter}} = 100$ iterations; i.e., we neglect all convergence criteria in (40) with the exception of the upper bound on the number of iterations. This also allows us to explore adequate values for the tolerances for the stopping conditions listed in (40) for the considered problem (registration of mitral valve data). Notice that these experiments are different from our previously reported results [62]. In [62] we performed the registration in a single patient in time (frame-to-frame). Here, we fix the frame to the time point corresponding to mid-systole and register two data sets from different patients. We expect this to be a more challenging problem due to an expected increase in shape variability across patients.

Setup We limit the exposition in the main text to a single patient. We provide additional results in Appendix B. The overall performance for these additional patients is consistent with the results reported here. We vary the scaling τ_s in (42) for the RKHS representation of the bandwidth σ_v of the Gaussian kernel in (6) between 3 and 8. More precisely, we select values for τ_v in $\{3, 4, 6, 8\}$. For values smaller than 3 and values larger than 8 we observed a deterioration of the performance of our solver. For each choice of τ_v we change the scaling τ_s in (43) for the bandwidth σ_s of the kernel distance in (20). More precisely, we select values for τ_s in $\{3/4, 1, 2, 4, 6\}$. Likewise to τ_v, we observed a deterioration of the performance of our

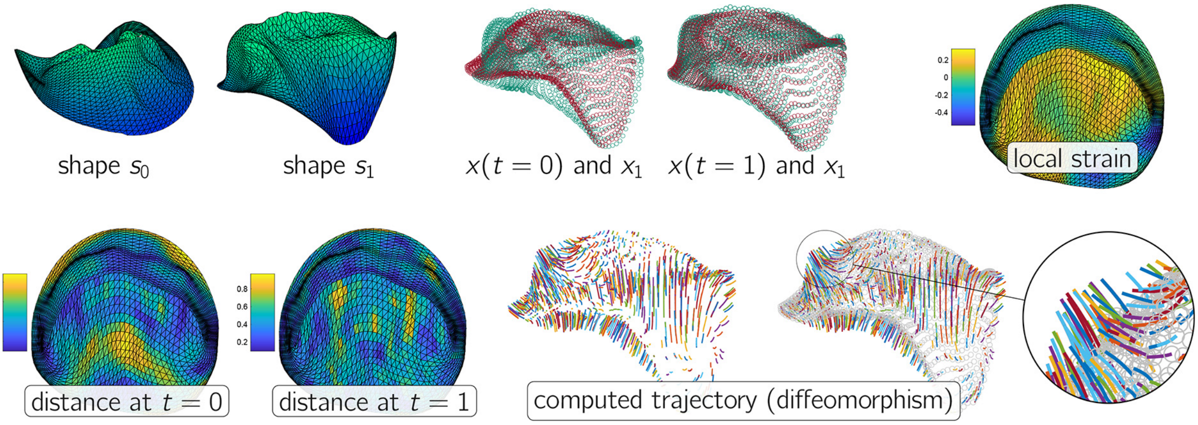

Fig. 5. Results for the matching of two mitral valve surfaces acquired from two different patients from our database. The results shown here correspond to the run with $\tau_s = 1$ and $\tau_v = 6$ reported in Table 3 (row 12). The top row (from left to right) shows the original template shape s_0 and the target shape s_1 (Delaunay triangulations), the associated meshes in overlay before registration (template shape $x(t=0)=x_0$ in red and target shape x_1 in teal) and after registration (deformed template shape $x(t=1)$ in red and target shape x_1 in teal), and the computed strain intensity values overlaid on the target shape. The first two views in the top row show the entire mitral valve in their initial configuration (both leaflets). The first two views in the bottom row (point-wise distances from the template to the target shape) show the mitral valve from the top. For these two views one can identify the coaptation line that separates both leaflets; the coaptation line appears as a ridge between the anterior and posterior leaflet (see Fig. 4 for an illustration). The bottom row shows (on the left) the point-wise distance from the template to the target shape before (at pseudo-time $t=0$) and after (at pseudo-time $t=1$) registration. The values are normalized with respect to the mean edge length of the target shape (i.e., values of 1.0 are equal to the average edge length). We also show the computed trajectory for 75% of the points (randomly chosen; left: vector field only; right: vector field overlaid on the template shape $x(t=0)=x_0$). The circle shows a closeup of the computed vector field (zoomed view).

methodology if we selected parameters outside the range $[3/4, 6]$ for the considered data sets. Moreover, for $\tau_s = 1/2$ the estimate for the bandwidth σ_s is below $\bar{h}_1 = 1.3$.

Results We visualize results for the considered dataset in Fig. 5. These results correspond to the parameter choices $\tau_v = 6$ and $\tau_s = 1$. In particular, we show the original template shape s_0 and the target shape s_1 in the top left corner (the surfaces are approximated from the meshes x_0 and x_1 through a Delaunay triangulation). In the middle block of the first row in Fig. 5 we show the template shape $x(t=0)=x_0$ and the reference shape before and after diffeomorphic shape matching. We can observe that the distance between the mesh points is initially large and becomes significantly smaller after diffeomorphic registration. The rightmost plot of the top row in Fig. 5 shows the associated isotropic strain intensity (see §2.5). We can observe for this particular patient pair that the strain is on average larger on the anterior leaflet than the posterior leaflet. The bottom row shows the point-wise distance of each mesh point from the template to the target shape before (left) and after (right) diffeomorphic shape matching. Notice that the color map captures a different range of values before and after matching. On the right of the bottom row we illustrate the computed diffeomorphism (trajectory of the points x_0, i.e., the state variable $x(t)$) for 75% of the points on the template shape selected at random. We provide two illustrations, one (left) that only shows the vector field and (right) the vector field together with the template shape $x_0 = x(t=0)$ in faint gray. We also show a closeup of the computed trajectory on the right.

Overall, the most important observations of this experiment are: (i) Although the shapes are drastically different, we obtain a good matching quality, with pointwise distances below $1/2$ of the target mesh resolution. (ii) The computed diffeomorphism (trajectory) is quite complicated.

In Table 3 we summarize the overall performance of our solver for the considered values for τ_v and τ_s, respectively. We report the final (censored) Hausdorff distance (after diffeomorphic matching) in the third column from the left. We also report (in brackets) the percentage the obtained distance is with respect to the original Hausdorff distance of 3.0. We also report the absolute and relative values of the norm of the primal and dual residuals in columns four and five. The last column shows the runtime for the solver in seconds (for 100 iterations).

The most important observations are: (i) We can reduce the distance by up to 70% for the choices $\tau_v = 4$ and $\tau_v = 6$ for $\tau_s \in \{3/4, 1\}$, respectively. (ii) The performance of our methodology in terms of registration quality (Hausdorff distance) as well as runtime is much more sensitive to the choice of τ_s compared to τ_v. (iii) The final norms of the residuals are only slightly affected by the choices for τ_v and τ_s.

In Fig. 6 we illustrate the convergence behavior of our solver. The horizontal axis is the iteration count of the splitting method (outer iteration index k) for all plots. We show (from left to right) (i) the trend of the Hausdorff distance (normalized by the initial distance at iteration 1), (ii) the trend of the updates of the Hausdorff distance, (iii) the trend of the relative primal residual, and (iv) the trend of the relative dual residual.

The most important observations are the following: (i) The smaller τ_s the more accurate the matching. The results for $\tau_s = 3/4$ and $\tau_s = 1$ are quite similar. (ii) Increasing τ_v improves the registration accuracy; the optimal accuracy is obtained for $\tau_v = 6$. (In Table 3 we saw that for $\tau_v = 8$, the accuracy decreases again.) (iii) The updates for the Hausdorff distance

Table 3
Performance for different choices of the scaling parameters τ_v and τ_s that control the bandwidth σ_v and σ_s of the RKHS parameterization and the kernel distance, respectively. The results are for two representative patients from our database. The average mesh size $\bar{h}_1$ of the target shape is 1.3. We select τ_v in $\{3, 4, 6, 8\}$ and τ_s in $\{3/4, 1, 2, 4, 6\}$. The initial (censored) Hausdorff distance for this dataset is 3.0. We report (from left to right) the final Hausdorff distance (after matching; absolute value and percentage of initial value in brackets), the (relative) norm of the primal residual, the (relative) norm of the dual residual, and the runtime of the solver (in seconds).

τ_v	τ_s	final distance (%)	residuals		runtime
			primal (relative)	dual (relative)	
3	0.75	1.2 (38.97)	4.5×10^{-3} (3.6×10^{-4})	5.4×10^{-1} (4.3×10^{-2})	100.4
	1	1.2 (39.36)	2.9×10^{-3} (2.1×10^{-4})	4.8×10^{-1} (3.5×10^{-2})	83.4
	2	1.5 (48.66)	2.1×10^{-3} (1.5×10^{-4})	3.8×10^{-1} (2.7×10^{-2})	68.9
	4	2.1 (70.68)	2.1×10^{-3} (2.1×10^{-4})	3.0×10^{-1} (3.0×10^{-2})	63.8
	6	2.5 (85.00)	1.9×10^{-3} (2.8×10^{-4})	2.8×10^{-1} (4.2×10^{-2})	62.4
4	0.75	0.9 (30.38)	5.9×10^{-3} (4.7×10^{-4})	5.9×10^{-1} (4.7×10^{-2})	91.3
	1	0.9 (31.38)	4.0×10^{-3} (2.9×10^{-4})	5.2×10^{-1} (3.8×10^{-2})	82.9
	2	1.2 (40.64)	2.8×10^{-3} (2.0×10^{-4})	4.2×10^{-1} (3.0×10^{-2})	69.3
	4	1.9 (63.04)	2.8×10^{-3} (2.9×10^{-4})	3.5×10^{-1} (3.6×10^{-2})	61.8
	6	2.3 (78.46)	2.5×10^{-3} (3.8×10^{-4})	3.4×10^{-1} (5.2×10^{-2})	60.7
6	0.75	0.9 (30.43)	7.6×10^{-3} (6.1×10^{-4})	6.8×10^{-1} (5.4×10^{-2})	92.4
	1	0.9 (31.00)	5.0×10^{-3} (3.6×10^{-4})	6.1×10^{-1} (4.5×10^{-2})	78.3
	2	1.1 (38.34)	3.3×10^{-3} (2.3×10^{-4})	4.9×10^{-1} (3.5×10^{-2})	78.8
	4	1.7 (56.93)	3.2×10^{-3} (3.2×10^{-4})	4.1×10^{-1} (4.1×10^{-2})	65.6
	6	2.2 (72.29)	2.7×10^{-3} (4.1×10^{-4})	4.1×10^{-1} (6.2×10^{-2})	64.2
8	0.75	1.0 (33.19)	8.6×10^{-3} (6.9×10^{-4})	7.5×10^{-1} (6.0×10^{-2})	94.1
	1	1.0 (33.86)	6.4×10^{-3} (4.7×10^{-4})	6.7×10^{-1} (4.9×10^{-2})	86.1
	2	1.2 (41.49)	3.5×10^{-3} (2.5×10^{-4})	5.2×10^{-1} (3.7×10^{-2})	75.4
	4	1.7 (56.72)	3.0×10^{-3} (3.0×10^{-4})	4.0×10^{-1} (4.1×10^{-2})	64.8
	6	2.1 (70.46)	2.6×10^{-3} (4.0×10^{-4})	4.0×10^{-1} (6.1×10^{-2})	59.5

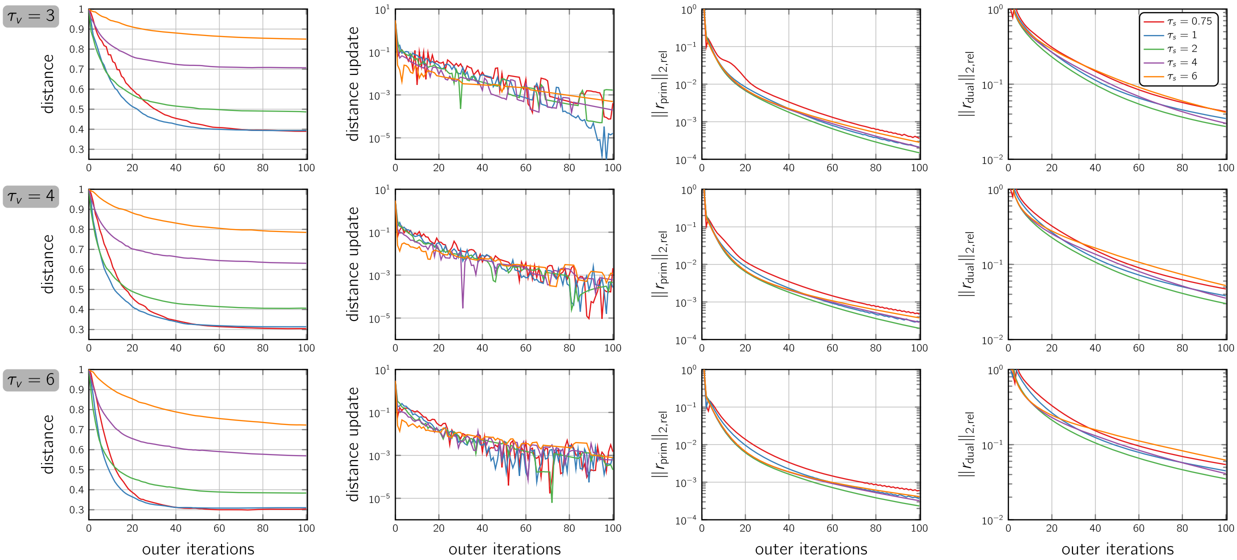

Fig. 6. Performance for different choices of the scaling parameters τ_v and τ_s that control the bandwidth σ_v and σ_s of the RKHS parameterization and the kernel distance, respectively. The results are for two representative patients from our database. The average mesh size $\bar{h}_1$ of the target shape is 1.3. The results correspond to the first, second and third block in Table 3. From top to bottom (row 1 through 3) we report results for $\tau_v = 3$, $\tau_v = 4$ and $\tau_v = 6$. Each plot shows results for $\tau_s \in \{3/4, 1, 2, 4, 6\}$ (see legend). The first column shows the trend of the Hausdorff distance with respect to the number of iterations (normalized with respect to the initial value). Notice that we do not minimize the Hausdorff distance in our optimization but the kernel distance in (20). The second column shows the update of the Hausdorff distance per iteration. The third and fourth column shows the norm of the primal and dual residual versus the iteration count. These residuals are normalized with respect to the residual obtained at iteration 2.

decrease with increasing iterates. (iv) The norms of the residuals decrease (almost) monotonically across all choices of parameters τ_s and τ_v. (v) The trend of the residual norms is only slightly affected by the choices of the parameters τ_s and τ_v, respectively.

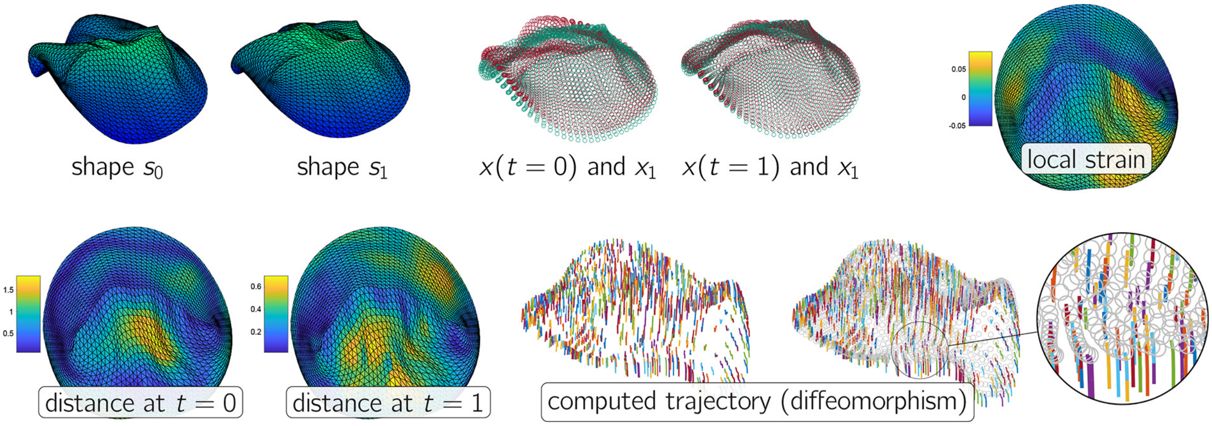

Fig. 7. Results for diffeomorphic matching between the two mitral valve surface snapshots acquired from a single patient at mid systole and end systole (first patient in Table 4). We set $\tau_s = 1$ and $\tau_v = 6$. The top row (from left to right) shows the template shape s_0 (mitral valve at mid-systole; Delaunay triangulation) and the target shape s_1 (mitral valve at end-systole; Delaunay triangulation), the associated meshes in overlay before registration (template shape $x(t=0) = x_0$ in red and target shape x_1 in teal) and after registration (deformed template shape $x(t=1)$ in red and target shape x_1 in teal), and the computed strain intensity values overlaid on the target shape. The first two views in the top row show the entire mitral valve in their initial configuration (both leaflets). The first two views in the bottom row (point-wise distances from the template to the target shape) show the mitral valve from the top. For these two views one can identify the coaptation line that separates both leaflets; the coaptation line appears as a ridge between the anterior and posterior leaflet (see Fig. 4 for an illustration). The bottom row shows (on the left) the point-wise distance from the template to the target shape before (at pseudo-time $t=0$) and after (at pseudo-time $t=1$) registration. The distance values are normalized with respect to the mean edge length of the target shape (i.e., values of 1.0 are equal to the average edge length). We also show the computed trajectory for 75% of the points (randomly chosen; left: vector field only; right: vector field overlaid on the template shape $x(t=0) = x_0$). The circle shows a closeup of the computed vector field (zoomed view).

3.3. *Multi-frame registration*

We report results for the registration across time of successive mitral valve surface snapshots acquired between mid-systole and end-systole for one individual patient. That is, we are given data at different points in time and try to compute the diffeomorphic flow from the first frame at mid systole through all data to the last frame at end systole. All available frames enter the kernel distance (i.e., we sum over all frames). This setup is equivalent to the one considered in [57,62]. The real time duration of this typical 25 frames sequence is less than half a second.

Remark 12. We note that the mitral valve changes topology as we go from mid-systole to end-systole. That is, the mitral valve opens. This constitutes a significant problem when trying to compute a diffeomorphism that maps corresponding points to one another and preserves the integrity of the considered anatomy. Based on our parameterization of the curved surface, we compute a diffeomorphism that takes us from one set of points to the other. However, it is not guaranteed that we do not match points situated on one leaflet to those on another. One simple fix to this problem is to register the leaflets of the mitral valves individually. Our data has the anatomical annotations to do so (see Fig. 4). In fact, our software supports this, and we have tested this strategy to classify patient data [46]. In the present work, we are solely interested in studying the performance of the proposed solver. As such, we consider the entire mitral valve data and do not register the leaflets individually. Since the boundaries of individual leaflets are well identified even when the mitral valve is closed, one can mitigate the initial topology change by using high weights for leaflet boundary points in the surface matching terms (see §2.3.3 for a discussion).

On a more general note, using diffeomorphic registration in problems where the topology changes and/or structures are not present in both datasets to be registered is a delicate problem, and an active area of research. We refer to [171–177] for examples for medical images and [143,144,178,179] for work on shape representations that attempt to deal with this issue.

Setup We use the parameter values for our surface matching solver as suggested by our results in Table 2.

More precisely, we set the weight for the kernel distance is set to $\alpha = 1$. The scaling parameters τ_s and τ_v are set to 1 and 6, respectively (based on the observations from our prior numerical experiments reported in §3.2). The parameter ρ is set to 1. The tolerances for the primal and dual residual are $\epsilon_{\text{prim}} = 1 \times 10^{-3}$ and $\epsilon_{\text{dual}} = 1 \times 10^{-3}$, respectively. The scaling for the tolerance of the censored Hausdorff distance is $\tau_{\text{haus}} = 1/2$. We limit the maximum number of iterations to 100 (not actually reached).

The main difference compared to the experiments reported in the former section is that we now have several 3D-image data available to control our diffeomorphic registration results: multiple frames that monitor the heart cycle of an individual patients. We compute the diffeomorphism that flows the first shape at time $t = 0$ (mid-systole) through all these data until we reach the final shape at $t = 1$ (end-systole). See [62] for additional details.

Table 4
Results for diffeomorphic matching between mid systole and end systole mitral valve surface snapshots for 20 representative patients in our database. All run times are reported in seconds. We report (first four columns from left to right): (i) the run id (different patients in each row), (ii) the number of outer iterations of our splitting algorithm, (iii) the final censored Hausdorff distance after diffeomorphic matching (% of original distance in brackets), and (iv) the overall time to solution (TTS). In the fifth and the sixth column we report run times for the distance subproblem. In particular, we report (v) the runtime spent on solving the distance subproblem (percentage of TTS in brackets) and (vi) the time spent on the solving for the search direction (Newton step; percentage of time for overall solution of distance subproblem in brackets). The seventh, eighths and ninths column report run times for the kinetic energy subproblem. In particular, we report (vii) time spent on the setup of the KKT system, $(viii)$ time spent on solving the kinetic energy subproblem (percentage of TTS in brackets), and (ix) time spent on solving the KKT system (percentage of time for overall solution of kinetic energy subproblem in brackets).

run	iter	distance	TTS	distance subproblem		kinetic energy subproblem		
				solve (%)	newton (%)	setup	solve (%)	solve KKT (%)
1	25	0.7 (43)	87.6	50.9 (58)	50.9 (100)	1.8	36.5 (42)	34.8 (95)
2	12	0.6 (44)	52.0	32.5 (63)	32.5 (100)	1.7	19.4 (37)	18.7 (96)
3	70	0.7 (60)	153.7	92.3 (60)	92.3 (100)	1.5	61.0 (40)	57.4 (94)
4	49	0.7 (31)	158.7	109.9 (69)	109.8 (100)	1.5	48.5 (31)	46.0 (95)
5	46	0.7 (34)	136.7	81.6 (60)	81.6 (100)	1.7	54.8 (40)	52.0 (95)
6	5	0.5 (60)	19.6	12.4 (63)	12.4 (100)	1.4	7.2 (37)	6.9 (96)
7	5	0.5 (43)	24.9	14.9 (60)	14.9 (100)	1.7	10.0 (40)	9.7 (97)
8	23	0.4 (40)	114.4	86.5 (76)	86.5 (100)	1.5	27.7 (24)	26.0 (94)
9	13	0.6 (42)	47.8	31.4 (66)	31.3 (100)	1.4	16.4 (34)	15.7 (96)
10	45	0.6 (48)	81.1	51.1 (63)	51.1 (100)	1.2	29.7 (37)	27.8 (93)
11	6	0.6 (46)	26.6	18.0 (68)	18.0 (100)	1.4	8.5 (32)	8.2 (96)
12	67	0.7 (31)	264.0	157.5 (60)	157.4 (100)	2.0	106.2 (40)	101.4 (95)
13	11	0.5 (47)	50.7	31.7 (62)	31.6 (100)	1.8	18.9 (37)	18.3 (96)
14	14	0.5 (41)	67.6	41.1 (61)	41.1 (100)	1.8	26.4 (39)	25.4 (96)
15	33	0.7 (44)	174.1	103.1 (59)	103.0 (100)	2.2	70.8 (41)	67.7 (96)
16	51	0.7 (40)	230.7	129.2 (56)	129.2 (100)	2.4	101.2 (44)	96.9 (96)
17	9	0.5 (39)	58.7	41.5 (71)	41.5 (100)	1.7	17.2 (29)	16.6 (97)
18	58	0.6 (56)	204.4	115.2 (56)	115.2 (100)	1.9	88.9 (43)	84.7 (95)
19	41	0.7 (36)	176.6	102.0 (58)	101.9 (100)	2.1	74.4 (42)	71.4 (96)
20	24	0.5 (59)	75.2	47.5 (63)	47.5 (100)	1.5	27.5 (37)	26.2 (95)

Results We show a representative registration result in Fig. 7. The setup of this figure is exactly the same as for the results reported in Fig. 6. We refer to §3.2 for an explanation of the layout of this figure. The observations are similar to those we have made in §3.2.

In addition, we report performance results for 20 patients from our database in Table 4. For each patient, we report the following quantities of interest (from left to right; all times are reported in seconds): (i) The number of outer iterations of our splitting algorithm. (ii) The Hausdorff distance after diffeomorphic shape matching and the associated percentage of the original Hausdorff distance (in brackets). (iii) The time-to-solution (TTS; runtime of the solver). (iv) The overall time spent on solving the distance subproblem and the percentage this amounts to with respect to the time-to-solution (in brackets). (v) The overall time spent on computing the Newton steps during the solution of the distance subproblem. We also provide (in brackets) the percentage this computation amounts to with respect to the time needed to solve the distance subproblem. (vi) The time spent to setup the KKT system for the kinetic energy subproblem. (vii) The overall time spent on solving the kinetic energy system and the percentage this amounts to with respect to the time-to-solution (in brackets). (viii) The overall time spent on solving the KKT system and the percentage this amounts to with respect to the solution of the distance subproblem (in brackets).

The most important observations are: (i) We can see that we can reduce the distance by up to 60% compared to the original distance between the shape at $t = 0$ and $t = 1$. This gain is smaller than for the multi-patient case, but we also (generally) start with a smaller initial distance between the surfaces to be matched. This is also reflected by the smaller Hausdorff distance after diffeomorphic matching compared to the best results reported in Table 3. (ii) The time until convergence ranges from about 25 seconds up to about 250 seconds (depending on the dataset). (iii) On average, we spend about 60% of the total runtime on solving the distance subproblem and about 40% of the runtime on solving the kinetic energy subproblem. (iv) The runtime in each subproblem is dominated by solving the optimality conditions/computing the search direction; the percentages are on average at the order of almost 100% for the distance subproblem and about 95% for the kinetic energy subproblem.

4. Conclusions

We have presented and studied an improved matrix-free implementation of an operator splitting optimization algorithm for the solution of variational optimal control formulations for diffeomorphic shape matching. As opposed to many other works in the field, we consider a discretize-then-optimize approach to solve this problem. The associated infinite-dimensional variational problem is naturally recast as a finite-dimensional optimization problem in a RKHS. Our algorithmic improvements yield a speedup at the order of 2× compared to our prior work in [62]. Moreover, we have reduced the memory pressure since we no longer explicitly form and store matrix operators as they appear in the optimality conditions of the individual subproblems. Aside from introducing these numerical developments, we have reported new results for the registration of smooth 3D surfaces acquired from different subjects (multi-subject registration) and for time series of smooth 3D surfaces acquired from individual patients (multi-frame registration). In addition, we have surveyed our prior work in this area, which spans a period of more than one decade, with work touching on the mathematical problem formulation [56,57], algorithmic developments [57–62], clinical applications [65–67,170], as well as machine learning approaches for classification in shape space [46].

Our current and future work is targeted towards developing computational approaches for classification in shape space as well as improving computational throughput for algorithms for diffeomorphic shape matching and image registration. For the work described in the present paper we have identified several future research directions. An open problem is the tuning of hyper-parameters. The search for an adequate weight α for the shape distance can be performed based on a parameter continuation scheme [15,57,98]. However, an open question is how to most effectively combine this search with a multi-scale policy for selecting the parameters σ_v and σ_s, respectively. From a numerical perspective, several components of our solver can be improved. The most critical component is the implementation of the kernel matrices K. One possibility for improvements is to explore other, more efficient kernel representations [116]. An efficient implementation for the associated kernel operations is readily available in Python, and described in [180]. Moreover, although we have significantly reduced the memory footprint of our solver in the implementation described in this manuscript, we still store the kernel matrices and evaluate them using tensor algebra. Implementing more efficient kernel evaluations that are entirely matrix-free remains subject to future work. Lastly, our implementation is not deployed on any modern computing architectures. Porting our software to graphics-processing units may significantly improve the computational throughput [99,100].

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: The authors report financial support was provided by National Science Foundation (**NSF**) under the awards DMS-1854853, DMS-2012825, and DMS-2145845.

Data availability

Data will be made available on request.

Acknowledgements

This work was partly supported by the National Science Foundation (**NSF**) under the awards DMS-1854853, DMS-2012825, and DMS-2145845. Any opinions, findings, and conclusions or recommendations expressed herein are those of the authors and do not necessarily reflect the views of NSF. This work was completed in part with resources provided by the Research Computing Data Core at the University of Houston. We thank our collaborators Dr. William Zoghbi (MD) and Dr. Carlos El Tallawi (MD), from the DeBakey Heart and Vascular Center of the Houston Methodist Hospital for sharing and curating the annotated database of 3D echocardiology image sequences of the mitral valve acquired from a cohort of mitral valve patients according to local clinical protocols.

Our thanks also go out to our colleague, Dr. James Herring, and former PhD students, Drs. Jeff Friedman, Saurhab Jain, Aarti Jajoo, Yue Qin, and Peng Zhang, listed in alphabetical order. Their significant contributions over the course of this research project have been invaluable. We would like to convey our special gratitude to our esteemed late colleagues, Dr. Roland Glowinski and Dr. Ronald Hoppe. Their profound insights and advice, stemming from numerous discussions over the past two decades, have left an indelible mark on our work.

Appendix A. Hardware

Standard desktop computer: Mac Studio (Mac13,1) equipped with an Apple M1 Max 10-core CPU and 32 GB LPDDR5 memory. The code is implemented in Matlab (R2022b, Release 9.13.0.2105380).

Appendix B. Additional numerical results

We report additional results for six additional patients selected from our database for the experiments performed in §3.2. We refer to §3.2 for more details about the setup of these numerical experiments. We report the final Hausdorff distance, the norm (absolute and relative) of the primal and dual residual and the runtime for different choices of the scaling parameters τ_s and τ_v for the bandwidth of the Gaussian kernels used in our kernel distance and the parameterization in a RKHS in Table B.5, Table B.6, and Table B.7, respectively. Each table contains results for a different pair of patients. In addition, we illustrate the convergence behavior of our solver in Fig. B.8, Fig. B.9, and Fig. B.10 associated with a subset of these experiments. Again, each figure corresponds to a different patient. The layout for the tables as well as the figures is the same as in §3.2.

Table B.5
Performance for different choices of the scaling parameters τ_v and τ_s that control the bandwidth σ_v and σ_s of the RKHS parameterization and the kernel distance, respectively. The results concern the diffeomorphic matching of two mitral valve surfaces acquired at midsystole from two distinct patients (representative of our patients data base). The average mesh size $\bar{h}_1$ of the target shape is 1.2. We select τ_v in $\{3, 4, 6, 8\}$. We select τ_s in $\{3/4, 1, 2, 4, 6\}$. The initial (censored) Hausdorff distance for this dataset is 4.6. We report (from left to right) the final Hausdorff distance (after matching; absolute value and percentage of initial value in brackets), the (relative) norm of the primal residual, the (relative) norm of the dual residual, and the runtime of the solver (in seconds).

τ_v	τ_s	final distance (%)	residuals		runtime
			primal (relative)	dual (relative)	
3	0.75	2.3 (50)	1.3×10^{-2} (6.9×10^{-4})	1.0×10^{0} (5.3×10^{-2})	108
	1	2.2 (49)	6.4×10^{-3} (3.0×10^{-4})	8.0×10^{-1} (3.8×10^{-2})	89
	2	3.0 (67)	3.9×10^{-3} (1.9×10^{-4})	6.3×10^{-1} (3.0×10^{-2})	67
	4	4.1 (89)	3.0×10^{-3} (2.9×10^{-4})	5.0×10^{-1} (4.7×10^{-2})	64
	6	4.4 (96)	2.0×10^{-3} (4.2×10^{-4})	3.2×10^{-1} (6.7×10^{-2})	61
4	0.75	1.6 (36)	9.0×10^{-3} (4.7×10^{-4})	9.3×10^{-1} (4.8×10^{-2})	101
	1	1.8 (38)	6.4×10^{-3} (3.1×10^{-4})	8.2×10^{-1} (3.9×10^{-2})	82
	2	2.5 (54)	5.4×10^{-3} (2.6×10^{-4})	7.2×10^{-1} (3.4×10^{-2})	67
	4	3.8 (83)	4.2×10^{-3} (4.0×10^{-4})	5.9×10^{-1} (5.6×10^{-2})	62
	6	4.2 (92)	2.4×10^{-3} (5.0×10^{-4})	4.0×10^{-1} (8.3×10^{-2})	60
6	0.75	1.3 (29)	7.7×10^{-3} (4.0×10^{-4})	9.2×10^{-1} (4.8×10^{-2})	92
	1	1.4 (31)	6.7×10^{-3} (3.2×10^{-4})	8.5×10^{-1} (4.1×10^{-2})	79
	2	2.0 (45)	6.2×10^{-3} (2.9×10^{-4})	7.8×10^{-1} (3.7×10^{-2})	68
	4	3.6 (78)	4.8×10^{-3} (4.5×10^{-4})	7.0×10^{-1} (6.6×10^{-2})	58
	6	4.1 (89)	2.3×10^{-3} (4.7×10^{-4})	4.8×10^{-1} (1.0×10^{-1})	57
8	0.75	1.3 (30)	8.7×10^{-3} (4.5×10^{-4})	9.9×10^{-1} (5.2×10^{-2})	89
	1	1.4 (31)	7.6×10^{-3} (3.6×10^{-4})	9.5×10^{-1} (4.5×10^{-2})	80
	2	2.1 (45)	5.9×10^{-3} (2.8×10^{-4})	7.9×10^{-1} (3.8×10^{-2})	67
	4	3.5 (77)	4.6×10^{-3} (4.4×10^{-4})	7.0×10^{-1} (6.7×10^{-2})	58
	6	4.1 (89)	2.0×10^{-3} (4.1×10^{-4})	4.9×10^{-1} (1.0×10^{-1})	54

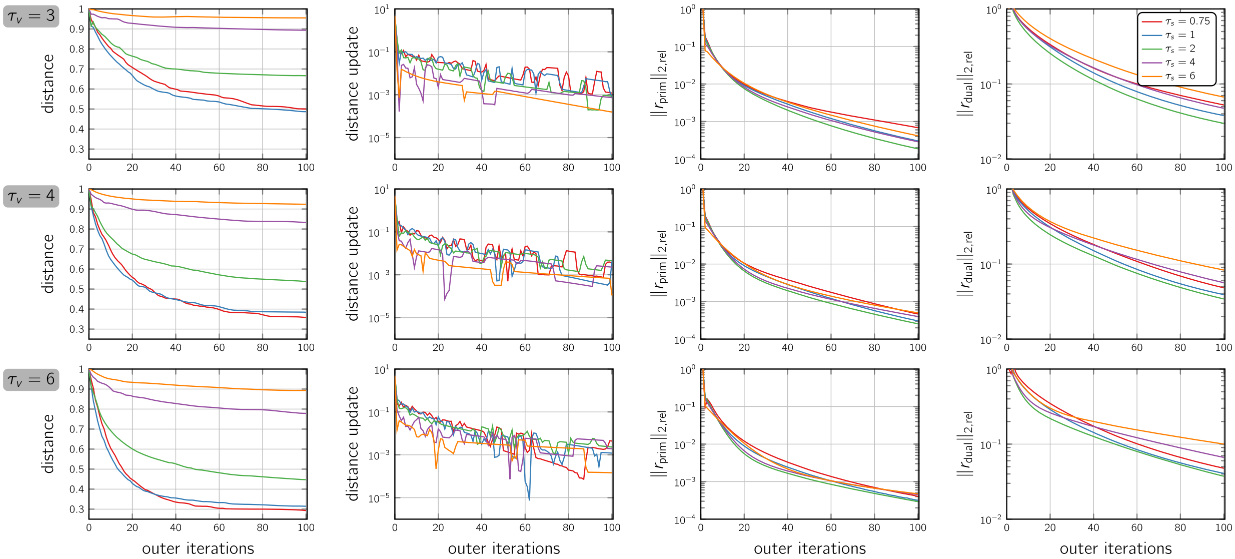

Fig. B.8. Performance for different choices of the scaling parameters τ_v and τ_s that control the bandwidth σ_v and σ_s of the RKHS parameterization and the kernel distance, respectively. The results are for two representative patients from our database. The average mesh size $\bar{h}_1$ of the target shape is 1.2. The results correspond to the first, second and third block in Table B.5. From top to bottom (row 1 through 3) we report results for $\tau_v = 3$, $\tau_v = 4$ and $\tau_v = 6$. Each plot shows results for $\tau_s \in \{3/4, 1, 2, 4, 6\}$ (see legend). The first column shows the trend of the Hausdorff distance with respect to the number of iterations (normalized with respect to the initial value). Notice that we do not minimize the Hausdorff distance in our optimization but the kernel distance in (20). The second column shows the update of the Hausdorff distance per iteration. The third and fourth column shows the norm of the primal and dual residual versus the iteration count. These residuals are normalized with respect to the residual obtained at iteration 2.

Table B.6
Performance for different choices of the scaling parameters τ_v and τ_s that control the bandwidth σ_v and σ_s of the RKHS parameterization and the kernel distance, respectively. The results are for two representative patients from our database. The average mesh size $\bar{h}_1$ of the target shape is 1.1. We select τ_v in $\{3, 4, 6, 8\}$. We select τ_s in $\{3/4, 1, 2, 4, 6\}$. The initial (censored) Hausdorff distance for this dataset is 3.2. We report (from left to right) the final Hausdorff distance (after matching; absolute value and percentage of initial value in brackets), the (relative) norm of the primal residual, the (relative) norm of the dual residual, and the runtime of the solver (in seconds).

τ_v	τ_s	final distance (%)	residuals		runtime
			primal (relative)	dual (relative)	
3	0.75	1.0 (30)	3.0×10^{-3} (1.9×10^{-4})	5.3×10^{-1} (3.3×10^{-2})	94
	1	1.1 (33)	2.7×10^{-3} (1.6×10^{-4})	5.1×10^{-1} (3.0×10^{-2})	81
	2	1.7 (52)	2.2×10^{-3} (1.3×10^{-4})	3.9×10^{-1} (2.4×10^{-2})	74
	4	2.5 (78)	2.5×10^{-3} (2.8×10^{-4})	3.5×10^{-1} (3.9×10^{-2})	66
	6	2.8 (85)	1.9×10^{-3} (4.4×10^{-4})	3.3×10^{-1} (7.6×10^{-2})	63
4	0.75	0.8 (26)	4.3×10^{-3} (2.7×10^{-4})	6.4×10^{-1} (4.0×10^{-2})	88
	1	0.9 (27)	3.8×10^{-3} (2.2×10^{-4})	6.0×10^{-1} (3.5×10^{-2})	79
	2	1.3 (42)	2.9×10^{-3} (1.8×10^{-4})	4.6×10^{-1} (2.8×10^{-2})	75
	4	2.2 (69)	3.1×10^{-3} (3.5×10^{-4})	3.9×10^{-1} (4.4×10^{-2})	65
	6	2.7 (82)	2.6×10^{-3} (6.0×10^{-4})	4.0×10^{-1} (9.4×10^{-2})	61
6	0.75	0.9 (28)	6.8×10^{-3} (4.2×10^{-4})	7.6×10^{-1} (4.7×10^{-2})	91
	1	0.9 (29)	4.9×10^{-3} (2.9×10^{-4})	6.9×10^{-1} (4.0×10^{-2})	81
	2	1.3 (40)	3.4×10^{-3} (2.1×10^{-4})	5.2×10^{-1} (3.2×10^{-2})	69
	4	2.0 (60)	3.4×10^{-3} (3.9×10^{-4})	4.4×10^{-1} (5.0×10^{-2})	61
	6	2.5 (76)	3.0×10^{-3} (7.0×10^{-4})	4.7×10^{-1} (1.1×10^{-1})	58
8	0.75	1.0 (31)	8.4×10^{-3} (5.2×10^{-4})	8.0×10^{-1} (5.0×10^{-2})	97
	1	1.1 (33)	6.4×10^{-3} (3.7×10^{-4})	7.4×10^{-1} (4.3×10^{-2})	84
	2	1.4 (42)	3.3×10^{-3} (2.0×10^{-4})	5.4×10^{-1} (3.3×10^{-2})	69
	4	1.9 (58)	3.3×10^{-3} (3.7×10^{-4})	4.5×10^{-1} (5.0×10^{-2})	61
	6	2.4 (73)	3.0×10^{-3} (7.0×10^{-4})	4.7×10^{-1} (1.1×10^{-1})	58

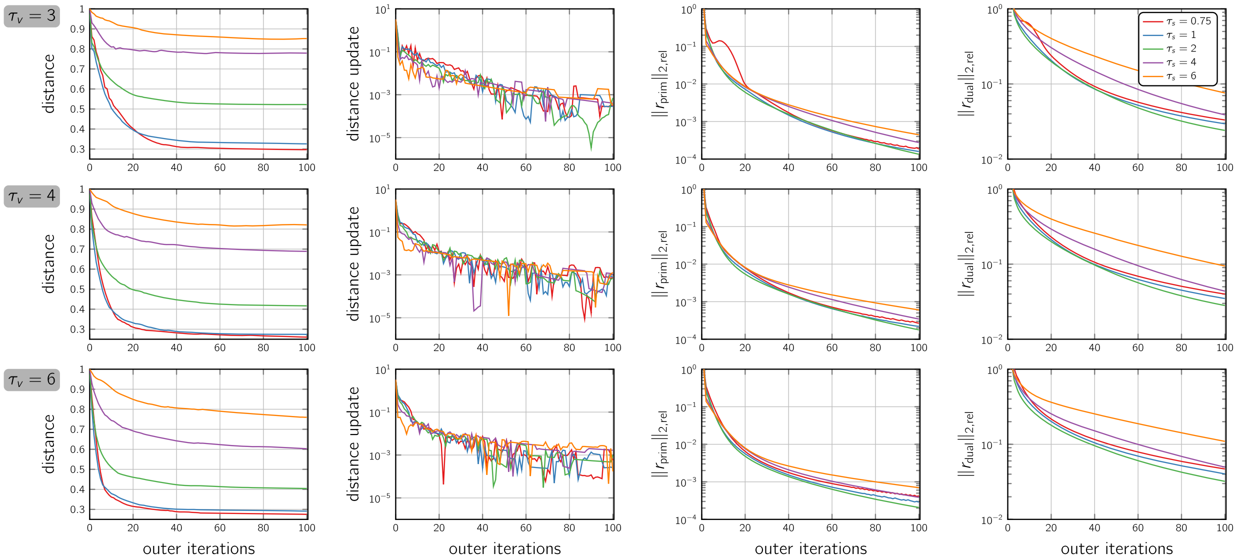

Fig. B.9. Performance for different choices of the scaling parameters τ_v and τ_s that control the bandwidth σ_v and σ_s of the RKHS parameterization and the kernel distance, respectively. The results are for two representative patients from our database. The average mesh size $\bar{h}_1$ of the target shape is 1.1. The results correspond to the first, second and third block in Table B.6. From top to bottom (row 1 through 3) we report results for $\tau_v = 3$, $\tau_v = 4$ and $\tau_v = 6$. Each plot shows results for $\tau_s \in \{1/4, 1, 2, 4, 6\}$ (see legend). The first column shows the trend of the Hausdorff distance with respect to the number of iterations (normalized with respect to the initial value). Notice that we do not minimize the Hausdorff distance in our optimization but the kernel distance in (20). The second column shows the update of the Hausdorff distance per iteration. The third and fourth column shows the norm of the primal and dual residual versus the iteration count. These residuals are normalized with respect to the residual obtained at iteration 2.

Table B.7
Performance for different choices of the scaling parameters τ_v and τ_s that control the bandwidth σ_v and σ_s of the RKHS parameterization and the kernel distance, respectively. The results are for two representative patients from our database. The average mesh size $\bar{h}_1$ of the target shape is 1.2. We select τ_v in $\{3, 4, 6, 8\}$. We select τ_s in $\{3/4, 1, 2, 4, 6\}$. The initial (censored) Hausdorff distance for this dataset is 5.6. We report (from left to right) the final Hausdorff distance (after matching; absolute value and percentage of initial value in brackets), the (relative) norm of the primal residual, the (relative) norm of the dual residual, and the runtime of the solver (in seconds).

τ_v	τ_s	final distance (%)	residuals		runtime
			primal (relative)	dual (relative)	
3	0.75	2.6 (47)	2.6×10^{-2} (8.4×10^{-4})	1.6×10^{0} (5.2×10^{-2})	109
	1	2.7 (47)	1.1×10^{-2} (3.3×10^{-4})	1.2×10^{0} (3.5×10^{-2})	93
	2	3.5 (62)	7.1×10^{-3} (2.3×10^{-4})	1.0×10^{0} (3.3×10^{-2})	72
	4	5.0 (90)	3.5×10^{-3} (3.7×10^{-4})	6.6×10^{-1} (7.0×10^{-2})	66
	6	5.6 (99)	2.1×10^{-3} (6.7×10^{-4})	3.2×10^{-1} (1.0×10^{-1})	61
4	0.75	1.9 (35)	1.4×10^{-2} (4.4×10^{-4})	1.3×10^{0} (4.1×10^{-2})	102
	1	2.1 (37)	9.6×10^{-3} (2.8×10^{-4})	1.1×10^{0} (3.2×10^{-2})	88
	2	3.1 (55)	8.2×10^{-3} (2.7×10^{-4})	1.0×10^{0} (3.4×10^{-2})	70
	4	4.8 (85)	4.5×10^{-3} (4.8×10^{-4})	8.7×10^{-1} (9.3×10^{-2})	61
	6	5.5 (98)	2.2×10^{-3} (7.1×10^{-4})	3.8×10^{-1} (1.2×10^{-1})	58
6	0.75	1.4 (25)	1.4×10^{-2} (4.4×10^{-4})	1.3×10^{0} (4.3×10^{-2})	93
	1	1.6 (29)	9.6×10^{-3} (2.8×10^{-4})	1.2×10^{0} (3.5×10^{-2})	85
	2	2.6 (46)	8.5×10^{-3} (2.8×10^{-4})	1.1×10^{0} (3.6×10^{-2})	70
	4	4.6 (82)	4.9×10^{-3} (5.3×10^{-4})	1.0×10^{0} (1.1×10^{-1})	57
	6	5.4 (97)	1.9×10^{-3} (6.2×10^{-4})	4.2×10^{-1} (1.3×10^{-1})	54
8	0.75	1.3 (24)	1.7×10^{-2} (5.5×10^{-4})	1.3×10^{0} (4.4×10^{-2})	93
	1	1.6 (28)	1.1×10^{-2} (3.3×10^{-4})	1.2×10^{0} (3.6×10^{-2})	85
	2	2.4 (44)	8.4×10^{-3} (2.8×10^{-4})	1.1×10^{0} (3.6×10^{-2})	68
	4	4.5 (81)	4.9×10^{-3} (5.2×10^{-4})	1.0×10^{0} (1.1×10^{-1})	57
	6	5.4 (96)	1.6×10^{-3} (5.2×10^{-4})	4.1×10^{-1} (1.3×10^{-1})	49

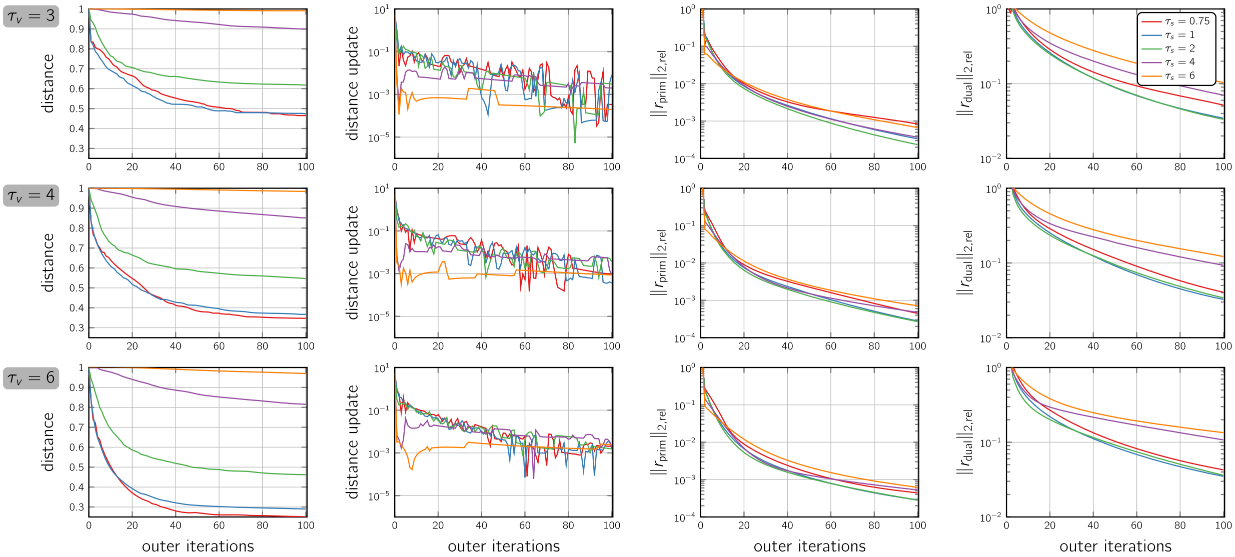

Fig. B.10. Performance for different choices of the scaling parameters τ_v and τ_s that control the bandwidth σ_v and σ_s of the RKHS parameterization and the kernel distance, respectively. The results are for two representative patients from our database. The average mesh size $\bar{h}_1$ of the target shape is 1.2. The results correspond to the first, second and third block in Table B.7. From top to bottom (row 1 through 3) we report results for $\tau_v = 3$, $\tau_v = 4$ and $\tau_v = 6$. Each plot shows results for $\tau_s \in \{0.75, 1, 2, 4, 6\}$ (see legend). The first column shows the trend of the Hausdorff distance with respect to the number of iterations (normalized with respect to the initial value). Notice that we do not minimize the Hausdorff distance in our optimization but the kernel distance in (20). The second column shows the update of the Hausdorff distance per iteration. The third and fourth column shows the norm of the primal and dual residual versus the iteration count. These residuals are normalized with respect to the residual obtained at iteration 2.

References

[1] J. Modersitzki, Numerical Methods for Image Registration, Oxford University Press, New York, 2004.

[2] J. Modersitzki, FAIR: Flexible Algorithms for Image Registration, SIAM, Philadelphia, Pennsylvania, US, 2009.

[3] B. Fischer, J. Modersitzki, Ill-posed medicine – an introduction to image registration, Inverse Probl. 24 (2008) 1–16.

[4] J.V. Hajnal, D.L.G. Hill, D.J. Hawkes (Eds.), Medical Image Registration, CRC Press, Boca Raton, Florida, US, 2001.

[5] A. Sotiras, C. Davatzikos, N. Paragios, Deformable medical image registration: a survey, IEEE Trans. Med. Imaging 32 (2013) 1153–1190.

[6] L. Younes, Shapes and Diffeomorphisms, vol. 171, 3 ed., Springer Verlag, Berlin Heidelberg, 2019.

[7] D.L.G. Hill, P.G. Batchelor, M. Holden, D.J. Hawkes, Medical image registration, Phys. Med. Biol. 46 (2001) R1.

[8] A.W. Toga, P.M. Thompson, The role of image registration in brain mapping, Image Vis. Comput. 19 (2001) 3–24.

[9] M. Burger, J. Modersitzki, L. Ruthotto, A hyperelastic regularization energy for image registration, SIAM J. Sci. Comput. 35 (2013) B132–B148.

[10] J. Kybic, M. Unser, Fast parametric elastic image registration, IEEE Trans. Image Process. 12 (2003) 1427–1442.

[11] C. Davatzikos, Spatial transformation and registration of brain images using elastically deformable models, Comput. Vis. Image Underst. 66 (1997) 207–222.

[12] E. Haber, J. Modersitzki, Numerical methods for volume preserving image registration, Inverse Probl. 20 (2004) 1621.

[13] J. Modersitzki, FLIRT with rigidity—image registration with a local non-rigidity penalty, Int. J. Comput. Vis. 76 (2008) 153–163.

[14] T. Mansi, X. Pennec, M. Sermesant, H. Delingette, N. Ayache, iLogDemons: a demons-based registration algorithm for tracking incompressible elastic biological tissues, Int. J. Comput. Vis. 92 (2011) 92–111.

[15] A. Mang, G. Biros, An inexact Newton–Krylov algorithm for constrained diffeomorphic image registration, SIAM J. Imaging Sci. 8 (2015) 1030–1069.

[16] A. Mang, G. Biros, Constrained H^1-regularization schemes for diffeomorphic image registration, SIAM J. Imaging Sci. 9 (2016) 1154–1194.

[17] A. Mang, J.A. Schnabel, W.R. Crum, M. Modat, O. Camara-Rey, C. Palm, G.B. Caseiras, H.R. Jäger, S. Ourselin, T.M. Buzug, D.J. Hawkes, Consistency of parametric registration in serial MRI studies of brain tumor progression, Int. J. Comput. Assisted Radiol. Surg. 3 (2008) 201–211.

[18] C. Misra, Y. Fan, C. Davatzikos, Baseline and longitudinal patterns of brain atrophy in MCI patients, and their use in prediction of short-term conversion to AD: results from ADNI, NeuroImage 44 (2009) 1415–1422.

[19] N. Guigui, X. Pennec, Parallel transport, a central tool in geometric statistics for computational anatomy: application to cardiac motion modeling, in: Handbook of Statistics, vol. 46, Elsevier, 2022, pp. 285–326.

[20] T. Makela, P. Clarysse, O. Sipila, N. Pauna, Q.C. Pham, T. Katila, I.E. Magnin, A review of cardiac image registration methods, IEEE Trans. Med. Imaging 21 (2002) 1011–1021.

[21] V. Gorbunova, J. Sporring, P. Lo, M. Loeve, H.A. Tiddens, M. Nielsen, A. Dirksen, M. de Bruijne, Mass preserving image registration for lung CT, Med. Image Anal. 16 (2012) 786–795.

[22] J. Ehrhardt, R. Werner, A. Schmidt-Richberg, H. Handels, Statistical modeling of 4D respiratory lung motion using diffeomorphic image registration, IEEE Trans. Med. Imaging 30 (2010) 251–265.

[23] G.E. Christensen, J.H. Song, W. Lu, I. El Naqa, D.A. Low, Tracking lung tissue motion and expansion/compression with inverse consistent image registration and spirometry, Med. Phys. 34 (2007) 2155–2163.

[24] S. Lee, N. Charon, B. Charlier, K. Popuri, E. Lebed, M.V. Sarunic, A. Trouvé, M.F. Beg, Atlas-based shape analysis and classification of retinal optical coherence tomography images using the functional shape (fshape) framework, Med. Image Anal. 35 (2017) 570–581.

[25] S. Joshi, B. Davis, M. Jomier, G. Gerig, Unbiased diffeomorphic atlas construction for computational anatomy, NeuroImage 23 (2004) S151–S160.

[26] K.M. Campbell, H. Dai, Z. Su, M. Bauer, P.T. Fletcher, S.C. Joshi, Structural connectome atlas construction in the space of Riemannian metrics, in: International Conference on Information Processing in Medical Imaging, Springer, 2021, pp. 291–303.

[27] M. Zhang, N. Singh, P.T. Fletcher, Bayesian estimation of regularization and atlas building in diffeomorphic image registration, in: International Conference on Information Processing in Medical Imaging, Springer, 2013, pp. 37–48.

[28] A. Serag, P. Aljabar, G. Ball, S.J. Counsell, J.P. Boardman, M.A. Rutherford, A.D. Edwards, J.V. Hajnal, D. Rueckert, Construction of a consistent high-definition spatio-temporal atlas of the developing brain using adaptive kernel regression, NeuroImage 59 (2012) 2255–2265.

[29] M. Cabezas, A. Oliver, X. Lladó, J. Freixenet, M.B. Cuadra, A review of atlas-based segmentation for magnetic resonance brain images, Comput. Methods Programs Biomed. 104 (2011) e158–e177.

[30] B.C. Vemuri, J. Ye, Y. Chen, C.M. Leonard, Image registration via level-set motion: applications to atlas-based segmentation, Med. Image Anal. 7 (2003) 1–20.

[31] T. Rohlfing, R. Brandt, R. Menzel, C.R. Maurer Jr, Evaluation of atlas selection strategies for atlas-based image segmentation with application to confocal microscopy images of bee brains, NeuroImage 21 (2004) 1428–1442.

[32] J. Ashburner, Computational anatomy with the SPM software, Magn. Reson. Imaging 27 (2009) 1163–1174.

[33] P.M. Thompson, A.W. Toga, A framework for computational anatomy, Comput. Vis. Sci. 5 (2002) 13–34.

[34] U. Grenander, M.I. Miller, Computational anatomy: an emerging discipline, Q. Appl. Math. 56 (1998) 617–694.

[35] M.I. Miller, Computational anatomy: shape, growth and atrophy comparison via diffeomorphisms, NeuroImage 23 (2004) S19–S33.

[36] M.I. Miller, A. Trouvé, L. Younes, On the metrics and Euler–Lagrange equations of computational anatomy, Annu. Rev. Biomed. Eng. 4 (2002) 375–405.

[37] L. Younes, F. Arrate, M.I. Miller, Evolutions equations in computational anatomy, NeuroImage 45 (2009) S40–S50.

[38] M.I. Miller, A. Trouvé, L. Younes, Hamiltonian systems and optimal control in computational anatomy: 100 years since D'Arcy Thompson, Annu. Rev. Biomed. Eng. 17 (2015).

[39] B.B. Avants, C.L. Epstein, M. Grossman, J.C. Gee, Symmetric diffeomorphic image registration with cross-correlation: evaluating automated labeling of elderly and neurodegenerative brain, Med. Image Anal. 12 (2008) 26–41.

[40] L. Risser, F.X. Vialard, R. Wolz, D.D. Holm, D. Rueckert, Simultaneous fine and coarse diffeomorphic registration: application to atrophy measurement in Alzheimer's disease, in: Proc Medical Image Computing and Computer-Assisted Intervention, in: LNCS, vol. 6362, 2010, pp. 610–617.

[41] L. Wang, F. Beg, T. Ratnanather, C. Ceritoglu, L. Younes, J.C. Morris, J.G. Csernansky, M.I. Miller, Large deformation diffeomorphism and momentum based hippocampal shape discrimination in dementia of the Alzheimer type, IEEE Trans. Med. Imaging 26 (2007) 462–470.

[42] N.C. Fox, P.A. Freeborough, Brain atrophy progression measured from registered serial MRI: validation and application to Alzheimer's disease, J. Magn. Reson. Imaging 7 (1997) 1069–1075.

[43] X. Hua, A.D. Loew, S. Lee, A.D. Klunder, A.W. Toga, N. Lepore, Y.Y. Chou, C. Brun, M.C. Chiang, M. Barysheva, C.R. Jack, M.A. Bernstein, P.J. Britson, C.P. Ward, J.L. Whitwell, B. Borowski, A.S. Fleisher, N.C. Fox, R.G. Boyes, J. Barnes, D. Harvey, J. Kornak, N. Schuff, L. Boreta, G.E. Alexander, M.W. Weiner, P.M. Thompson, ADNI, 3D characterization of brain atrophy in Alzheimer's disease and mild cognitive impairment using tensor-based morphometry, NeuroImage 41 (2008) 19–34.

[44] A. Joy, R. Nagarajan, E.S. Daar, J. Paul, A. Saucedo, S.K. Yadav, M. Guerrero, E. Haroon, P. Macey, M.A. Thomas, Alterations of gray and white matter volumes and cortical thickness in treated HIV-positive patients, Magn. Reson. Imaging 95 (2023) 27–38.

[45] C. Davatzikos, A. Genc, D. Xu, S.M. Resnick, Voxel-based morphometry using the RAVENS maps: methods and validation using simulated longitudinal atrophy, NeuroImage 14 (2001) 1361–1369.

[46] H. Dabirian, R. Sultamuratov, J. Herring, C.E. Tallawi, W. Zoghbi, A. Mang, R. Azencott, Automatic classification of deformable shapes, arXiv e-prints, arXiv:2211.02530, 2022.

[47] A. Trouvé, Diffeomorphism groups and pattern matching in image analysis, Int. J. Comput. Vis. 28 (1998) 213–221.
[48] A. Trouvé, A infinite dimensional group approach for physics based models in pattern recognition, Technical Report, Laboratoir d'Analyse Numerique CNRS URA, Universite Paris, 1995.
[49] J. Glaunes, A. Trouvé, L. Younes, Diffeomorphic matching of distributions: a new approach for unlabelled point-sets and sub-manifolds matching, in: Proc IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, 2004, pp. 712–718.
[50] J. Glaunès, A. Qiu, M.I. Miller, L. Younes, Large deformation diffeomorphic metric curve mapping, Int. J. Comput. Vis. 80 (2008) 317–336.
[51] M.F. Beg, M.I. Miller, A. Trouvé, L. Younes, Computing large deformation metric mappings via geodesic flows of diffeomorphisms, Int. J. Comput. Vis. 61 (2005) 139–157.
[52] S.C. Joshi, M.I. Miller, Landmark matching via large deformation diffeomorphisms, IEEE Trans. Image Process. 9 (2000) 1357–1370.
[53] V.I. Arnold, Sur la géométrie différentiell des gropues de Lie de dimension infine et ses applications al'hydrodynamique des fluides parfaits, Ann. Inst. Fourier 16 (1966) 319–361.
[54] V.I. Arnold, Les méthods mathématiques de la méchanique classique, MIR, Moscow, 1976.
[55] D.G. Ebin, J. Marsden, Groups of diffeomorphisms and the motion of an incompressible fluid, Ann. Math. (1970) 102–163.
[56] R. Azencott, R. Glowinski, A.M. Ramos, A controllability approach to shape identification, Appl. Math. Lett. 21 (2008) 861–865.
[57] R. Azencott, R. Glowinski, J. He, A. Jajoo, Y. Li, A. Martynenko, R.H.W. Hoppe, S. Benzekry, S.H. Little, Diffeomorphic matching and dynamic deformable surfaces in 3D medical imaging, Comput. Methods Appl. Math. 10 (2010) 235–274.
[58] J. Freeman, Combining diffeomorphic matching with image sequence intensity registration, Ph.D. thesis, University of Houston, 2014.
[59] A. Jajoo, Diffeomorphic matching and dynamic deformable shapes, Ph.D. thesis, University of Houston, 2011.
[60] Y. Qin, A second order variational approach for diffeomorphic matching of 3D surfaces, Ph.D. thesis, University of Houston, 2013.
[61] P. Zhang, Diffeomorphic shape matching based on an operator splitting method, Ph.D. thesis, University of Houston, 2019.
[62] P. Zhang, A. Mang, J. He, R. Azencott, K.C. El-Tallawi, W.A. Zoghbi, Diffeomorphic shape matching by operator splitting in 3D cardiology imaging, J. Optim. Theory Appl. 188 (2021) 143–168.
[63] S.B. Zekry, G. Lawrie, S. Little, W. Zoghbi, J. Freeman, A. Jajoo, S. Jain, J. He, A. Martynenko, R. Azencott, Comparative evaluation of mitral valve strain by deformation tracking in 3D-echocardiography, Cardiovasc. Eng. Technol. 3 (2012) 402–412.
[64] S.B. Zekry, J. Freeman, A. Jajoo, J. He, S.H. Little, G.M. Lawrie, R. Azencott, W.A. Zoghbi, Patient-specific quantitation of mitral valve strain by computer analysis of three-dimensional echocardiography: a pilot study, Circ. Cardiovasc. Imaging 9 (2016) e003254.
[65] S.B. Zekry, J. Freeman, A. Jajoo, J. He, S.H. Little, G.M. Lawrie, R. Azencott, W.A. Zoghbi, Effect of mitral valve repair on mitral valve leaflets strain: a pilot study, JACC Cardiovasc. Imag. 11 (2018) 776–777.
[66] K.C. El-Tallawi, P. Zhang, R. Azencott, J. He, E.L. Herrera, J. Xu, M. Chamsi-Pasha, J. Jacob, G.M. Lawrie, W.A. Zoghbi, Valve strain quantitation in normal mitral valves and mitral prolapse with variable degrees of regurgitation, JACC Cardiovasc. Imag. 14 (2021) 1099–1109.
[67] K.C. El-Tallawi, P. Zhang, R. Azencott, J. He, E. Herrera, M. Chamsi-Pasha, J. Jacob, G.M. Lawrie, W. Zoghbi, Quantitation of mitral valve strain in normals and in patients with mitral valve prolapse, J. Am. Coll. Cardiol. 73 (2019) 1953.
[68] T.A. Davis, Direct Methods for Sparse Linear Systems, SIAM, 2006.
[69] I.S. Duff, A.M. Erisman, J.K. Reid, Direct Methods for Sparse Matrices, Oxford University Press, 2017.
[70] J. Nocedal, S.J. Wright, Numerical Optimization, Springer, New York, New York, US, 2006.
[71] M. Bauer, M. Bruveris, P.W. Michor, Overview of the geometries of shape spaces and diffeomorphism groups, J. Math. Imaging Vis. 50 (2014) 60–97.
[72] C. von Tycowicz, F. Ambellan, A. Mukhopadhyay, S. Zachow, An efficient Riemannian statistical shape model using differential coordinates: with application to the classification of data from the osteoarthritis initiative, Med. Image Anal. 43 (2018) 1–9.
[73] D. Lüdke, T. Amiranashvili, F. Ambellan, I. Ezhov, B.H. Menze, S. Zachow, Landmark-free statistical shape modeling via neural flow deformations, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, 2022, pp. 453–463.
[74] T. Heimann, H.-P. Meinzer, Statistical shape models for 3D medical image segmentation: a review, Med. Image Anal. 13 (2009) 543–563.
[75] F. Ambellan, H. Lamecker, C.v. Tycowicz, S. Zachow, Statistical shape models: understanding and mastering variation in anatomy, in: Biomedical Visualisation, Springer, 2019, pp. 67–84.
[76] J.A. Iglesias, M. Rumpf, O. Scherzer, Shape-aware matching of implicit surfaces based on thin shell energies, Found. Comput. Math. 18 (2018) 891–927.
[77] M. Kilian, N.J. Mitra, H. Pottmann, Geometric modeling in shape space, in: ACM SIGGRAPH 2007 Papers, 2007, p. 64.
[78] R.S. Torres, A.X. Falcao, L.F. Costa, A graph-based approach for multiscale shape analysis, Pattern Recognit. 37 (2004) 1163–1174.
[79] A. Bône, O. Colliot, S. Durrleman, ADNI, learning the spatiotemporal variability in longitudinal shape data sets, Int. J. Comput. Vis. 128 (2020) 2873–2896.
[80] J.J. Junior, A.R. Backes, O.M. Bruno, Randomized neural network based descriptors for shape classification, Neurocomputing 312 (2018) 201–209.
[81] R.M. Rustamov, M. Ovsjanikov, O. Azencot, M. Ben-Chen, F. Chazal, L. Guibas, Map-based exploration of intrinsic shape differences and variability, ACM Trans. Graph. 32 (2013) 1–12.
[82] A. Srivastava, E. Klassen, S.H. Joshi, I.H. Jermyn, Shape analysis of elastic curves in Euclidean spaces, IEEE Trans. Pattern Anal. Mach. Intell. 33 (2010) 1415–1428.
[83] P.T. Fletcher, C. Lu, S.M. Pizer, S. Joshi, Principal geodesic analysis for the study of nonlinear statistics of shape, IEEE Trans. Med. Imaging 23 (2004) 995–1005.
[84] S. Arguillere, E. Trélat, A. Trouvé, L. Younes, Shape deformation analysis from the optimal control viewpoint, J. Math. Pures Appl. 104 (2015) 139–178.
[85] P. Dupuis, U. Gernander, M.I. Miller, Variational problems on flows of diffeomorphisms for image matching, Q. Appl. Math. 56 (1998) 587–600.
[86] I. Kaltenmark, B. Charlier, N. Charon, A general framework for curve and surface comparison and registration with oriented varifolds, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 3346–3355.
[87] B. O'Donghue, G. Stathopoulos, S. Boyd, A splitting method for optimal control, IEEE Trans. Control Syst. Technol. 21 (2013) 2432–2442.
[88] H. Bauschke, P. Combettes, Convex Analysis and Monotone Operator Theory in Hilbert Spaces, Springer-Verlag, New York, 2011.
[89] R. Glowinski, T.W. Pan, X.C. Tai, Some facts about operator-splitting and alternating direction methods, in: Splitting Methods in Communication, Imaging, Science, and Engineering, Springer International Publishing, 2016, pp. 19–94.
[90] R. Glowinski, S. Osher, W. Yin (Eds.), Splitting Methods in Communications, Imaging, Science, and Engineering, Springer International Publishing, 2016.
[91] R. Glowinski, S. Leung, J. Qian, Operator-splitting based fast sweeping methods for isotropic wave propagation in a moving floid, SIAM J. Sci. Comput. 38 (2016) A1195–A1223.
[92] M. Bukac, S. Canic, B. Muha, R. Glowinski, An operator splitting approach to the solution of fluid-structure interaction problems in hemodynamics, in: Splitting Methods in Communication, Imaging, Science, and Engineering, Springer International Publishing, 2016, pp. 731–772.
[93] R. Glowinski, A. Marroco, Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité d'une classe de problèmes de dirichlet non linéaires, Modél. Math. Anal. Numér. 9 (1975) 41–76.
[94] D. Gabay, B. Mercier, A dual algorithm for the solution of nonlinear variational problems via finite element approximation, Comput. Math. Appl. 2 (1976) 17–40.
[95] A. Mang, A. Gholami, G. Biros, Distributed-memory large-deformation diffeomorphic 3D image registration, in: Proc ACM/IEEE Conference on Supercomputing, 2016, pp. 842–853.

[96] A. Mang, L. Ruthotto, A Lagrangian Gauss–Newton–Krylov solver for mass- and intensity-preserving diffeomorphic image registration, SIAM J. Sci. Comput. 39 (2017) B860–B885.
[97] A. Mang, G. Biros, A semi-Lagrangian two-level preconditioned Newton–Krylov solver for constrained diffeomorphic image registration, SIAM J. Sci. Comput. 39 (2017) B1064–B1101.
[98] A. Mang, A. Gholami, C. Davatzikos, G. Biros, CLAIRE: a distributed-memory solver for constrained large deformation diffeomorphic image registration, SIAM J. Sci. Comput. 41 (2019) C548–C584.
[99] M. Brunn, N. Himthani, G. Biros, M. Mehl, A. Mang, Multi-node multi-GPU diffeomorphic image registration for large-scale imaging problems, in: Proc ACM/IEEE Conference on Supercomputing, 2020, pp. 523–539.
[100] M. Brunn, N. Himthani, G. Biros, M. Mehl, A. Mang, Fast GPU 3D diffeomorphic image registration, J. Parallel Distrib. Comput. 149 (2021) 149–162.
[101] M. Brunn, N. Himthani, G. Biros, M. Mehl, A. Mang, CLAIRE: constrained large deformation diffeomorphic image registration on parallel architectures, J. Open Sour. Softw. 6 (2021) 3038.
[102] N. Himthani, M. Brunn, J.Y. Kim, M. Schulte, A. Mang, G. Biros, CLAIRE: parallelized diffeomorphic image registration for large-scale biomedical imaging applications, J. Imaging 8 (2022) 251.
[103] Y. Cao, M.I. Miller, R.L. Winslow, L. Younes, Large deformation diffeomorphic metric mapping of vector fields, IEEE Trans. Med. Imaging 24 (2005) 1216–1230.
[104] D.-N. Hsieh, S. Arguillère, N. Charon, L. Younes, Mechanistic modeling of longitudinal shape changes: equations of motion and inverse problems, SIAM J. Appl. Dyn. Syst. 21 (2022) 80–101.
[105] T. Polzin, M. Niethammer, M.P. Heinrich, H. Handels, J. Modersitzki, Memory efficient LDDMM for lung CT, in: Proc. Medical Image Computing and Computer-Assisted Intervention, 2016, pp. 28–36.
[106] T. Polzin, M. Niethammer, F.-X. Vialard, J. Modersitzki, A discretize–optimize approach for LDDMM registration, in: Riemannian Geometric Statistics in Medical Image Analysis, Elsevier, 2020, pp. 479–532.
[107] F.-X. Vialard, L. Risser, D. Rueckert, C.J. Cotter, Diffeomorphic 3D image registration via geodesic shooting using an efficient adjoint calculation, Int. J. Comput. Vis. 97 (2012) 229–241.
[108] M. Niethammer, G.L. Hart, C. Zach, An optimal control approach for the registration of image time-series, in: Proceedings of the 48h IEEE Conference on Decision and Control, IEEE, 2009, pp. 2427–2434.
[109] G.L. Hart, C. Zach, M. Niethammer, An optimal control approach for deformable registration, in: Proc IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 9–16.
[110] S. Arguillere, E. Trélat, A. Trouvé, L. Younes, Registration of multiple shapes using constrained optimal control, SIAM J. Imaging Sci. 9 (2016) 344–385.
[111] M. Zhang, P.T. Fletcher, Finite-dimensional Lie algebras for fast diffeomorphic image registration, in: Proc Information Processing in Medical Imaging, vol. 24, 2015, pp. 249–259.
[112] J. Ashburner, K.J. Friston, Diffeomorphic registration using geodesic shooting and Gauss–Newton optimisation, NeuroImage 55 (2011) 954–967.
[113] M.I. Miller, A. Trouvé, L. Younes, Geodesic shooting for computational anatomy, J. Math. Imaging Vis. 24 (2006) 209–228.
[114] T. Polzin, Large deformation diffeomorphic metric mappings: Theory, numerics, and applications, Ph.D. thesis, University of Luebeck, 2018.
[115] N. Aronszajn, Theory of reproducing kernels, Trans. Am. Math. Soc. 68 (1950) 337–404.
[116] A. Jain, L. Younes, A kernel class allowing for fast computations in shape spaces induced by diffeomorphisms, J. Comput. Appl. Math. 245 (2013) 162–181.
[117] C.L. Richardson, L. Younes, Metamorphosis of images in reproducing kernel Hilbert spaces, Adv. Comput. Math. 42 (2016) 573–603.
[118] H.-W. Hsieh, N. Charon, Diffeomorphic registration with density changes for the analysis of imbalanced shapes, in: International Conference on Information Processing in Medical Imaging, Springer, 2021, pp. 31–42.
[119] H.-W. Hsieh, N. Charon, Diffeomorphic registration of discrete geometric distributions, in: Mathematics of Shapes and Applications, 2020, pp. 45–74.
[120] A. Thorley, X. Jia, H.J. Chang, B. Liu, K. Bunting, V. Stoll, A. de Marvao, D.P. O'Regan, G. Gkoutos, D. Kotecha, et al., Nesterov accelerated ADMM for fast diffeomorphic image registration, in: Medical Image Computing and Computer Assisted Intervention, 2021, pp. 150–160.
[121] Y.T. Lee, K.C. Lam, L.M. Lui, Landmark-matching transformation with large deformation via n-dimensional quasi-conformal maps, J. Sci. Comput. 67 (2016) 926–954.
[122] A. François, P. Gori, J. Glaunès, Metamorphic image registration using a semi-Lagrangian scheme, in: International Conference on Geometric Science of Information, Springer, 2021, pp. 781–788.
[123] A. Bône, M. Louis, B. Martin, S. Durrleman, Deformetrica 4: an open-source software for statistical shape analysis, in: International Workshop on Shape in Medical Imaging, Springer, 2018, pp. 3–13.
[124] E. Hartman, Y. Sukurdeep, E. Klassen, N. Charon, M. Bauer, Elastic shape analysis of surfaces with second-order Sobolev metrics: a comprehensive numerical framework, Int. J. Comput. Vis. (2023) 1–27.
[125] Z. Shen, J. Feydy, P. Liu, A.H. Curiale, R. San Jose Estepar, R. San Jose Estepar, M. Niethammer, Accurate point cloud registration with robust optimal transport, Adv. Neural Inf. Process. Syst. 34 (2021) 5373–5389.
[126] L. Tian, H. Greer, F.-X. Vialard, R. Kwitt, R.S.J. Estépar, M. Niethammer, GradICON: approximate diffeomorphisms via gradient inverse consistency, arXiv preprint, arXiv:2206.05897, 2022.
[127] B.B. Amor, S. Arguillère, L. Shao, ResNet-LDDMM: advancing the LDDMM framework using deep residual networks, arXiv preprint, arXiv:2102.07951, 2021.
[128] J. Krebs, H. Delingette, B. Mailhé, N. Ayache, T. Mansi, Learning a probabilistic model for diffeomorphic registration, IEEE Trans. Med. Imaging 38 (2019) 2165–2176.
[129] S. Sun, K. Han, D. Kong, H. Tang, X. Yan, X. Xie, Topology-preserving shape reconstruction and registration via neural diffeomorphic flow, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 20845–20855.
[130] X. Yang, R. Kwitt, M. Styner, M. Niethammer, Quicksilver: fast predictive image registration—a deep learning approach, NeuroImage 158 (2017) 378–396.
[131] N. Wu, M. Zhang, NeurEPDiff: neural operators to predict geodesics in deformation spaces, in: International Conference on Information Processing in Medical Imaging, Springer, 2023, pp. 588–600.
[132] S. Bharati, M. Mondal, P. Podder, V.B. Prasath, Deep learning for medical image registration: a comprehensive review, arXiv preprint, arXiv:2204.11341, 2022.
[133] H. Huang, B.B. Amor, X. Lin, F. Zhu, Y. Fang, Residual networks as flows of velocity fields for diffeomorphic time series alignment, arXiv preprint, arXiv:2106.11911, 2021.
[134] Y. Wu, T.Z. Jiahao, J. Wang, P.A. Yushkevich, M.A. Hsieh, J.C. Gee, NODEO: a neural ordinary differential equation based optimization framework for deformable image registration, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 20804–20813.
[135] H. Engl, M. Hanke, A. Neubauer, Regularization of Inverse Problems, Kluwer Academic Publishers, Dordrecht, NL, 1996.
[136] W.P. Ziemer, Weakly Differentiable Functions: Sobolev Spaces and Functions of Bounded Variation, vol. 120, Springer Science & Business Media, 2012.
[137] L. Younes, B. Gris, A. Trouvé, Sub–Riemannian methods in shape analysis, in: Handbook of Variational Methods for Nonlinear Geometric Data, 2020, pp. 463–495.

[138] M.D. Gunzburger, Perspectives in Flow Control and Optimization, SIAM, Philadelphia, Pennsylvania, US, 2003.
[139] B. Schölkopf, A.J. Smola, Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond, MIT Press, 2002.
[140] J. Meinguet, Multivariate interpolation at arbitrary points made simple, Z. Angew. Math. Phys. 30 (1979) 292–304.
[141] A. Quarteroni, R. Sacco, F. Saleri, Numerical Mathematics, Springer, 2010.
[142] M. Vaillant, J. Glaunes, Surface matching via currents, in: International Conference on Information Processing in Medical Imaging, Springer, 2005, pp. 381–392.
[143] P.-L. Antonsanti, J. Glaunès, T. Benseghir, V. Jugnon, I. Kaltenmark, Partial matching in the space of varifolds, in: International Conference on Information Processing in Medical Imaging, Springer, 2021, pp. 123–135.
[144] H.-W. Hsieh, N. Charon, Weight metamorphosis of varifolds and the LDDMM–Fisher–Rao metric, Calc. Var. Partial Differ. Equ. 61 (2022) 165.
[145] N. Charon, A. Trouvé, The varifold representation of nonoriented shapes for diffeomorphic registration, SIAM J. Imaging Sci. 6 (2013) 2547–2580.
[146] S. Boyd, L. Vandenberghe, Convex Optimization, Cambridge University Press, 2004.
[147] G. Biros, O. Ghattas, Parallel Lagrange-Newton-Krylov-Schur methods for PDE-constrained optimization—Part I: the Krylov-Schur solver, SIAM J. Sci. Comput. 27 (2005) 687–713.
[148] G. Biros, O. Ghattas, Parallel Lagrange-Newton-Krylov-Schur methods for PDE-constrained optimization—Part II: the Lagrange-Newton solver and its application to optimal control of steady viscous flows, SIAM J. Sci. Comput. 27 (2005) 714–739.
[149] L.T. Biegler, O. Ghattas, M. Heinkenschloss, B. van Bloemen Waanders, Large-Scale PDE-Constrained Optimization, Springer, 2003.
[150] L.T. Biegler, J. Nocedal, C. Schmid, A reduced Hessian method for large-scale constrained optimization, SIAM J. Optim. 5 (1995) 314–347.
[151] R. Herzog, K. Kunisch, Algorithms for PDE-constrained optimization, GAMM-Mitt. 33 (2010) 163–176.
[152] V. Akçelik, G. Biros, O. Ghattas, J. Hill, D. Keyes, B. van Bloemen Waanders, Parallel algorithms for PDE-constrained optimization, in: Parallel Processing for Scientific Computing, SIAM, 2006, pp. 291–322.
[153] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, Distributed optimization and statistical learning via the alternating direction method of multipliers, Found. Trends Mach. Learn. 1 (2011) 1–122.
[154] B. O'Donoghue, G. Stathopoulos, S. Boyd, A splitting method for optimal control, IEEE Trans. Control Syst. Technol. 21 (2013) 2432–2442.
[155] N. Parikh, S. Boyd, Proximal algorithms, Found. Trends Optim. 1 (2013) 123–231.
[156] J. Douglas, H.H. Rachford, On the numerical solution of heat conduction problems in two and three space variables, Trans. Am. Math. Soc. 82 (1956) 421–439.
[157] R. Glowinski, S.J. Osher, W. Yin, Splitting Methods in Communication, Imaging, Science, and Engineering, Springer, 2017.
[158] R. Glowinski, P. Tallec, Augmented Lagrangian and Operator-Splitting Methods in Nonlinear Mechanics, SIAM, Philadelphia, Pennsylvania, US, 1989.
[159] E.J. Dean, R. Glowinski, Operator-splitting methods for the simulation of Bingham visco-plastic flow, Chin. Ann. Math. 23 (2002) 187–204.
[160] M. Bukač, S. Čanić, R. Glowinski, B. Muha, A. Quaini, A modular, operator-splitting scheme for fluid–structure interaction problems with thick structures, Int. J. Numer. Methods Fluids 74 (2014) 577–604.
[161] L.-J. Deng, R. Glowinski, X.-C. Tai, A new operator splitting method for the Euler elastica model for image smoothing, SIAM J. Imaging Sci. 12 (2019) 1190–1230.
[162] D. Goldfarb, S. Ma, Fast multiple-splitting algorithms for convex optimization, SIAM J. Optim. 22 (2012) 533–556.
[163] N.J. Higham, Cholesky factorization, Wiley Interdiscip. Rev.: Comput. Stat. 1 (2009) 251–254.
[164] C. Ashcraft, R.G. Grimes, J.G. Lewis, Accurate symmetric indefinite linear equation solvers, SIAM J. Matrix Anal. Appl. 20 (1998) 513–561.
[165] S.C. Eisentat, H.F. Walker, Choosing the forcing terms in an inexact Newton method, SIAM J. Sci. Comput. 17 (1996) 16–32.
[166] P.C. Hansen, Discrete Inverse Problems: Insight and Algorithms, SIAM, Philadelphia, Pennsylvania, US, 2010.
[167] P.C. Hansen, Rank-Deficient and Discrete Ill-Posed Problems, SIAM Monographs on Mathematical Modeling and Computation, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, Philadelphia, Pennsylvania, US, 1998.
[168] C. Hansen, Analysis of discrete ill-posed problems by means of the L-curve, SIAM Rev. 34 (1992) 561–580.
[169] C.R. Vogel, Computational Methods for Inverse Problems, SIAM, Philadelphia, Pennsylvania, US, 2002.
[170] K.C. El-Tallawi, P. Zhang, R. Azencott, J. He, J. Xu, E.L. Herrera, J. Jacob, M. Chamsi-Pasha, G.M. Lawrie, W.A. Zoghbi, Mitral valve remodeling and strain in secondary mitral regurgitation: comparison with primary regurgitation and normal valves, JACC Cardiovasc. Imag. 14 (2021) 782–793.
[171] K. Scheufele, A. Mang, A. Gholami, C. Davatzikos, G. Biros, M. Mehl, Coupling brain-tumor biophysical models and diffeomorphic image registration, Comput. Methods Appl. Mech. Eng. 347 (2019) 533–567.
[172] K. Scheufele, S. Subramanian, A. Mang, G. Biros, M. Mehl, Image-driven biophysical tumor growth model calibration, SIAM J. Sci. Comput. 42 (2020) B549–B580.
[173] E.I. Zacharaki, C.S. Hogea, D. Shen, G. Biros, C. Davatzikos, Non-diffeomorphic registration of brain tumor images by simulating tissue loss and tumor growth, NeuroImage 46 (2009) 762–774.
[174] A. Gooya, K.M. Pohl, M. Bilello, L. Cirillo, G. Biros, E.R. Melhem, C. Davatzikos, GLISTR: glioma image segmentation and registration, IEEE Trans. Med. Imaging 31 (2013) 1941–1954.
[175] C. Hogea, C. Davatzikos, G. Biros, Brain–tumor interaction biophysical models for medical image registration, SIAM J. Sci. Comput. 30 (2008) 3050–3072.
[176] E.I. Zacharaki, S. Shen, S.K. Lee, C. Davatzikos, ORBIT: a multiresolution framework for deformable registration of brain tumor images, IEEE Trans. Med. Imaging 27 (2008) 1003–1017.
[177] X. Li, X. Long, P. Laurienti, C. Wyatt, Registration of images with varying topology using embedded maps, IEEE Trans. Med. Imaging 31 (2012) 749–765.
[178] A. François, M. Maillard, C. Oppenheim, J. Pallud, I. Bloch, P. Gori, J. Glaunès, Weighted metamorphosis for registration of images with different topologies, in: International Workshop on Biomedical Image Registration, Springer, 2022, pp. 8–17.
[179] Y. Sukurdeep, M. Bauer, N. Charon, A new variational model for shape graph registration with partial matching constraints, SIAM J. Imaging Sci. 15 (2022) 261–292.
[180] B. Charlier, J. Feydy, J.A. Glaunès, F.-D. Collin, G. Durif, Kernel operations on the GPU, with autodiff, without memory overflows, J. Mach. Learn. Res. 22 (2021) 1–6.