A second order variational approach for diffeomorphic matching of 3D surfaces

Y. QIN   J.W. HE   R. AZENCOTT

A Second Order Variational Approach For Diffeomorphic Matching Of 3D Surfaces

Y. Qin, J. He, and R. Azencott

Department of Mathematics, University of Houston, Houston, Texas 77204, USA

Abstract

In medical 3D-imaging, one of the main goals of image registration is to accurately compare two observed 3D-shapes. In this paper, we consider optimal matching of surfaces by a variational approach based on Hilbert spaces of diffeomorphic transformations. We first formulate, in an abstract setting, the optimal matching as an optimal control problem, where a vector field flow is sought to minimize a cost functional that consists of the kinetic energy and the matching quality. To make the problem computationally accessible, we then incorporate reproducing kernel Hilbert spaces with the Gaussian kernels and weighted sums of Dirac measures. We propose a second order method based the Bellman's optimality principle and develop a dynamic programming algorithm. We apply successfully the second order method to diffeomorphic matching of anterior leaflet and posterior leaflet snapshots. We obtain a quadratic convergence for data sets consisting of hundreds of points. To further enhance the computational efficiency for large data sets, we introduce new representations of shapes and develop a multi-scale method. Finally, we incorporate a stretching fraction in the cost function to explore the elastic model and provide a computationally feasible algorithm including the elasticity energy. The performance of the algorithm is illustrated by numerical results for examples from medical 3D-imaging of the mitral valve to reduce excessive contraction and stretching.

Keywords. diffeomorphic image matching, deformable shapes, reproducing kernel Hilbert spaces, Dirac measures, Newton descent, medical image analysis.

1 Introduction

As the medical imaging advances in the past few decades, 3D-imaging modalities, such as MRS (Magnetic Resonance Spectroscopy), PET (Positron Emission Tomograph), SPECT (Single Photon Emission Computed Tomograph) for functional information, and CT (Computed Tomography), MRI (Magnetic Resonance Imaging), Ultrasound Echography, and X-Ray, for anatomical visualization, have greatly increased the knowledge of normal and diseased anatomy and therefore been increasingly used to support clinical diagnosis and treatment planning. The growing size and number of these medical images have necessitated the use of computers to facilitate processing and analysis.

Image registration is a process of aligning two images acquired by same or different sensors, at different

times or from different viewpoint. It is desirable to compare or integrate the data obtained from two or more studies of the same patient. For instance, in a radiation therapy planning, a CT scan is needed for dose distribution calculations, while the contours of the target lesion are often best outlined on MRI (e.g., [79]). In [18], Brown gives a broad overview of image registration problems in various contexts. A comprehensive survey of image registration methods is presented by Barbara Zitova and Jan Flusser [80].

Medical image matching is a difficult task due to the distinct physical realities resulting from different imaging modalities, the differences in patient positioning, and varying acquisition techniques. To establish correspondence for a pair of images, it requires geometric transformation of one image into another. Given reference and target shapes S_0 and S_1, image matching is generally achieved by an transformation F such that $F(S_0) = S_1$. The most common transformations are rigid, affine, projective, perspective, and global ([18, 80]).

One method of computing transformations is known as the small deformations approach. Valid transformations are computed using linearized model via displacement vector fields when the images are separated by small deformations. However, the transformations computed are not guaranteed to be one-to-one or invertible. One limitation as shown in [23] is that the neighborhood structure could be destroyed in some cases when folding the grid over itself.

For the study of anatomy, it is essential to preserve properties such as smoothness of curves, surfaces, or other features associated to anatomy. It is a natural choice constraining the transformations to be diffeomorphisms, since they are smooth invertible transformations with smooth inverse.

An variational approach has been developed by G. Dupuis, J. Glaunès, U. Grenander, M. Miller, D. Mumford, A. Trouvé, and L. Younes [27, 36, 73] and subsequently explored in [10, 34, 55, 21, 32, 56] for comparisons of key anatomic parts of human brains such as the hippocampus, the temporal lobes, etc.

Within this framework, several gradient descent algorithms with respect to the landmark trajectories have been developed in [42, 10, 20]. The gradient method can be easily implemented, but it can be inefficient. On the other hand, it has been well received that Newton steps (cf., e.g., [17, 25]) are typically much more efficient than nonlinear conjugate gradient steps. Unfortunately, because of the numerical cost of computing and inverting the Hessian in large size problems, the second order optimization strategies have rarely been explored.

The main contribution of this paper is to develop Newton descent steps based on the Bellman's optimality principle and the second order information. To further improve the convergence, we properly discretize some of the components of the diffeomorphic machinery and open a way to well-defined and computationally robust multi-scale procedures. Moreover, to establish a more natural mathematical model, we incorporate an elasticity energy into the cost functional. For applications, the performance is illustrated by numerical deformation results from 3D echocardiographic data of the mitral valve.

2   Brief History

Motivated by the development of image acquisition methods (e.g., [61, 51, 58]) and segmentations algorithms (e.g., [62, 38]), the mathematical analysis of shapes has been a significant area of interest. For incompressible fluid obeying Euler equations, let $F_t(x)$ be the position of a fluid particle at time t starting at position x. Pioneers Arnold, Ebin, and Marsden showed (e.g., [3]) that the spatial displacement $F_t(x)$ between time 0 and t minimize the integral in time and space of the fluid kinetic energy. The continuous path defined by the time dependent $\mathbb{R}^3$-diffeomorphisms F_t is a geodesic $t \to F_t$ of an infinitely dimensional Lie group of $\mathbb{R}^3$-diffeomorphisms. In the past few decades, geodesics in groups of diffeomorphisms have provided a fertile framework for optimal matching of curves and surfaces (e.g.,

[27, 36, 73, 33, 54]).

Let U be a Hilbert subspace with strong Lipschitz-continuity in t, consisting of vector field flows $v : t \to v_t$, $0 \leq t \leq 1$, where v_t tends zero at infinity in $\mathbb{R}^3$. Consider

$$\partial_t F_t = v_t(F_t), \quad t \in (0, 1] \tag{1}$$

$$F_0 = \text{Id} \tag{2}$$

where Id is the identity map. G. Dupuis, U. Grenander, M. Miller, and A. Trouvé have shown in [27, 72], F^v generated by integration each time dependent flow $v = v_t$ between times 0 and 1 of the O.D.E. is a group of diffeomorphisms. Furthermore, for two smooth shapes S_0, S_1 with $k \in \{1, 2, 3\}$ in $\mathbb{R}^3$, as shown by M. Miller, L. Younes, and A. Trouvé in [56, 72, 55], $\inf \Delta_0^1 \|v_t\|_U dt$, the length of the shortest path connecting S_0 and S_1 defines a metric. With all these great contributions, the basic variational problem seeking minimizer in the velocity vector fields U to the cost functional

$$J(v) = \Delta_0^1 \|v_t\|_U^2 dt + \lambda \text{dis}(F(S_0), S_1), \qquad \text{for some fixed constant } \lambda > 0$$

is introduced in [10, 27, 33, 34]. In these papers, the regularization is achieved through replacement the rigid constraint $F(S_0) = S_1$ by a soft constraint using suitably chosen geometric surface matching distances, $\text{dis}(F(S_0), S_1)$. Several important metrics have been discussed in publication [76, 77, 50, 52, 41, 78].

In Berg et al. [10], the Euler-Lagrange equations are derived and the large deformation metric mappings algorithm is developed. Most importantly, they have shown that the metric distance between given shapes is computed by a geodesic path on the manifold of diffeomorphisms connecting the images. Following the geometric view outlined above, Glaunes et al. [33] have introduced weighted sums of Dirac measures to compare two arbitrary shapes which are considered as unlabeled landmarks. Both deformations fields and measures are modeled as linear combinations of kernels functions (e.g., [4]). By their synthetic experiments, measure matching demonstrates robustness against noise and outliers, or against different resamplings of the shapes. To incorporate both location and first order geometric structure, Glaunès et al. [32] represent curves as vector-valued measures and integrate curve matching into the variational framework. Numerical results from 2D and 3D curve mappings indicate better matching quality compared to landmark matching algorithms [42].

As discussed above, for the diffeomorphic matching of two static shapes S_0 and S_1, the variational approach has been intensively explored, and numerically implemented for quantified comparisons of key anatomic parts of human brains [10, 33]. Inspired by the general framework outlined above, Azencott et al. [6, 8] have extended the framework to finding an optimal matching for multiple sub-manifolds in $\mathbb{R}^3$. Given an arbitrary number of snapshots S_{t_j}, $j = 0, \cdots, q$ of a deforming object available at time instances $t_0 < t_1 \cdots < t_q$, an optimal time dependent $\mathbb{R}^3$-diffeomorphism F_t is obtained such that $F_{t_j}(S_{t_0}) = S_{t_j}$. Unlike [33], Gaussian kernels have been chosen for reproducing kernel Hilbert spaces [27]. It has been numerically implemented to the reconstruction of the deformations of the mitral valve apparatus.

For many shape matching applications, the gradient descent method is commonly used and produces good numerical results as seen in [6, 10, 33]. However, the convergence of the gradient descent method is slow. A fast algorithm is in great demand so that medical data can be processed in a timely manner. It is well known that Newton steps (e.g., [17, 25]) are typically much more efficient than nonlinear conjugate gradient steps. Our aim is to follow the variational approach in [33, 6] and develop improved numerical optimization strategies based on the Bellman's optimality principle [11] and the second order information.

3   Mathematical Background

In this section, we provide some mathematical tools needed and formulate the variational problem for dynamical matching.

3.1   Reproducing Kernel Hilbert Space

Reproducing kernel Hilbert spaces (RKHS) arise in a number of areas. The basic mathematical properties were studied by Moore (1935), Bergman (1950) and Aronszajn (1950). Aside from the shape-matching applications, RKHS have been found incredibly useful in other fields such as machine learning ([19, 67, 69]), statistical signal analysis ([75, 59]), image analysis ([63, 70, 49]), etc.

Definition 3.1. Let H be an inner product space ([39]) over $\mathbb{R}^d$ with the norm $\|\cdot\|_H$. If this space is complete, it is a *Hilbert space*.

With inner product $\langle\cdot,\cdot\rangle_H$ the associated norm is

$$\|h\|_H = \sqrt{\langle h, h\rangle_H}, \quad \text{for } h \in H.$$

A metric space H is complete if every Cauchy sequence in H converges in H. In other words, Hilbert spaces are Banach spaces endowed with a norm induced by an inner product. Now we refer to [4] for the definition of reproducing kernel Hilbert space.

Definition 3.2. Let H be a class of functions defined in $\mathbb{R}^d$, forming a Hilbert space. The function $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is called a *reproducing kernel* of H if

1. For every $x \in R^d$, the function K_x belongs to H. i.e.,

$$K_x(y) = K(y, x) \qquad \text{for all } y \in R^d. \tag{3}$$

2. The reproducing property: for every $x \in R^d$ and every $h \in H$,

$$h(x) = \langle h, K_x\rangle_H. \tag{4}$$

Definition 3.3. If there exists a reproducing kernel K on a Hilbert space H, then H is a *Reproducing Kernel Hilbert Space (RKHS)*.

A kernel $K(x, y)$ may be characterized as a function of two points according to [53] and it has several interesting properties. First, for $y \in \mathbb{R}^3$, applying (4) to $h = K_y$ yields

$$K(x, y) = K_y(x) = \langle K_y, K_x\rangle_H.$$

Since the last term is symmetric, we have

$$K(x, y) = K(y, x).$$

Additionally, for any $x \in R^d$ we have

$$\|K_x\|_H = \sqrt{\langle K_x, K_x\rangle_H} = \sqrt{K(x, x)}.$$

A second property is the fact that $K(x, y)$ is a positive matrix in the sense of E.H. Moore ([57]) shown in the theorem below.

Theorem 3.4. *Let* $\{x_1, x_2, \cdots, x_n\}$ *be a finite set in* $\mathbb{R}^d$, *and then the quadratic form in* $\xi_1, \xi_2, \cdots, \xi_n \in \mathbb{R}$,

$$\sum_{i,j=1}^{n} \xi_i \xi_j K(x_i, x_j), \tag{5}$$

is nonnegative and vanishes if and only if all ξ_i *equals 0.*

Proof. Apply the reproducing property (4) to the summation (5) and we have

$$\sum_{i,j=1}^{n} \xi_i \xi_j K(x_i, x_j) = \sum_{i,j=1}^{n} \xi_i \xi_j \langle K_{x_i}, K_{x_j} \rangle_H$$
$$= \langle \sum_{i=1}^{n} \xi_i K_{x_i}, \sum_{i=1}^{n} \xi_i K_{x_i} \rangle_H$$
$$= \| \sum_{i=1}^{n} \xi_i K_{x_i} \|_H^2 \geq 0.$$

If it vanishes, then $\sum_{i=1}^{n} \xi_i K_{x_i} = 0$. By the equation (4), for every $h \in H$,

$$\sum_{i=1}^{n} \xi_i h(x_i) = 0,$$

and therefore $\xi_1 = \cdots = \xi_n = 0$.
Additionally, it follows naturally that if $\xi_1 = \cdots = \xi_n = 0$, then

$$\sum_{i,j=1}^{n} \xi_i \xi_j K(x_i, x_j) = 0.$$

$\square$

We have defined a kernel function in terms of a reproducing kernel Hilbert space and discovered that the kernel is symmetric positive definite. Now we introduce the Moore-Aronszajn theorem to explore the converse direction: to every positive matrix $K(x, y)$ there corresponds one and only one class of functions with a uniquely determined quadratic form in it, forming a Hilbert space and admitting $K(x, y)$ as a reproducing kernel. The theorem was first brought up in [4] by Aronszajn although he attributes it to E. H. Moore.

Theorem 3.5. *(Moore-Aronszajn theorem) Let* $K : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$ *be a positive definite kernel. There is a unique reproducing kernel Hilbert space with reproducing kernel* K.

Proof. Let $H_0 = \text{span}\{K_x | x \in \mathbb{R}^d\}$ endowed with the inner product

$$\langle f, g \rangle_{H_0} = \sum_{i=1}^{n} \sum_{j=1}^{m} a_i b_j K(x_i, y_j), \tag{6}$$

where $f = \sum_{i=1}^{n} a_i K_{x_i}, a_i \in \mathbb{R}, \ i = 1, \cdots, n$, and $g = \sum_{j=1}^{m} b_j K_{y_j}, b_j \in \mathbb{R}, \ j = 1, \cdots, m$. We first have

to show (6) indeed defines a valid inner product. First it is independent of a_i, b_j used to define f, g since

$$\langle f, g \rangle_{H_0} = \langle \sum_{i=1}^{n} a_i K_{x_i}, g \rangle_{H_0} = \sum_{i=1}^{n} a_i \langle K_{x_i}, g \rangle_{H_0} = \sum_{i=1}^{n} a_i g(x_i),$$

and similarly

$$\langle f, g \rangle_{H_0} = \sum_{i=1}^{n} a_i g(x_i) = \sum_{j=1}^{m} b_j f(y_j).$$

Next for any $x \in \mathbb{R}^d$, by the Cauchy-Schwarz inequality

$$f(x) = \langle f, K_x \rangle_{H_0} \leq \|f\|_{H_0} K^{\frac{1}{2}}(x, x),$$

so if $\langle f, f \rangle_{H_0} = 0$, then $f = 0$.

Let H be the completion of H_0, i.e. for $h \in H$,

$$h = \sum_{i=1}^{\infty} a_i K_{x_i} \quad \text{where} \quad \sum_{i=1}^{\infty} a_i^2 K(x_i, x_i) < \infty.$$

Consider two Cauchy sequences $\{f_l\}, \{g_k\}$ in H_0 converging to $f, g \in H$ respectively and define the inner product in H as

$$\langle f, g \rangle_H = \lim_{l,k \to \infty} \langle f_l, g_k \rangle_{H_0}.$$

By the Cauchy-Schwarz inequality,

$$\langle f, g \rangle_H \leq \lim_{l,k \to \infty} \|f_l\|_{H_0} \|g_k\|_{H_0} < \infty,$$

and hence the inner product is well defined.

Now we may verify the equation (4), for $x \in \mathbb{R}^d$,

$$\langle f, K_x \rangle_H = \langle \lim_{l \to \infty} f_l, K_x \rangle_H = \lim_{l \to \infty} \langle f_l, K_x \rangle_{H_0} = \lim_{l \to \infty} f_l(x) = f(x).$$

As to the uniqueness, let G be another Hilbert space with K being a reproducing kernel. For any $x, y \in \mathbb{R}^d$,

$$\langle x, y \rangle_H = K(x, y) = \langle x, y \rangle_G,$$

and $\langle \cdot, \cdot \rangle_H = \langle \cdot, \cdot \rangle_G$ on the span of $\{K_x, x \in \mathbb{R}^d\}$. By the uniqueness of completion, $G = H$.
(One may refer to [60] for the proof on complex Hilbert spaces.) $\qquad \square$

There are quite a few well-known examples of kernels and RKHS in $\mathbb{R}^d$. Schoenberg shows in [66] that

$$K(x, y) = \exp(-\frac{\|x - y\|^p}{\sigma^2}), \quad x, y \in \mathbb{R}^d,$$

is positive definite if and only if $0 \leq p \leq 2$. When $p = 1$, we have the Laplacian kernel

$$K(x, y) = \exp(-a|x - y|), \quad a > 0. \tag{7}$$

6

The most popular kernel in practice is the Gaussian kernel when $p = 2$,

$$K(x, y) = \exp(-\frac{\|x - y\|^2}{\sigma^2}), \quad x, y \in \mathbb{R}^d. \tag{8}$$

For our research, we choose the Gaussian kernel. To begin with, the positive definite kernel K in our context is assumed to be bounded, smooth and invariant under translations and the Gaussian Kernel (8) provides more smoothing effects than the Laplacian kernel given in (7). Besides, it appears to be a good choice for diffeomorphic shape matching in [6, 36, 40].

3.2   Dynamic System

Definition 3.6. Given two manifolds X and Y, a map F from X to Y is called a *diffeomorphism* if it is a bijection (one-to-one correspondence) and both

$$F : X \to Y$$

and its inverse

$$F^{-1} : Y \to X$$

are differentiable.

For the study of anatomy, it is essential to preserve properties such as smoothness of curves, surfaces or other features associated to anatomy. It is a natural choice constraining the transformations to be diffeomorphisms, since they are smooth invertible transformations with smooth inverse.

Choose a Hilbert space U of smooth vector fields on $\mathbb{R}^d$ with norm $\| \cdot \|_U$ and consider the associated Hilbert space $L^2(I, U)$ of vector field flows where $I = [0, 1]$. Any time-dependent vector field flows: $v : t \mapsto v_t \in U, t \in I$ is associated the flow equation

$$\partial_t F_t = v_t(F_t), \quad t \in I, \tag{9}$$
$$F_0 = \text{Id}, \tag{10}$$

where Id refers to the identity map of $\mathbb{R}^d$.

It is shown in [27, 33] that the dynamic system has a unique solution when $t \mapsto \|v_t\|_U$ is integrable under suitable regularity condition on the elements of U. Thus we assume that the Hilbert space U of $\mathbb{R}^d$-vector fields is continuously embedded in a Sobolev space $W^{s,2}(\mathbb{R}^3)$ for some $s > 5/2$ and define the finite *kinetic energy* Kin(v) as

$$\text{Kin}(v) := \frac{1}{2}\|v\|^2_{L^2(I,U)} = \frac{1}{2}\Delta_0^1 \|v_t\|^2_U dt. \tag{11}$$

Theorem 3.7. *Assume $v \in L^2(I, U)$ where U is continuously embedded in $W^{s,2}(\mathbb{R}^3)$ for some $s > 5/2$. Then, the dynamic system (9),(10) admits a unique solution F_t with each F_t being an $\mathbb{R}^3$-diffeomorphism of smoothness class $1 \le r \le s - 3/2$.*

Proof. We refer to [27]. $\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad\square$

3.3 Distance of Two Shapes

It is common sense to calculate the distance of two points in the Euclidean space. However, comparing two curves or surfaces is much more complex. One commonly used distance is the classic Hausdorff distance ([65]).

Definition 3.8. Let $\mathbf{x}$ be a point and S be a non-empty set in $\mathbb{R}^d$, then,

$$d(\mathbf{x}, S) = \min_{(\mathbf{y} \in S)} |\mathbf{x} - \mathbf{y}|,$$

is *the distance of $\mathbf{x}$ to S*.

Definition 3.9. Let S and S' be two non-empty subsets in $\mathbb{R}^d$, we define the *Hausdorff distance $d_H(S, S')$* by

$$d_H(S, S') = \max\{\max_{x \in S} dist(x, S'), \max_{y \in S'} dist(y, S)\}.$$

Even though Hausdorff distances are quite useful in comparison of numerical results, in the variational framework, they introduce theoretical difficulties because $d_H(X, Y)$ is not smooth in general. To apply the gradient descent method to diffeomorphic matching problems, [6] introduces a global Hausdorff disparity, a smoothed version of the Hausdorff disparity. Several important metrics have been discussed in publication [76, 77, 50, 52, 41, 78].

Here we introduce a positive measure as used in many shape-matching applications (see [33, 54]). For submanifold S regularly embedded in $\mathbb{R}^3$, let μ be a bounded Borel measure induced on S. For two positive measures μ_1 and μ_2, define the Hilbert scalar product as

$$\langle \mu_1, \mu_2 \rangle_{\mathcal{H}} = \Delta \, \Delta \, K(x, y) d\mu_1 d\mu_2,$$

with norm

$$\|\mu\|_H^2 = \Delta \, \Delta \, K(x, y) d\mu d\mu,$$

where $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is a reproducing kernel. Thus the disparity of μ_1 and μ_2 is

$$\phi(\mu_1, \mu_2) = \|\mu_1 - \mu_2\|_{\mathcal{H}}^2.$$

In our context, we choose the Gaussian kernel in (8) for $K(x, y)$ with a scale parameter σ'. In section 6, we will discuss the choice of σ' more in detail.

For two shapes, the reference S_0 and the target S_1 in $\mathbb{R}^d$, the geometric disparity between $F_1^v(S_0)$ and S_T can be defined as

$$\phi(F_1^v) = \|F_1^v(\mu(S_0)) - (\mu(S_T)\|^2, \tag{12}$$

where F_1^v is the final diffeomorphism of $\mathbb{R}^d$ reached at time 1.

3.4 Variational Formulations

Following [7, 35] we define the penalized unconstrained function $J : L^2(I, U) \to \mathbb{R}$ using the kinetic energy $\mathrm{Kin}(v)$ in (11) and the disparity function $\phi(F_1^v)$ in (12),

$$J(v) := \mathrm{Kin}(v) + \lambda \phi(F_1^v), \quad v \in L^2(I, V), \tag{13}$$

8

where F_1^v is the solution of the ODE (9) with initial condition (10) and λ being large is a trade-off parameter.

Theorem 3.10. *The minimization problem*

$$\inf_{v \in L^2(I,U)} J(v),$$

associated with the dynamic system (9), (10) has a solution $v^* \in L^2(I,U)$.

Proof. We refer to [6]. $\qquad\square$

3.5 Application to Point Sets

Recall the dynamic system and denote $x_i(t)$ as the trajectory starting at x_i, $1 \le i \le N$. Then the ODE (9) and (10) can be translated as

$$\frac{dx_i(t)}{dt} = v_t(x_i(t)), \quad t \in (0,1], \tag{14}$$
$$x_i(0) = x_i, \tag{15}$$

for $i = 1, \cdots, N$. Take $V = V_K$ with a Gaussian kernel $K = K_\sigma$. It is shown in [20, 42] that the search for $v_t \in U$ of lowest energy is restricted to linear combinations of $K(x_i(t), \cdot)$, $i = 1, \cdots, N$, i.e.,

$$v_t(x) = \sum_{n=1}^{N} K_\sigma(x_i(t), x)\alpha_i(t), \quad \text{for any } x \in \mathbb{R}^d.$$

With control α_i introduced, we limit the search for optimal solution in a finite dimension space, $\mathbb{R}^N$ specifically.
By the definition of Hilbert norm, we have

$$\|v_t(x)\|_V^2 = \sum_{i=1}^{N} \sum_{j=1}^{N} K_\sigma(x_i(t), x_j(t))\alpha_i^T(t)\alpha_j(t).$$

With the kinetic energy taken care of, we look into the disparity function. For any piecewise smooth compact surfaces S in $\mathbb{R}^d$, let μ be a positive measure of S which can be approximated by linear combination of Dirac measures,

$$\mu = \sum_i c_i \delta_{x_i}, \tag{16}$$

where δ_{x_i} is the Dirac mass at nodes x_i. Naturally, for any diffeomorphism F of $\mathbb{R}^d$ acting on μ, we have

$$F\mu = \sum_i c_i \delta_{F(x_i)}. \tag{17}$$

Consider the reference shape $S_0 = \{x_i\}_{i=1}^N$ and the target shape $S_1 = \{y_j\}_{j=1}^M$. When weighted sums of the Dirac measures are used as in (16), let

$$\mu(S_0) = \sum_{i=1}^N a_i \delta_{x_i}, \quad \mu(S_1) = \sum_{j=1}^M b_j \delta_{y_j}.$$

Consider the diffeomorphism F_t^v associated with the trajectory $x_i(t)$. Then by (17) at $t = 1$,

$$\mu(F_1^v(S_0)) = \sum_{i=1}^N a_i \delta_{x_i(1)},$$

which represents the Borel distance between the shape $F_1^v(S_0)$ and S_1. Finally, associated with Gaussian kernel $K_{\sigma'}$ for some suitable scale parameter $\sigma' > 0$, the disparity function in (12) is,

$$
\begin{aligned}
\phi(x(1)) = \phi(F_1^v) &= \| \sum_{i=1}^N a_i \delta_{x_i(1)} - \sum_{j=1}^N b_j \delta_{y_j} \|_H^2 \\
&= \langle \sum_{i=1}^N a_i \delta_{x_i(1)} - \sum_{j=1}^N b_j \delta_{y_j}, \sum_{i=1}^N a_i \delta_{x_i(1)} - \sum_{j=1}^N b_j \delta_{y_j} \rangle_H \\
&= \langle \sum_{i=1}^N a_i \delta_{x_i(1)}, \sum_{i=1}^N a_i \delta_{x_i(1)} \rangle_H - 2 \langle \sum_{i=1}^N a_i \delta_{x_i(1)}, \sum_{j=1}^N b_j \delta_{y_j} \rangle_H + \langle \sum_{j=1}^N b_j \delta_{y_j}, \sum_{j=1}^N b_j \delta_{y_j} \rangle_H \\
&= \sum_{i=1}^N \sum_{j=1}^N a_i a_j K_{\sigma'}(x_i(1), x_j(1)) - 2 \sum_{i=1}^N \sum_{j=1}^M a_i b_j K_{\sigma'}(x_i(1), y_j) + \sum_{i=1}^M \sum_{j=1}^M b_i b_j K_{\sigma'}(y_i, y_j). \quad (18)
\end{aligned}
$$

3.6 Necessary Optimality Conditions

Now we introduce the matrix-vector notations and derive the necessary optimality conditions for the minimization problem. Consider the reference set $S_0 = \{x_i\}_{i=1}^N$ and the target set $S_1 = \{y_j\}_{j=1}^M$, for any $t \in (0, 1]$, denote

$$x^{(0)} = (x_1, \cdots, x_N)^T \in \mathbb{R}^{Nd}, \quad x(t) = (x_1(t), \cdots, x_N(t))^T \in \mathbb{R}^{Nd}, \quad (19)$$

$$\alpha(t) = (\alpha_1(t), \cdots, \alpha_N(t))^T \in \mathbb{R}^{Nd}, \quad (20)$$

$$A(x(t)) = (A_{ij}(x(t))) \in \mathbb{R}^{Nd \times Nd}, \quad (A_{ij}(x(t))) = K_\sigma(x_i(t), x_j(t)) I_d \in \mathbb{R}^{d \times d}. \quad (21)$$

It follows that the kinetic energy in (11) is now

$$\text{Kin}(v) = \frac{1}{2} \Delta_0^1 \alpha(t)^T A(x(t)) \alpha(t) dt. \quad (22)$$

Furthermore, the cost function in (13) takes the form

$$J(\alpha) = \frac{1}{2} \Delta_0^1 \alpha(t)^T A(x(t)) \alpha(t) dt + \lambda \phi(x(1)), \quad (23)$$

for $\alpha(t) \in L^2(I, \mathbb{R}^{Nd})$, where $\phi(x(1))$ is the measure disparity between $x(1)$ and target set S_1 given in (18).

The diffeomorphic matching problem is now transformed into the optimal control problem,

$$
\begin{aligned}
&\inf_{\alpha \in L^2(I, \mathbb{R}^{Nd})} J(\alpha), \\
\text{subject to} \quad &\frac{dx}{dt} = A(x(t))\alpha(t), \quad t \in I, \\
&x(0) = x^0.
\end{aligned}
\tag{24}
$$

Based on the theorem 3.10, there exists an optimal solution $\alpha^* \in L^2(I, \mathbb{R}^{Nd})$.

Theorem 3.11. *Assume that $\alpha^*(\cdot)$ is the solution of the optimal control problem (24), and that $x^*(\cdot)$ is the corresponding trajectory. Then there exists a function $p^*(\cdot)$, called the adjoint state, such that the triple (x^*, p^*, α^*) satisfies*

$$
\frac{dx^*(t)}{dt} = A(x^*(t))\, \alpha^*(t), \quad t \in (0,1],
\tag{25}
$$

$$
x^*(0) = x^{(0)},
\tag{26}
$$

$$
A(x^*(t))(\alpha^*(t) + p^*(t)) = 0, \qquad t \in (0,1].
\tag{27}
$$

$$
-\frac{dp^*(t)}{dt} = B(x^*(t), \alpha^*(t))^T (p^*(t) + \frac{1}{2}\alpha^*(t)), t \in (0,1],
\tag{28}
$$

$$
p^*(1) = \lambda \nabla \phi(x^*(1)),
\tag{29}
$$

where

$$
B(x^*(t), \alpha^*(t)) = \nabla_x \left(A(x^*(t))\, \alpha^*(t) \right),
$$

is given by

$$
B(x^*(t), \alpha^*(t)) = B_{ij}(x^*(t), \alpha^*(t)))_{n,m=1}^N \in \mathbb{R}^{Nd \times Nd},
$$

$$
B_{ij}(x^*(t), \alpha^*(t)) := \alpha_j^*(t)(\nabla_2 K_{\sigma_0}(x_i^*(t), x_j^*(t)))^T + \delta_{ij} \sum_{k=1}^N \alpha_k^*(t)(\nabla_1 K_{\sigma_0}(x_i^*(t), x_k^*(t)))^T.
$$

Proof. Letting $p(t) = (p_1(t), \cdots, p_N(t))^T \in \mathbb{R}^{Nd}$ be Lagrange multipliers ([17]), the Lagrangian associated with the optimal control problem (24) is

$$
\begin{aligned}
L(\alpha, x, p) &:= J(\alpha) - \Delta_0^1 p \cdot \left(\frac{dx(t)}{dt} - A(x(t))\alpha(t)\right) dt \\
&= -\Delta_0^1 p \cdot \frac{dx(t)}{dt} dt + \Delta_0^1 (p + \frac{1}{2}\alpha) \cdot A(x(t))\alpha(t) dt + \lambda \phi(x(1)).
\end{aligned}
$$

11

For (α^*, x^*, p^*) to be a critical point of $L(\alpha, x, p)$, the optimality conditions are

$$L_\alpha(\alpha^*, x^*, p^*) = 0, \tag{30}$$
$$L_x(\alpha^*, x^*, p^*) = 0, \tag{31}$$
$$L_p(\alpha^*, x^*, p^*) = 0. \tag{32}$$

(30) gives

$$A(x^*(t))p^* + A(x^*(t))\alpha^* = 0,$$

and therefore implies (27). Moreover, (32) indicates the dynamic system (25) and (26). As to condition (31), we first rewrite

$$-\Delta_0^1 \, p \cdot \frac{dx}{dt} dt = \Delta_0^1 \, \frac{dp}{dt} \cdot x dt - p(1) \cdot x(1) + p(0) \cdot x(0),$$

and it implies (28) and (29). □

4 The Continuous-time Nonlinear Optimal Control Problem

In this section, we will introduce the dynamic programming and derive continuous-time Riccati equations with the second order information for nonlinear optimal control problems. At the end, we briefly review the two different optimization approaches, optimize-then-discretize and discretize-then-optimize. For many shape-matching applications, the gradient descent method is commonly used and produces good numerical results as seen in [6, 10, 33]. However, clinicians and medical researchers, as natural users for automated 3D images registration, demands faster computing algorithms. It is well known that Newton steps (e.g., [17, 25]) are typically much more efficient than nonlinear conjugate gradient steps. Unfortunately, because of the numerical cost of computing and inverting Hessian matrices of the objective function in large size problems, the second order optimization strategies have rarely been used so far for diffeomorphic matching. Considering that we mostly deal with medium size problems, it shows great strength.

In the imaging field, dynamic programming has been used as in [47, 64, 31] and appears to be a fast and elegant method on finding the global solution. We also cooperate dynamic programming into the numerical algorithms so that the computing time increases linearly with respect to time steps and the errors stay controllable even for a longer time window.

4.1 Dynamic Programming

Dynamic programming ([28]) is originally brought up by Richard Bellman ([12]) and later refined to specifically referring to nesting smaller decision problems inside larger decisions. The crucial concept of the dynamic programming method is to break down a complex problem into a few consecutive overlapping subproblems, which often are really the same, and then combine the solutions of the subproblems to reach an overall solution. For a rigorous treatment on dynamic programming, see [13].

In continuous-time optimization problems, let us consider the controlled dynamics,

$$\frac{d\mathbf{x}}{dt} = \mathbf{F}(\mathbf{x}(t), \boldsymbol{\alpha}(t)), \quad t \in (0, 1],$$
$$\mathbf{x}(0) = \mathbf{x}^0,$$

(33)

with the associated cost functional (i.e., payoff function),

$$J(\boldsymbol{\alpha}(\cdot)) = \Delta_0^1 g(\mathbf{x}(t), \boldsymbol{\alpha}(t))dt + \phi(\mathbf{x}(1)),$$

(34)

where we call $g(\mathbf{x}(t), \boldsymbol{\alpha}(t))$ the *running payoff* and $\phi(\mathbf{x}(1))$ the *terminal payoff*.

Definition 4.1. For $\mathbf{x} \in \mathbb{R}^d$, $0 \le t \le 1$, define *the value function* $V(\mathbf{x}, t)$ to be the least payoff possible if we start at $\mathbf{x} \in \mathbb{R}^d$ at time t. That is,

$$V(\mathbf{x}, t) = \inf_{\boldsymbol{\alpha}(\cdot)} \left(\Delta_t^1 g(\mathbf{x}(s), \boldsymbol{\alpha}(s))ds + \phi(\mathbf{x}(1)) \right).$$

The value function $V(\mathbf{x}, t)$ represents the cost incurred from starting in state $\mathbf{x}$ at time t and controlling the system optimally from then until final time $t = 1$. The definition is in fact very natural, and by this definition we notice some very interesting facts.

First, the value function at final time $t = 1$ is essentially the terminal function, i.e., given $\mathbf{x} \in \mathbb{R}^d$,

$$V(\mathbf{x}, 1) = \inf_{\boldsymbol{\alpha}(\cdot)} \left(\Delta_1^1 g(\mathbf{x}(t), \boldsymbol{\alpha}(t))dt + \phi(\mathbf{x}) \right) = \phi(\mathbf{x}).$$

Also, at starting time $t = 0$,

$$V(\mathbf{x}, 0) = \inf_{\boldsymbol{\alpha}(\cdot)} J(\boldsymbol{\alpha}(\cdot)),$$

which reveals the original cost functional.

Theorem 4.2. *(HAMILTON-JACOB-BELLMAN EQUATION). For the dynamic system 33, assume that the value function V is a C^1 function of the variable (x, t). Then V solves the nonlinear partial differential equation*

$$V_t(x, t) + \min_a \{\mathbf{F}(x, a) \cdot \nabla_x V(x, t) + g(x, a)\} = 0,$$

(35)

with final state condition for $x \in \mathbb{R}^d$,

$$V(x, 1) = \phi(x).$$

Proof. Let $x \in \mathbb{R}^d$, $0 \le t < 1$ and $h > 0$ such that $t + h < 1$, and use the constant control $\boldsymbol{\alpha}(\cdot) = a$ for times $t \le s \le t + h$.

Now, consider the following dynamic system for times $t \le s \le t + h$,

$$\frac{d\mathbf{x}}{ds} = \mathbf{F}(\mathbf{x}(s), a), \quad s \in [t, t + h],$$
$$\mathbf{x}(t) = x,$$

(36)

where the dynamics starts at given x at time t and then arrives at the point $\mathbf{x}(t + h)$. Furthermore, starting at time $t + h$, we switch to an optimal control and use $V(\mathbf{x}(t + h), t + h)$ for the remaining times

13

$t + h \leq s \leq 1$. Combining those two time intervals, we should have the total payoff as

$$\Delta_t^{t+h} g(\mathbf{x}(s), a)ds + V(\mathbf{x}(t+h), t+h).$$

On the other hand, by definition 4.1, $V(x,t)$ is the least possible payoff if we start from (x,t). Therefore,

$$V(x,t) \leq \Delta_t^{t+h} g(\mathbf{x}(s), a)ds + V(\mathbf{x}(t+h), t+h). \tag{37}$$

To convert the inequality (37) into a differential form, we rearrange it and divide by h,

$$\frac{V(\mathbf{x}(t+h), t+h) - V(x,t)}{h} + \frac{1}{h} \Delta_t^{t+h} g(\mathbf{x}(s), a)ds \geq 0.$$

Let h go to 0, and then we have

$$\nabla_x V(x,t) \cdot \dot{\mathbf{x}}(t) + V_t(x,t) + g(x,a) \geq 0.$$

Besides, since $\mathbf{x}$ solve the differential equation system 36, we discover

$$\nabla_x V(x,t) \cdot \mathbf{F}(x,a) + V_t(x,t) + g(x,a) \geq 0.$$

The inequality holds for all control parameters, so

$$\min_a \{V_t(x,t) + g(x,a) + \nabla_x V(x,t) \cdot \mathbf{F}(x,a)\} \geq 0.$$

We now want to show the minimum above is actually equal to zero. Assume $\boldsymbol{\alpha}^*, \mathbf{x}^*(\cdot)$ are optimal for the minimization problem. Thus, for times $t \leq s \leq t + h$, the payoff is

$$\Delta_t^{t+h} g(\mathbf{x}^*(s), \boldsymbol{\alpha}^*(s))ds$$

and the remaining payoff for times $t + h \leq s \leq 1$ is $V(\mathbf{x}^*(t+h), t+h)$. Therefore, the total payoff is

$$V(x,t) = \Delta_t^{t+h} g(\mathbf{x}^*(s), \boldsymbol{\alpha}^*(s))ds + V(\mathbf{x}^*(t+h), t+h).$$

Again let us rearrange the equation and divide by h. Thus,

$$\frac{1}{h} \Delta_t^{t+h} g(\mathbf{x}^*(s), \boldsymbol{\alpha}^*(s))ds + \frac{V(\mathbf{x}^*(t+h), t+h) - V(\mathbf{x}^*, t)}{h} = 0.$$

Take h go to 0 and suppose $\boldsymbol{\alpha}^* = a^*$ for the time interval $t \leq s \leq t + h$, and then

$$g(x, a^*) + \nabla_x V(x,t) \cdot \mathbf{F}(x, a^*) + V_t(x,t) = 0,$$

for some parameter a^*. Therefore,

$$\min_a\{V_t(x,t) + g(x,a) + \nabla_x V(x,t) \cdot \mathbf{F}(x,a)\} = 0,$$

and consequently the Hamilton-Jacobi-Bellman equation is proved. □

As shown in [14], the HJB equation is a sufficient condition for an optimum. The solution of the HJB equation is the value function, which gives the optimal cost-to-go for the corresponding controlled dynamical system.

4.2 The Continuous-time Riccati Equations

To solve the nonlinear optimal control problem, we now seek for a value function that satisfies the HJB equation (35). Since the partial differential equation is highly nonlinear, we first introduce variations from nominal values,

$$\mathbf{x} = \bar{\mathbf{x}} + \delta\mathbf{x}, \qquad \boldsymbol{\alpha} = \bar{\boldsymbol{\alpha}} + \delta\boldsymbol{\alpha}.$$

Then we develop quadratic linear approximations to all the functions involved, that is,

$$\mathbf{F}(\mathbf{x}, \boldsymbol{\alpha}) = \frac{1}{2}\delta\mathbf{x}^T\mathbf{F}_{xx}\delta\mathbf{x} + \frac{1}{2}\delta\boldsymbol{\alpha}^T\mathbf{F}_{\alpha\alpha}\delta\boldsymbol{\alpha} + \delta\boldsymbol{\alpha}^T\mathbf{F}_{\alpha x}\delta\mathbf{x} + \mathbf{F}_x\delta\mathbf{x} + \mathbf{F}_\alpha\delta\boldsymbol{\alpha} + \mathbf{F}(\bar{\mathbf{x}}, \bar{\boldsymbol{\alpha}}),$$

$$g(\mathbf{x}, \boldsymbol{\alpha}) = \frac{1}{2}\delta\mathbf{x}^T\mathbf{g}_{xx}\delta\mathbf{x} + \frac{1}{2}\delta\boldsymbol{\alpha}^T\mathbf{g}_{\alpha\alpha}\delta\boldsymbol{\alpha} + \delta\boldsymbol{\alpha}^T\mathbf{g}_{\alpha x}\delta\mathbf{x} + \mathbf{g}_x^T\delta\mathbf{x} + \mathbf{g}_\alpha^T\delta\boldsymbol{\alpha} + g(\bar{\mathbf{x}}, \bar{\boldsymbol{\alpha}}),$$

$$V(\mathbf{x}, t) = \frac{1}{2}\delta\mathbf{x}^T\mathbf{P}(t)\delta\mathbf{x} + \mathbf{q}(t)^T\delta\mathbf{x} + \Theta(t),$$

where the Hessian is $\mathbf{P}(t) = V_{xx}$ and $\mathbf{q}(t) = V_x$. Thus,

$$V_t(\mathbf{x}, t) = \frac{1}{2}\delta\mathbf{x}^T\dot{\mathbf{P}}(t)\delta\mathbf{x} + \dot{\mathbf{q}}(t)^T\delta\mathbf{x} + \dot{\Theta}(t), \tag{38}$$

$$\nabla_x V(\mathbf{x}, t) = \mathbf{P}(t)\delta\mathbf{x} + \mathbf{q}(t). \tag{39}$$

Substitute all the terms above into equation (35), and we have

$$\frac{1}{2}\delta\mathbf{x}^T\dot{\mathbf{P}}(t)\delta\mathbf{x} + \dot{\mathbf{q}}(t)^T\delta\mathbf{x} + \dot{\Theta}(t)$$

$$+ \min_{\delta\boldsymbol{\alpha}}\{(\mathbf{P}(t)\delta\mathbf{x} + \mathbf{q}(t))^T(\frac{1}{2}\delta\mathbf{x}^T\mathbf{F}_{xx}\delta\mathbf{x} + \frac{1}{2}\delta\boldsymbol{\alpha}^T\mathbf{F}_{\alpha\alpha}\delta\boldsymbol{\alpha} + \delta\boldsymbol{\alpha}^T\mathbf{F}_{\alpha x}\delta\mathbf{x} + \mathbf{F}_x\delta\mathbf{x} + \mathbf{F}_\alpha\delta\boldsymbol{\alpha} + \mathbf{F}(\bar{\mathbf{x}}, \bar{\boldsymbol{\alpha}}))$$

$$+ (\frac{1}{2}\delta\mathbf{x}^T\mathbf{g}_{xx}\delta\mathbf{x} + \frac{1}{2}\delta\boldsymbol{\alpha}^T\mathbf{g}_{\alpha\alpha}\delta\boldsymbol{\alpha} + \delta\boldsymbol{\alpha}^T\mathbf{g}_{\alpha x}\delta\mathbf{x} + \mathbf{g}_x^T\delta\mathbf{x} + \mathbf{g}_\alpha^T\delta\boldsymbol{\alpha} + g(\bar{\mathbf{x}}, \bar{\boldsymbol{\alpha}}))\} = 0.$$

Keeping all quadratic linear terms, we obtain

$$\frac{1}{2}\delta\mathbf{x}^T\dot{\mathbf{P}}(t)\delta\mathbf{x} + \dot{\mathbf{q}}(t)^T\delta\mathbf{x} + \dot{\Theta}(t) + \min_{\delta\boldsymbol{\alpha}}\{\delta\mathbf{x}^T\mathbf{P}(t)(\mathbf{F}_x\delta\mathbf{x} + \mathbf{F}_\alpha\delta\boldsymbol{\alpha} + \mathbf{F}(\bar{\mathbf{x}}, \bar{\boldsymbol{\alpha}}))$$

$$+ \mathbf{q}(t)^T(\frac{1}{2}\delta\mathbf{x}^T\mathbf{F}_{xx}\delta\mathbf{x} + \frac{1}{2}\delta\boldsymbol{\alpha}^T\mathbf{F}_{\alpha\alpha}\delta\boldsymbol{\alpha} + \delta\boldsymbol{\alpha}^T\mathbf{F}_{\alpha x}\delta\mathbf{x} + \mathbf{F}_x\delta\mathbf{x} + \mathbf{F}_\alpha\delta\boldsymbol{\alpha} + \mathbf{F}(\bar{\mathbf{x}}, \bar{\boldsymbol{\alpha}}))$$

$$+ (\frac{1}{2}\delta\mathbf{x}^T\mathbf{g}_{xx}\delta\mathbf{x} + \frac{1}{2}\delta\boldsymbol{\alpha}^T\mathbf{g}_{\alpha\alpha}\delta\boldsymbol{\alpha} + \delta\boldsymbol{\alpha}^T\mathbf{g}_{\alpha x}\delta\mathbf{x} + \mathbf{g}_x^T\delta\mathbf{x} + \mathbf{g}_\alpha^T\delta\boldsymbol{\alpha} + g(\bar{\mathbf{x}}, \bar{\boldsymbol{\alpha}}))\} = 0. \tag{40}$$

To solve the equation above, we first need to attain the minimizer. It can be done by computing the gradient with respect to $\boldsymbol{\alpha}$. Thus, letting the gradient equal 0,

$$\mathbf{F}_\alpha^T \mathbf{P}(t)\delta\mathbf{x} + \mathbf{F}_{\alpha\alpha}\mathbf{q}(t)\delta\boldsymbol{\alpha} + \mathbf{F}_{\alpha x}\mathbf{q}(t)\delta\mathbf{x} + \mathbf{F}_\alpha^T\mathbf{q}(t) + \mathbf{g}_{\alpha\alpha}\delta\boldsymbol{\alpha} + \mathbf{g}_{\alpha x}\delta\mathbf{x} + \mathbf{g}_\alpha = 0.$$

Therefore,

$$\delta\boldsymbol{\alpha} = (\mathbf{g}_{\alpha\alpha} + \mathbf{F}_{\alpha\alpha}\mathbf{q}(t))^{-1}(\mathbf{g}_\alpha + \mathbf{F}_\alpha^T\mathbf{q}(t))$$

$$- (\mathbf{g}_{\alpha\alpha} + \mathbf{F}_{\alpha\alpha}\mathbf{q}(t))^{-1}(\mathbf{g}_{\alpha x} + \mathbf{F}_{\alpha x}\mathbf{q}(t) + \mathbf{F}_\alpha^T\mathbf{P}(t))\delta\mathbf{x}. \tag{41}$$

The equation (41) is crucial to dynamical programming and referred to as the feedback control law. For any variation of the state information $\mathbf{x}$, the optimal control $\boldsymbol{\alpha}$ can be correspondingly updated using the feedback control law.

Plug the equation above back to (40) and gather the terms of the same order. We have

$$\frac{1}{2}\delta\mathbf{x}^T(\dot{\mathbf{P}}(t) - (\mathbf{g}_{\alpha x} + \mathbf{F}_{\alpha x}\mathbf{q}(t) + \mathbf{F}_\alpha^T\mathbf{P}(t))^T(\mathbf{g}_{\alpha\alpha} + \mathbf{F}_{\alpha\alpha}\mathbf{q}(t))^{-1}(\mathbf{g}_{\alpha x} + \mathbf{F}_{\alpha x}\mathbf{q}(t) + \mathbf{F}_\alpha^T\mathbf{P}(t))$$

$$+ \mathbf{g}_{xx} + \mathbf{F}_{xx}\mathbf{q}(t) + 2\mathbf{F}_x^T\mathbf{P}(t))\delta\mathbf{x}$$

$$+ (\dot{\mathbf{q}}(t) - (\mathbf{g}_{\alpha x} + \mathbf{F}_{\alpha x}\mathbf{q}(t) + \mathbf{F}_\alpha^T\mathbf{P}(t))^T(\mathbf{g}_{\alpha\alpha} + \mathbf{F}_{\alpha\alpha}\mathbf{q}(t))^{-1}(\mathbf{g}_\alpha + \mathbf{F}_\alpha^T\mathbf{q}(t))$$

$$+ \mathbf{P}(t)\mathbf{F}(\bar{\mathbf{x}}, \bar{\boldsymbol{\alpha}}) + \mathbf{g}_x + \mathbf{F}_x^T\mathbf{q}(t))^T\delta\mathbf{x}$$

$$+ \dot{\Theta}(t) - \frac{1}{2}(\mathbf{g}_\alpha + \mathbf{F}_\alpha^T\mathbf{q}(t))^T(\mathbf{g}_{\alpha\alpha} + \mathbf{F}_{\alpha\alpha}\mathbf{q}(t))^{-1}(\mathbf{g}_\alpha + \mathbf{F}_\alpha^T\mathbf{q}(t)) + \mathbf{F}(\bar{\mathbf{x}}, \bar{\boldsymbol{\alpha}})\mathbf{q}(t) + g(\bar{\mathbf{x}}, \bar{\boldsymbol{\alpha}}) = 0.$$

Therefore, we obtain the Riccati equations as follows,

$$\dot{\mathbf{P}}(t) - (\mathbf{g}_{\alpha x} + \mathbf{F}_{\alpha x}\mathbf{q}(t) + \mathbf{F}_\alpha^T\mathbf{P}(t))^T(\mathbf{g}_{\alpha\alpha} + \mathbf{F}_{\alpha\alpha}\mathbf{q}(t))^{-1}(\mathbf{g}_{\alpha x} + \mathbf{F}_{\alpha x}\mathbf{q}(t) + \mathbf{F}_\alpha^T\mathbf{P}(t))$$

$$+ \mathbf{g}_{xx} + \mathbf{F}_{xx}\mathbf{q}(t) + 2\mathbf{F}_x^T\mathbf{P}(t) = 0, \tag{42}$$

$$\dot{\mathbf{q}}(t) - (\mathbf{g}_{\alpha x} + \mathbf{F}_{\alpha x}\mathbf{q}(t) + \mathbf{F}_\alpha^T\mathbf{P}(t))^T(\mathbf{g}_{\alpha\alpha} + \mathbf{F}_{\alpha\alpha}\mathbf{q}(t))^{-1}(\mathbf{g}_\alpha + \mathbf{F}_\alpha^T\mathbf{q}(t))$$

$$+ \mathbf{P}(t)\mathbf{F}(\bar{\mathbf{x}}, \bar{\boldsymbol{\alpha}}) + \mathbf{g}_x + \mathbf{F}_x^T\mathbf{q}(t) = 0, \tag{43}$$

$$\dot{\Theta}(t) - \frac{1}{2}(\mathbf{g}_\alpha + \mathbf{F}_\alpha^T\mathbf{q}(t))^T(\mathbf{g}_{\alpha\alpha} + \mathbf{F}_{\alpha\alpha}\mathbf{q}(t))^{-1}(\mathbf{g}_\alpha + \mathbf{F}_\alpha^T\mathbf{q}(t)) + \mathbf{F}(\bar{\mathbf{x}}, \bar{\boldsymbol{\alpha}})\mathbf{q}(t) + g(\bar{\mathbf{x}}, \bar{\boldsymbol{\alpha}}) = 0, \tag{44}$$

with final conditions,

$$\mathbf{P}(1) = \nabla_{xx}\phi(\mathbf{x}(1)),$$

$$\mathbf{q}(1) = \nabla_x\phi(\mathbf{x}(1)),$$

$$\Theta(1) = \phi(\mathbf{x}(1)).$$

Explicit Euler time discretization method can be applied to the system of the Riccati equations to reveal the value function backward in time.

4.3 Discretization in Time

We introduce time partition

$$0 = t_0 < t_1 < t_2 < \cdots < t_{L-1} < t_L = 1, \tag{45}$$

and define the step size accordingly $\tau_k = t_{k+1} - t_k$, $k = 0, 1, \cdots, L-1$. Also, notate

$$\mathbf{x}_k = \mathbf{x}(t_k), \quad \boldsymbol{\alpha}_k = \boldsymbol{\alpha}(t_k),$$

$$\mathbf{P}_k = \mathbf{P}(t_k), \quad \mathbf{q}_k = \mathbf{q}(t_k), \quad \Theta_k = \Theta(t_k),$$

$$\mathbf{F}^k = \mathbf{F}(\mathbf{x}_k, \boldsymbol{\alpha}_k), \quad g^k = g(\mathbf{x}, \boldsymbol{\alpha}).$$

Then, applying the Euler method to the Riccati equations (42)-(42), we have

$$\frac{\mathbf{P}_{k+1} - \mathbf{P}_k}{\tau_k} - (\mathbf{g}_{\alpha x}^k + \mathbf{F}_{\alpha x}^k \mathbf{q}_{k+1} + \mathbf{F}_\alpha^{kT} \mathbf{P}_k)^T (\mathbf{g}_{\alpha\alpha}^k + \mathbf{F}_{\alpha\alpha}^k \mathbf{q}_{k+1})^{-1} (\mathbf{g}_{\alpha x}^k + \mathbf{F}_{\alpha x}^k \mathbf{q}_{k+1} + \mathbf{F}_\alpha^{kT} \mathbf{P}_{k+1})$$

$$+ \mathbf{g}_{xx}^k + \mathbf{F}_{xx}^k \mathbf{q}_{k+1} + 2\mathbf{F}_x^{kT} \mathbf{P}_{k+1} = 0,$$

$$\frac{\mathbf{q}_{k+1} - \mathbf{q}_k}{\tau_k} - (\mathbf{g}_{\alpha x}^k + \mathbf{F}_{\alpha x}^k \mathbf{q}_{k+1} + \mathbf{F}_\alpha^{kT} \mathbf{P}_{k+1})^T (\mathbf{g}_{\alpha\alpha}^k + \mathbf{F}_{\alpha\alpha}^k \mathbf{q}_{k+1})^{-1} (\mathbf{g}_\alpha^k + \mathbf{F}_\alpha^{kT} \mathbf{q}_{k+1})$$

$$+ \mathbf{P}_{k+1} \mathbf{F}^k(\bar{\mathbf{x}}, \bar{\boldsymbol{\alpha}}) + \mathbf{g}_x^k + \mathbf{F}_x^{kT} \mathbf{q}_{k+1} = 0,$$

$$\frac{\Theta_{k+1} - \Theta_k}{\tau_k} - \frac{1}{2} (\mathbf{g}_\alpha^k + \mathbf{F}_\alpha^{kT} \mathbf{q}_{k+1})^T (\mathbf{g}_{\alpha\alpha}^k + \mathbf{F}_{\alpha\alpha}^k \mathbf{q}_{k+1})^{-1} (\mathbf{g}_\alpha^k + \mathbf{F}_\alpha^{kT} \mathbf{q}_{k+1})$$

$$+ \mathbf{F}^k(\bar{\mathbf{x}}, \bar{\boldsymbol{\alpha}}) \mathbf{q}_{k+1} + g^k(\bar{\mathbf{x}}, \bar{\boldsymbol{\alpha}}) = 0.$$

Therefore, the discrete Riccati equations are

$$\mathbf{P}_k = \mathbf{P}_{k+1} - \tau_k (\mathbf{g}_{\alpha x}^k + \mathbf{F}_{\alpha x}^k \mathbf{q}_{k+1} + \mathbf{F}_\alpha^{kT} \mathbf{P}_k)^T (\mathbf{g}_{\alpha\alpha}^k + \mathbf{F}_{\alpha\alpha}^k \mathbf{q}_{k+1})^{-1} (\mathbf{g}_{\alpha x}^k + \mathbf{F}_{\alpha x}^k \mathbf{q}_{k+1} + \mathbf{F}_\alpha^{kT} \mathbf{P}_{k+1})$$

$$+ \tau_k (\mathbf{g}_{xx}^k + \mathbf{F}_{xx}^k \mathbf{q}_{k+1} + 2\mathbf{F}_x^{kT} \mathbf{P}_{k+1}), \tag{46}$$

$$\mathbf{q}_k = \mathbf{q}_{k+1} - \tau_k (\mathbf{g}_{\alpha x}^k + \mathbf{F}_{\alpha x}^k \mathbf{q}_{k+1} + \mathbf{F}_\alpha^{kT} \mathbf{P}_{k+1})^T (\mathbf{g}_{\alpha\alpha}^k + \mathbf{F}_{\alpha\alpha}^k \mathbf{q}_{k+1})^{-1} (\mathbf{g}_\alpha^k + \mathbf{F}_\alpha^{kT} \mathbf{q}_{k+1})$$

$$+ \tau_k (\mathbf{P}_{k+1} \mathbf{F}^k(\bar{\mathbf{x}}, \bar{\boldsymbol{\alpha}}) + \mathbf{g}_x^k + \mathbf{F}_x^{kT} \mathbf{q}_{k+1}),$$

$$\Theta_k = \Theta_{k+1} - \frac{1}{2} \tau_k (\mathbf{g}_\alpha^k + \mathbf{F}_\alpha^{kT} \mathbf{q}_{k+1})^T (\mathbf{g}_{\alpha\alpha}^k + \mathbf{F}_{\alpha\alpha}^k \mathbf{q}_{k+1})^{-1} (\mathbf{g}_\alpha^k + \mathbf{F}_\alpha^{kT} \mathbf{q}_{k+1})$$

$$+ \tau_k (\mathbf{F}^k(\bar{\mathbf{x}}, \bar{\boldsymbol{\alpha}}) \mathbf{q}_{k+1} + g^k(\bar{\mathbf{x}}, \bar{\boldsymbol{\alpha}})),$$

with final conditions,

$$\mathbf{P}_L = \nabla_{xx}\phi(\mathbf{x}_L),$$

$$\mathbf{q}_L = \nabla_x\phi(\mathbf{x}_L),$$

$$\Theta_L = \phi(\mathbf{x}_L).$$

We will further compare the discrete Riccati equations from the continuous-time setting to those from solving the discrete nonlinear optimal control problem in section 5.

4.4 The Discretize-then-optimize Approach

There are two approaches to solve optimal control problems, optimize-then-discretize and discretize-then-optimize. The first approach is to find the necessary continuous optimality conditions analytically and then optimize the resulting equivalent system. This technique is employed in [15, 16].
For the latter approach, standard discretization techniques are used to transform the original problem into an optimization problem. Then, discrete optimality conditions are derived from the fully discretized optimization problem. This approach has been gaining more attention [1, 16, 43, 30].
In this thesis, we choose to follow the discretize-then-optimize approach. We present the discrete nonlinear optimal control problem and derive optimality conditions in the next chapter.

5 The Discrete Nonlinear Optimal Control Problem

In this section, we focus on developing the dynamic programming for discrete nonlinear optimal control problems. We first discretize the continuous problem using the explicit Euler method and introduce local quadratic approximation around nominal values. Following the Bellman's principle of optimality, the feedback control law will be carefully derived.

5.1 Time Discretization

The Bellman's principle usually refers to the dynamic system associated with discrete-time optimization problems and is considered as the discrete form of the HJB equation. Richard Bellman descries the principle of optimality in [11] as

Theorem 5.1. *Bellman's Principle of Optimality: An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision.*

For the time discretization of the optimal control problem (33), we introduce time partition

$$0 = t_0 < t_1 < t_2 < \cdots < t_{L-1} < t_L = 1, \tag{47}$$

and define the step size accordingly $\tau_k = t_{k+1} - t_k$, $k = 0, 1, \cdots, L - 1$. Also, notate $\mathbf{x}_k = \mathbf{x}(t_k)$ and $\boldsymbol{\alpha}_k = \boldsymbol{\alpha}(t_k)$. Discretize the ordinary equation (33) as

$$\mathbf{x}_{k+1} - \mathbf{x}_k = \tau_k \mathbf{F}(\mathbf{x}_k, \boldsymbol{\alpha}_k; t_k), \quad k = 0, \cdots, L - 1,$$

18

and for the matter of simplicity, introduce new notation

$$\mathbf{f}(\mathbf{x}_k, \boldsymbol{\alpha}_k; t_k) = \mathbf{x}_k + \tau_k \mathbf{F}(\mathbf{x}_k, \boldsymbol{\alpha}_k; t_k).$$

Now let us consider the generalized discrete nonlinear optimal control problem

$$\begin{aligned}
\mathbf{x}_{k+1} &= \mathbf{f}(\mathbf{x}_k, \boldsymbol{\alpha}_k; t_k) \qquad k = 0, \cdots, L-1, \\
\mathbf{x}_0 &= \bar{\mathbf{x}}_0,
\end{aligned} \tag{48}$$

with the cost function

$$J(\{\boldsymbol{\alpha}_k\}_{k=0}^{L-1}; \bar{\mathbf{x}}_0) = \sum_{k=0}^{L-1} G(\mathbf{x}_k, \boldsymbol{\alpha}_k; t_k) + \phi(\mathbf{x}_L), \tag{49}$$

where

$$G(\mathbf{x}_k, \boldsymbol{\alpha}_k; t_k) = \tau_k g(\mathbf{x}_k, \boldsymbol{\alpha}_k; t_k).$$

Thus, by the definition 4.1, the cost-to-go function at stage k is

$$J(\{\boldsymbol{\alpha}_l\}_{l=k}^{L-1}; \mathbf{x}_k) = \sum_{l=k}^{L-1} G(\mathbf{x}_l, \boldsymbol{\alpha}_l; t_l) + \phi(\mathbf{x}_L), \tag{50}$$

and the value function is

$$V_k(\mathbf{x}_k) = \min_{\{\boldsymbol{\alpha}_l\}_{l=k}^{L-1}} J(\{\boldsymbol{\alpha}_l\}_{l=k}^{L-1}; \mathbf{x}_k).$$

Extracting the running payoff at time t_k in (50),

$$J(\{\boldsymbol{\alpha}_l\}_{l=k}^{L-1}; \mathbf{x}_k) = G(\mathbf{x}_k, \boldsymbol{\alpha}_k; t_k) + \sum_{l=k+1}^{L-1} G(\mathbf{x}_l, \boldsymbol{\alpha}_l; t_l) + \phi(\mathbf{x}_L),$$

and we have

$$J(\{\boldsymbol{\alpha}_l\}_{l=k}^{L-1}; \mathbf{x}_k) = G(\mathbf{x}_k, \boldsymbol{\alpha}_k; t_k) + J(\{\boldsymbol{\alpha}_l\}_{l=k+1}^{L-1}; \mathbf{x}_{k+1}). \tag{51}$$

We may rewrite (51) as

$$V_k(\mathbf{x}_k) = \min_{\boldsymbol{\alpha}_k} \Big\{ G(\mathbf{x}_k, \boldsymbol{\alpha}_k; t_k) + \min_{\{\boldsymbol{\alpha}_l\}_{l=k+1}^{L-1}} J(\{\boldsymbol{\alpha}_l\}_{l=k+1}^{L-1}; \mathbf{x}_{k+1}) \Big\}.$$

By Bellman's principle of optimality, Theorem 5.1,

$$V_k(\mathbf{x}_k) = \min_{\boldsymbol{\alpha}_k} \big\{ G(\mathbf{x}_k, \boldsymbol{\alpha}_k; t_k) + V_{k+1}(\mathbf{x}_{k+1}) \big\}, \tag{52}$$

with final condition

$$V_{Nt}(\mathbf{x}_L) = \phi(\mathbf{x}_L). \tag{53}$$

Therefore, the cost function (49) is now reformed as optimal cost from state $\bar{\mathbf{x}}_0$ at time 0 which is

$$V_0(\bar{\mathbf{x}}_0) = \min_{\{\boldsymbol{\alpha}_k\}_{k=0}^{L-1}} J(\{\boldsymbol{\alpha}_k\}_{k=0}^{L-1}; \bar{\mathbf{x}}_0),$$

with respect to the dynamic system (48).

5.2 Quadratic Approximation

To complete the dynamic programming procedure, now we want to derive the quadratic approximation for both sides of equation (52) and reveal the feedback control law. Introduce variations from nominal values

$$\mathbf{x}_k = \bar{\mathbf{x}}_k + \delta\mathbf{x}_k, \quad \boldsymbol{\alpha}_k = \bar{\boldsymbol{\alpha}}_k + \delta\boldsymbol{\alpha}_k,$$

where

$$\bar{\mathbf{x}}_{k+1} = \mathbf{f}(\bar{\mathbf{x}}_k, \bar{\boldsymbol{\alpha}}_k; t_k).$$

Let $\boldsymbol{\alpha}_k = \boldsymbol{\alpha}_k^* + \delta\boldsymbol{\alpha}_k$, where $\boldsymbol{\alpha}_k^*$ the optimal solution to the value function at t_k,

$$V_k(\bar{\mathbf{x}}_k) = \min_{\boldsymbol{\alpha}_k}\{G(\bar{\mathbf{x}}_k, \boldsymbol{\alpha}_k; t_k) + V_{k+1}(\mathbf{x}_{k+1})\}. \tag{54}$$

In order to obtain the update $\delta\boldsymbol{\alpha}_k$, we first develop linear quadratic expansions at $(\bar{\mathbf{x}}_k, \boldsymbol{\alpha}_k^*)$. In equation (54), there are two functions involved, $V_k(\bar{\mathbf{x}}_k)$ and $G(\bar{\mathbf{x}}_k, \boldsymbol{\alpha}_k; t_k)$. They are approximated as follows.

$$\text{QP}\big[V_k(\bar{\mathbf{x}}_k + \delta\mathbf{x}_k)\big] = \frac{1}{2}\delta\mathbf{x}_k^T \mathbf{V}_{xx}^k \delta\mathbf{x}_k + \mathbf{V}_x^{kT}\delta\mathbf{x}_k + \Theta_k + \bar{V}_k(\bar{\mathbf{x}}_k),$$

$$\text{QP}\big[G(\bar{\mathbf{x}}_k + \delta\mathbf{x}_k, \boldsymbol{\alpha}_k^* + \delta\boldsymbol{\alpha}_k; t_k)\big] = \frac{1}{2}\delta\mathbf{x}_k^T \mathbf{G}_{xx}^k \delta\mathbf{x}_k + \frac{1}{2}\delta\boldsymbol{\alpha}_k^T \mathbf{G}_{\alpha\alpha}^k \delta\boldsymbol{\alpha}_k + \delta\boldsymbol{\alpha}_k^T \mathbf{G}_{\alpha x}^k \delta\mathbf{x}_k$$

$$+ \mathbf{G}_x^{kT}\delta\mathbf{x}_k + \mathbf{G}_\alpha^{kT}\delta\boldsymbol{\alpha}_k + \Delta G_k + G(\bar{\mathbf{x}}_k, \bar{\boldsymbol{\alpha}}_k; t_k),$$

where

$$\Delta G_k = G(\bar{\mathbf{x}}_k, \boldsymbol{\alpha}_k^*; t_k) - G(\bar{\mathbf{x}}_k, \bar{\boldsymbol{\alpha}}_k; t_k),$$

$$\bar{V}_k(\bar{\mathbf{x}}_k) = J(\{\bar{\boldsymbol{\alpha}}_l\}_{l=k}^{N-1}; \bar{\mathbf{x}}_k), \tag{55}$$

$$\Theta_k = V_k(\bar{\mathbf{x}}_k) - \bar{V}_k(\bar{\mathbf{x}}_k). \tag{56}$$

Furthermore, expand $V_{k+1}(\mathbf{x}_{k+1})$ at $(\bar{\mathbf{x}}_{k+1}, \boldsymbol{\alpha}_{k+1}^*)$ which is

$$\text{QP}\big[V_{k+1}(\bar{\mathbf{x}}_{k+1} + \delta\mathbf{x}_{k+1})\big] = \frac{1}{2}\delta\mathbf{x}_{k+1}^T \mathbf{V}_{xx}^{k+1} \delta\mathbf{x}_{k+1} + \mathbf{V}_x^{k+1T}\delta\mathbf{x}_{k+1} + \Theta_{k+1} + \bar{V}_{k+1}(\bar{\mathbf{x}}_{k+1}),$$

where Θ_{k+1} and $\bar{V}_{k+1}(\bar{\mathbf{x}}_{k+1})$ follow the notation in (55), (56) respectively. With linear quadratic expansions, equation (52) is equivalent to the statement below.

$$\text{QP}\big[V_k(\bar{\mathbf{x}}_k + \delta\mathbf{x}_k)\big] \approx \min_{\delta\boldsymbol{\alpha}_k}\{\text{QP}\big[G(\bar{\mathbf{x}}_k + \delta\mathbf{x}_k, \boldsymbol{\alpha}_k^* + \delta\boldsymbol{\alpha}_k; t_k) + V_{k+1}(\bar{\mathbf{x}}_{k+1} + \delta\mathbf{x}_{k+1})\big]\},$$

20

which is organized as

$$\frac{1}{2}\delta\mathbf{x}_k^T \mathbf{V}_{xx}^k \delta\mathbf{x}_k + \mathbf{V}_x^{kT}\delta\mathbf{x}_k + \Theta_k + \bar{V}_k(\bar{\mathbf{x}}_k)$$

$$\approx \min_{\delta\boldsymbol{\alpha}_k}\left\{\frac{1}{2}\delta\mathbf{x}_k^T \mathbf{G}_{xx}^k \delta\mathbf{x}_k + \frac{1}{2}\delta\boldsymbol{\alpha}_k^T \mathbf{G}_{\alpha\alpha}^k \delta\boldsymbol{\alpha}_k + \delta\boldsymbol{\alpha}_k^T \mathbf{G}_{\alpha x}^k \delta\mathbf{x}_k \right.$$

$$+ \mathbf{G}_x^{kT}\delta\mathbf{x}_k + \mathbf{G}_\alpha^{kT}\delta\boldsymbol{\alpha}_k + \Delta G_k + G(\bar{\mathbf{x}}_k, \bar{\boldsymbol{\alpha}}_k; t_k)$$

$$\left. + \frac{1}{2}\delta\mathbf{x}_{k+1}^T \mathbf{V}_{xx}^{k+1}\delta\mathbf{x}_{k+1} + \mathbf{V}_x^{k+1T}\delta\mathbf{x}_{k+1} + \Theta_{k+1} + \bar{V}_{k+1}(\bar{\mathbf{x}}_{k+1})\right\}.$$

Since we need to minimize the objective function over $\delta\boldsymbol{\alpha}_k$, we want to substitute $\delta\mathbf{x}_{k+1}$ with information at time k. Consider

$$\delta\mathbf{x}_{k+1} = \mathbf{x}_{k+1} - \bar{\mathbf{x}}_{k+1} = \mathbf{f}(\bar{\mathbf{x}}_k + \delta\mathbf{x}_k, \boldsymbol{\alpha}_k^* + \delta\boldsymbol{\alpha}_k; t_k) - \mathbf{f}(\bar{\mathbf{x}}_k, \bar{\boldsymbol{\alpha}}_k; t_k)$$

$$= \delta\mathbf{f}_k + \Delta\mathbf{f}_k \tag{57}$$

where

$$\delta\mathbf{f}_k = \mathbf{f}(\bar{\mathbf{x}}_k + \delta\mathbf{x}_k, \boldsymbol{\alpha}_k^* + \delta\boldsymbol{\alpha}_k; t_k) - \mathbf{f}(\bar{\mathbf{x}}_k, \boldsymbol{\alpha}_k^*; t_k)$$

$$\approx \frac{1}{2}\delta\mathbf{x}_k^T \mathbf{f}_{xx}^k \delta\mathbf{x}_k + \frac{1}{2}\delta\boldsymbol{\alpha}_k^T \mathbf{f}_{\alpha\alpha}^k \delta\boldsymbol{\alpha}_k + \delta\boldsymbol{\alpha}_k^T \mathbf{f}_{\alpha x}^k \delta\mathbf{x}_k + \mathbf{f}_x^k \delta\mathbf{x}_k + \mathbf{f}_\alpha^k \delta\boldsymbol{\alpha}_k, \tag{58}$$

$$\Delta\mathbf{f}_k = \mathbf{f}(\bar{\mathbf{x}}_k, \boldsymbol{\alpha}_k^*; t_k) - \mathbf{f}(\bar{\mathbf{x}}_k, \bar{\boldsymbol{\alpha}}_k; t_k) = \mathbf{f}(\bar{\mathbf{x}}_k, \boldsymbol{\alpha}_k^*; t_k) - \bar{\mathbf{x}}_{k+1}. \tag{59}$$

Substitute $\delta\mathbf{x}_{k+1}$ with equation (57), we have

$$\frac{1}{2}\delta\mathbf{x}_k^T \mathbf{V}_{xx}^k \delta\mathbf{x}_k + \mathbf{V}_x^{kT}\delta\mathbf{x}_k + \Theta_k + \bar{V}_k(\bar{\mathbf{x}}_k)$$

$$\approx \min_{\delta\boldsymbol{\alpha}_k}\left\{\frac{1}{2}\delta\mathbf{x}_k^T \mathbf{G}_{xx}^k \delta\mathbf{x}_k + \frac{1}{2}\delta\boldsymbol{\alpha}_k^T \mathbf{G}_{\alpha\alpha}^k \delta\boldsymbol{\alpha}_k \right.$$

$$+ \delta\boldsymbol{\alpha}_k^T \mathbf{G}_{\alpha x}^k \delta\mathbf{x}_k + \mathbf{G}_x^{kT}\delta\mathbf{x}_k + \mathbf{G}_\alpha^{kT}\delta\boldsymbol{\alpha}_k + \Delta G_k + G(\bar{\mathbf{x}}_k, \bar{\boldsymbol{\alpha}}_k; t_k)$$

$$+ \frac{1}{2}\delta\mathbf{f}_k^T \mathbf{V}_{xx}^{k+1}\delta\mathbf{f}_k + \Delta\mathbf{f}_k^T \mathbf{V}_{xx}^{k+1}\delta\mathbf{f}_k + \mathbf{V}_x^{k+1T}\delta\mathbf{f}_k$$

$$\left. + \frac{1}{2}\Delta\mathbf{f}_k^T \mathbf{V}_{xx}^{k+1}\Delta\mathbf{f}_k + \mathbf{V}_x^{k+1T}\Delta\mathbf{f}_k + \Theta_{k+1} + \bar{V}_{k+1}(\bar{\mathbf{x}}_{k+1})\right\}. \tag{60}$$

To get the complete form of quadratic expansion, we still need to take care of terms involving $\delta\mathbf{f}_k$, i.e., expand them with respect to $\delta\boldsymbol{\alpha}_k$ and $\delta\mathbf{x}_k$. For the sake of simplicity, introduce

$$\mathbf{P}_k = \mathbf{V}_{xx}^k, \quad \mathbf{q}_k = \mathbf{V}_x^k,$$

$$\mathbf{w}_k = \mathbf{P}_{k+1}\Delta\mathbf{f}_k + \mathbf{q}_{k+1},$$

$$h(\mathbf{x}_k, \boldsymbol{\alpha}_k; t_k) = \mathbf{w}_k^T \mathbf{f}(\mathbf{x}_k, \boldsymbol{\alpha}_k; t_k),$$

and thus

$$\delta h^k = h(\bar{\mathbf{x}}_k + \delta\mathbf{x}_k, \boldsymbol{\alpha}_k^* + \delta\boldsymbol{\alpha}_k; t_k) - h(\bar{\mathbf{x}}_k, \boldsymbol{\alpha}_k^*; t_k) = \mathbf{w}_k^T \delta\mathbf{f}_k.$$

Since we are only deriving quadratic approximation, to expand the term on the right hand side of (60), $\frac{1}{2}\delta\mathbf{f}_k^T \mathbf{V}_{xx}^{k+1} \delta\mathbf{f}_k$, we only need use partial terms as in (58), namely $\left(\mathbf{f}_x^k \delta\mathbf{x}_k + \mathbf{f}_\alpha^k \delta\boldsymbol{\alpha}_k\right)$. Thus, with new notations we have

$$
\begin{aligned}
\frac{1}{2}\delta\mathbf{f}_k^T \mathbf{V}_{xx}^{k+1} \delta\mathbf{f}_k &= \frac{1}{2}\delta\mathbf{f}_k^T \mathbf{P}_{k+1} \delta\mathbf{f}_k \\
&\approx \frac{1}{2}\left(\mathbf{f}_x^k \delta\mathbf{x}_k + \mathbf{f}_\alpha^k \delta\boldsymbol{\alpha}_k\right)^T \mathbf{P}_{k+1}\left(\mathbf{f}_x^k \delta\mathbf{x}_k + \mathbf{f}_\alpha^k \delta\boldsymbol{\alpha}_k\right) \\
&= \frac{1}{2}\delta\mathbf{x}_k^T \mathbf{f}_x^{kT} \mathbf{P}_{k+1} \mathbf{f}_x^k \delta\mathbf{x}_k + \frac{1}{2}\delta\boldsymbol{\alpha}_k^T \mathbf{f}_\alpha^{kT} \mathbf{P}_{k+1} \mathbf{f}_\alpha^k \delta\boldsymbol{\alpha}_k + \delta\boldsymbol{\alpha}_k^T \mathbf{f}_\alpha^{kT} \mathbf{P}_{k+1} \mathbf{f}_x^k \delta\mathbf{x}_k.
\end{aligned}
$$

Furthermore,

$$
\begin{aligned}
\Delta\mathbf{f}_k^T \mathbf{V}_{xx}^{k+1} \delta\mathbf{f}_k + \mathbf{V}_x^{k+1T} \delta\mathbf{f}_k &= \Delta\mathbf{f}_k^T \mathbf{P}_{k+1} \delta\mathbf{f}_k + \mathbf{q}_{k+1}^T \delta\mathbf{f}_k = \mathbf{w}_k^T \delta\mathbf{f}_k \\
&= \delta h^k \approx \frac{1}{2}\delta\mathbf{x}_k^T \mathbf{h}_{xx}^k \delta\mathbf{x}_k + \frac{1}{2}\delta\boldsymbol{\alpha}_k^T \mathbf{h}_{\alpha\alpha}^k \delta\boldsymbol{\alpha}_k + \delta\boldsymbol{\alpha}_k^T \mathbf{h}_{\alpha x}^k \delta\mathbf{x}_k + \mathbf{h}_x^{kT} \delta\mathbf{x}_k + \mathbf{h}_\alpha^{kT} \delta\boldsymbol{\alpha}_k.
\end{aligned}
$$

Finally, we rewrite equation (60) as

$$
\begin{aligned}
\frac{1}{2}\delta\mathbf{x}_k^T \mathbf{P}_k \delta\mathbf{x}_k + \mathbf{q}_k^T \delta\mathbf{x}_k + \Theta_k \approx \min_{\delta\boldsymbol{\alpha}_k} \Bigg\{ &\frac{1}{2}\delta\mathbf{x}_k^T \mathbf{A}_k \delta\mathbf{x}_k + \frac{1}{2}\delta\boldsymbol{\alpha}_k^T \mathbf{C}_k \delta\boldsymbol{\alpha}_k + \delta\boldsymbol{\alpha}_k^T \mathbf{B}_k \delta\mathbf{x}_k \\
&+ \mathbf{e}_k^T \delta\mathbf{x}_k + \mathbf{d}_k^T \delta\boldsymbol{\alpha}_k + \Theta_{k+1} + \Delta g_k + \mathbf{q}_{k+1}^T \Delta\mathbf{f}_k + \frac{1}{2}\Delta\mathbf{f}_k^T \mathbf{P}_{k+1} \Delta\mathbf{f}_k \Bigg\},
\end{aligned}
\tag{61}
$$

where we define

$$\mathbf{A}_k = \mathbf{G}_{xx}^k + \mathbf{h}_{xx}^k + \mathbf{f}_x^{kT} \mathbf{P}_{k+1} \mathbf{f}_x^k, \tag{62}$$

$$\mathbf{C}_k = \mathbf{G}_{\alpha\alpha}^k + \mathbf{h}_{\alpha\alpha}^k + \mathbf{f}_\alpha^{kT} \mathbf{P}_{k+1} \mathbf{f}_\alpha^k, \tag{63}$$

$$\mathbf{B}_k = \mathbf{G}_{\alpha x}^k + \mathbf{h}_{\alpha x}^k + \mathbf{f}_\alpha^{kT} \mathbf{P}_{k+1} \mathbf{f}_x^k, \tag{64}$$

$$\mathbf{e}_k = \mathbf{G}_x^k + \mathbf{h}_x^k, \tag{65}$$

$$\mathbf{d}_k = \mathbf{G}_\alpha^k + \mathbf{h}_\alpha^k. \tag{66}$$

5.2.1   Feedback Control Law and Algorithm

In the preceding sections, we have the expansion work done and now we may develop the feedback control law. By inspecting the right-hand side of equation (61), we discover that only three terms depend on $\delta\boldsymbol{\alpha}_k$ and $\delta\boldsymbol{\alpha}_k$ is in fact the minimizer of

$$\min_{\delta\boldsymbol{\alpha}_k}\{\frac{1}{2}\delta\boldsymbol{\alpha}_k^T \mathbf{C}_k \delta\boldsymbol{\alpha}_k + \delta\boldsymbol{\alpha}_k^T \mathbf{B}_k \delta\mathbf{x}_k + \mathbf{d}_k^T \delta\boldsymbol{\alpha}_k\}.$$

Letting the gradient of the minimizing function above equal 0, we have $\delta\boldsymbol{\alpha}_k$ satisfying

$$\mathbf{C}_k\delta\boldsymbol{\alpha}_k + \mathbf{B}_k\delta\mathbf{x}_k + \mathbf{d}_k = 0.$$

Therefore the feedback control law is

$$\delta\boldsymbol{\alpha}_k = \mathbf{z}_k - \mathbf{K}_k\delta\mathbf{x}_k, \tag{67}$$

where

$$\mathbf{K}_k = \mathbf{C}_k^{-1}\mathbf{B}_k, \tag{68}$$

$$\mathbf{z}_k = -\mathbf{C}_k^{-1}\mathbf{d}_k. \tag{69}$$

Plugging equation (67) into equation (61) and grouping the terms of the same order, we get all the information at time k which are

$$\mathbf{P}_k = \mathbf{A}_k - \mathbf{B}_k^T\mathbf{C}_k^{-1}\mathbf{B}_k, \tag{70}$$

$$\mathbf{q}_k = \mathbf{e}_k + \mathbf{B}_k^T\mathbf{z}_k, \tag{71}$$

$$\Theta_k = \Theta_{k+1} + \frac{1}{2}\mathbf{d}_k^T\mathbf{z}_k. \tag{72}$$

Also we have the final conditions

$$\mathbf{P}_L = \nabla_{xx}\big[\phi(\bar{\mathbf{x}}_L)\big], \tag{73}$$

$$\mathbf{q}_L = \nabla_x\big[\phi(\bar{\mathbf{x}}_L)\big], \tag{74}$$

$$\Theta_L = 0, \tag{75}$$

$$V_L(\bar{\mathbf{x}}_L) = \phi(\bar{\mathbf{x}}_L). \tag{76}$$

Differential Dynamic Programming (DDP) ensures an improvement at each iteration under the condition that the Hessian matrix of the cost function, i.e., the matrix $\mathbf{C}_k$ is positive definite. The procedure based on *Pareto-curve continuation* can be also implemented to enforce the convexity.

Now we outline the algorithm for the discrete nonlinear optimal control problem.

Given a tolerance $\epsilon > 0$.

I. Initialization: For k from 0 to $L-1$, set $\bar{\boldsymbol{\alpha}}_k = 0$ and use (48) to compute $\bar{\mathbf{x}}_k$.

Repeat

II. Backward recursion:

1. Use (92)-(75) to compute final conditions $\mathbf{P}_L, \mathbf{q}_L, \Theta_L$.

2. For k from $L-1$ to 0, compute $\mathbf{A}_k, \mathbf{B}_k, \mathbf{C}_k, \mathbf{e}_k$ using (62)-(65). Then compute the feedback information $\mathbf{K}_k, \mathbf{z}_k$ using (68) and (69) and update $\mathbf{P}_k, \mathbf{q}_k, \Theta_k$ using (70)-(72).

3. Stopping criterion: Quit if $\sqrt{-\Theta_0} < \epsilon$.

III. Forward recursion: For k from 0 to $L-1$,

$$\mathbf{x}_0 = \bar{\mathbf{x}}_0,$$

$$\boldsymbol{\alpha}_k = \bar{\boldsymbol{\alpha}} + \mathbf{z}_k - \mathbf{K}_k(\mathbf{x}_k - \bar{\mathbf{x}}_k),$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \tau_k \mathbf{K}(\mathbf{x}_k)\boldsymbol{\alpha}_k.$$

Consider the term $\frac{1}{2}\mathbf{d}_k^T \mathbf{z}_k$ in (72). Based on (69), we have

$$\frac{1}{2}\mathbf{d}_k^T \mathbf{z}_k = -\frac{1}{2}\mathbf{d}_k^T \mathbf{C}_k^{-1}\mathbf{d}_k.$$

Since $\mathbf{C}_k$ is symmetric positive definite, $\frac{1}{2}\mathbf{d}_k^T \mathbf{z}_k < 0$ and therefore the Newton step is a descent direction. Moreover, $\Theta_0 < 0$, so the stopping criterion is $\sqrt{-\Theta_0} < \epsilon$.

5.3 Continuous Reccati Equations

In this section, we first introduce an informal way to derive the HJB equation. We then present the continuous-time Riccati equation to linear quadratic problems taking advantage of the sufficiency of the HJB equation. Lastly, we apply the same techniques to obtain the continuous-time Riccati equations for nonlinear optimal control problems.

5.3.1 The Hamilton-Jacobi-Bellman Equation

We have proved the HJB equation in Theorem 4.2, a sufficiency theorem. Now starting with the discrete formulism, the Bellman's principle of optimality, we introduce another approach to derive the equation. We will later apply it to our continuous-time settings.
We still use the same notations for the time discretizations as in (47),

$$0 = t_0 < t_1 < t_2 < \cdots < t_{L-1} < t_L = 1,$$

with $\tau_k = t_{k+1} - t_k$, $k = 0, 1, \cdots, L-1$. Also, we have

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \tau_k \mathbf{F}(\mathbf{x}_k, \boldsymbol{\alpha}_k; t_k),$$

and

$$G(\mathbf{x}_k, \boldsymbol{\alpha}_k) = \tau_k g(\mathbf{x}_k, \boldsymbol{\alpha}_k).$$

By Bellman's principle of optimality, Theorem 5.1, we have

$$V(\mathbf{x}, t_k) = \min_{\boldsymbol{\alpha}}\big\{\tau_k g(\mathbf{x}, \boldsymbol{\alpha}) + V(\mathbf{x} + \tau_k \mathbf{F}(\mathbf{x}, \boldsymbol{\alpha}), t_k + \tau_k)\big\}, \tag{77}$$

for $k = 0, \cdots, L-1$, and the terminal condition is,

$$V(\mathbf{x}, t_L) = \phi(\mathbf{x}).$$

24

Assuming that V has the required differentiability properties, we expand it into a first order Taylor series around $(\mathbf{x}, t_k)$,

$$
\begin{aligned}
V(\mathbf{x} + \tau_k \mathbf{F}(\mathbf{x}, \boldsymbol{\alpha}), t_k + \tau_k) = {}& V(\mathbf{x}, t_k) + \nabla_x V(\mathbf{x}, t_k)' F(\mathbf{x}, \boldsymbol{\alpha}) \tau_k \\
& + \nabla_t V(\mathbf{x}, t_k) \tau_k + o(\tau_k),
\end{aligned}
\tag{78}
$$

where $o(\tau_k)$ representing second order terms, satisfies

$$
\lim_{\tau_k \to 0, L \to \infty} o(\tau_k)/\tau_k = 0.
$$

Moreover, ∇_x denotes the column vector of partial derivatives with respect to x and ∇_t denotes partial derivatives with respect to t.

Substitute the expansion (78) into (77) and we obtain

$$
\begin{aligned}
V(\mathbf{x}, t_k) = \min_{\boldsymbol{\alpha}} \big\{ {}& \tau_k g(\mathbf{x}, \boldsymbol{\alpha}) + V(\mathbf{x}, t_k) + \nabla_x V(\mathbf{x}, t_k)' F(\mathbf{x}, \boldsymbol{\alpha}) \tau_k \\
& + \nabla_t V(\mathbf{x}, t_k) \tau_k + o(\tau_k) \big\}.
\end{aligned}
$$

Cancel $V(\mathbf{x}, t_k)$ from both sides, divide by τ_k, and we have

$$
0 = \min_{\boldsymbol{\alpha}} \big\{ g(\mathbf{x}, \boldsymbol{\alpha}) + \nabla_x V(\mathbf{x}, t_k)' F(\mathbf{x}, \boldsymbol{\alpha}) + \nabla_t V(\mathbf{x}, t_k) + o(\tau_k)/\tau_k \big\}.
$$

Assume that the discrete-time value function V yields its continuous-time counterpart by taking the limit as $\tau_k \to 0$ and $L \to \infty$, that is,

$$
\lim_{L \to \infty, \tau_k \to 0, t_k = t} V(\mathbf{x}, t_k) = V(\mathbf{x}, t),
\tag{79}
$$

for all $\mathbf{x}, t$. Therefore, taking the limit, we obtain

$$
0 = \min_{\boldsymbol{\alpha}} \big\{ g(\mathbf{x}, \boldsymbol{\alpha}) + \nabla_x V(\mathbf{x}, t)' F(\mathbf{x}, \boldsymbol{\alpha}) + \nabla_t V(\mathbf{x}, t) \big\},
$$

for all $\mathbf{x}, t$, and the boundary condition is

$$
V(\mathbf{x}, 1) = \phi(\mathbf{x}).
$$

This is the Hamilton-Jacobi-Bellman equation as we discussed in theorem 4.2. In the preceding derivation, the differentiability of $V(\mathbf{x}, t)$ is assumed among other things. Based on theorem 4.2, if we can solve the HJB equation analytically or computationally, then we can obtain an optimal control policy by minimizing its right-hand side. The statement is analogous to a corresponding statement for discrete-time dynamic programming: if we can execute the DP algorithm, we can find an optimal policy through minimization based on Bellman's principle 5.1.

5.3.2 Linear Quadratic Problems

We derive continuous-time Riccati equation for linear dynamic system with quadratic cost functional using the HJB equation.

Consider the n-dimensional linear system

$$\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + B\boldsymbol{\alpha}(t),$$

where A and B are given matrices, and the quadratic cost is

$$J(\boldsymbol{\alpha}) = \Delta_0^1 (\mathbf{x}(t)'Q\mathbf{x}(t) + \boldsymbol{\alpha}(t)R\boldsymbol{\alpha}(t))dt + \mathbf{x}(1)'Q_T\mathbf{x}(1),$$

where the matrices Q_T and Q are symmetric positive semidefinite, and the matrix R is symmetric positive definite.

Following the HJB equation (35), we have

$$\min_{\boldsymbol{\alpha}}\{V_t(\mathbf{x},t) + \nabla_x V(\mathbf{x},t)'(A\mathbf{x} + B\boldsymbol{\alpha}) + \mathbf{x}'Q\mathbf{x} + \boldsymbol{\alpha}'R\boldsymbol{\alpha}\} = 0, \tag{80}$$

with final state condition,

$$V(\mathbf{x},1) = \mathbf{x}(1)'Q_T\mathbf{x}(1).$$

Since HJB equation is sufficient, let us try a solution of the form, separation of variables,

$$V(\mathbf{x},t) = \mathbf{x}'K(t)\mathbf{x},$$

where $K(t)$ is symmetric. Thus, we have

$$V_t(\mathbf{x},t) = \mathbf{x}'\dot{K}(t)\mathbf{x}, \tag{81}$$

$$\nabla_x V(\mathbf{x},t) = 2K(t)\mathbf{x} \tag{82}$$

where elements of $\dot{K}(t)$ are the first order derivatives of the elements of $K(t)$ with respect to time. Substitute these expressions (81) and (82) in equation (80) and we obtain

$$\min_{\boldsymbol{\alpha}}\{\mathbf{x}'\dot{K}(t)\mathbf{x} + 2\mathbf{x}'K(t)(A\mathbf{x} + B\boldsymbol{\alpha}) + \mathbf{x}'Q\mathbf{x} + \boldsymbol{\alpha}'R\boldsymbol{\alpha}\} = 0. \tag{83}$$

The minimum is acquired when the gradient with respect to $\boldsymbol{\alpha}$ is zero, that is,

$$2B'K(t)\mathbf{x} + 2R\boldsymbol{\alpha} = 0.$$

Thus,

$$\boldsymbol{\alpha}(t) = -R^{-1}B'K(t)\mathbf{x}. \tag{84}$$

Substitute (84) back to equation (83), we have

$$\min_{\boldsymbol{\alpha}}\{\mathbf{x}'\dot{K}(t)\mathbf{x} + 2\mathbf{x}'K(t)(A\mathbf{x} - BR^{-1}B'K(t)\mathbf{x}) + \mathbf{x}'Q\mathbf{x} + \mathbf{x}'K(t)BR^{-1}B'K(t)\mathbf{x}\} = 0,$$

and
$$\min_{\boldsymbol{\alpha}}\{\mathbf{x}'(\dot{K}(t) + 2K(t)A - 2K(t)BR^{-1}B'K(t) + Q + K(t)BR^{-1}B'K(t))\mathbf{x}\} = 0.$$

Hence, $V(\mathbf{x}, t)$ must satisfy the following matrix differential equation,

$$\dot{K}(t) + 2K(t)A + Q - K(t)BR^{-1}B'K(t) = 0, \tag{85}$$

with the terminal condition,

$$K(1) = Q_T. \tag{86}$$

The differential equation (85) is known as continuous-time Riccati equation.
Therefore, we conclude the optimal value function is

$$V(\mathbf{x}, t) = \mathbf{x}'K(t)\mathbf{x},$$

and $K(t)$ is a solution of the Riccati equation (85) with the boundary condition (86). Furthermore, an optimal policy according to (84) is
$$\boldsymbol{\alpha}^*(t) = -R^{-1}B'K(t)\mathbf{x}.$$

5.3.3 Continuous-time Optimal Control

We follow the same notations as in section 5.1 and 5.3.1. The continuous-time Riccati equations are analogous to the dynamic programming algorithm. To derive them, we now apply DP to the discrete-time approximation. To begin with, we have

$$\mathbf{f}_x(\mathbf{x}_k, \boldsymbol{\alpha}_k) = I + \tau_k \mathbf{F}_x(\mathbf{x}_k, \boldsymbol{\alpha}_k),$$

$$\mathbf{f}_\alpha(\mathbf{x}_k, \boldsymbol{\alpha}_k) = \tau_k \mathbf{F}_\alpha(\mathbf{x}_k, \boldsymbol{\alpha}_k),$$

$$\mathbf{G}_x(\mathbf{x}_k, \boldsymbol{\alpha}_k) = \tau_k \mathbf{g}_x(\mathbf{x}_k, \boldsymbol{\alpha}_k),$$

$$\mathbf{G}_\alpha(\mathbf{x}_k, \boldsymbol{\alpha}_k) = \tau_k \mathbf{g}_\alpha(\mathbf{x}_k, \boldsymbol{\alpha}_k).$$

With all the gradients substituted in, the Hessian matrices and linear terms in (62)-(66) are

$$\mathbf{A}_k = \mathbf{P}_{k+1} + \tau_k(\mathbf{g}_{xx}^k + \mathbf{F}_{xx}^k \mathbf{q}_{k+1} + 2\mathbf{F}_x^{kT}\mathbf{P}_{k+1}) + \tau_k^2(\mathbf{P}_{k+1}\mathbf{F}_{xx}^k \Delta \mathbf{F}^k + \mathbf{F}_x^{kT}\mathbf{P}_{k+1}\mathbf{F}_x^k),$$

$$\mathbf{C}_k = \tau_k(\mathbf{g}_{\alpha\alpha}^k + \mathbf{F}_{\alpha\alpha}^k \mathbf{q}_{k+1}) + \tau_k^2(\mathbf{P}_{k+1}\mathbf{F}_{\alpha\alpha}^k \Delta \mathbf{F}^k + \mathbf{F}_\alpha^{kT}\mathbf{P}_{k+1}\mathbf{F}_\alpha^k),$$

$$\mathbf{B}_k = \tau_k(\mathbf{g}_{\alpha x}^k + \mathbf{F}_{\alpha x}^k \mathbf{q}_{k+1} + \mathbf{F}_\alpha^{kT}\mathbf{P}_{k+1}) + \tau_k^2(\mathbf{P}_{k+1}\mathbf{F}_{\alpha x}^k \Delta \mathbf{F}^k + \mathbf{F}_\alpha^{kT}\mathbf{P}_{k+1}\mathbf{F}_x^k),$$

$$\mathbf{e}_k = \mathbf{q}_{k+1} + \tau_k(\mathbf{g}_x + \mathbf{P}_{k+1}\Delta \mathbf{F}^k + \mathbf{F}_x^{kT}\mathbf{q}_{k+1}) + \tau_k^2 \mathbf{P}_{k+1}\mathbf{F}_x^k \Delta \mathbf{F}^k,$$

$$\mathbf{d}_k = \tau_k(\mathbf{g}_\alpha + \mathbf{F}_\alpha^T \mathbf{q}_{k+1}) + \tau_k^2 \mathbf{P}_{k+1}\mathbf{F}_\alpha^k \Delta \mathbf{F}^k.$$

As to backward recursions, we first examine (70) and have

$$\mathbf{P}_k = \mathbf{P}_{k+1} + \tau_k(\mathbf{g}_{xx}^k + \mathbf{F}_{xx}^k\mathbf{q}_{k+1} + 2\mathbf{F}_x^{kT}\mathbf{P}_{k+1}) + \tau_k^2(\mathbf{P}_{k+1}\mathbf{F}_{xx}^k\Delta\mathbf{F}^k + \mathbf{F}_x^{kT}\mathbf{P}_{k+1}\mathbf{F}_x^k)$$

$$- (\tau_k(\mathbf{g}_{\alpha x}^k + \mathbf{F}_{\alpha x}^k\mathbf{q}_{k+1} + \mathbf{F}_\alpha^{kT}\mathbf{P}_{k+1}) + \tau_k^2(\mathbf{P}_{k+1}\mathbf{F}_{\alpha x}^k\Delta\mathbf{F}^k + \mathbf{F}_\alpha^{kT}\mathbf{P}_{k+1}\mathbf{F}_x^k))^T$$

$$(\tau_k(\mathbf{g}_{\alpha\alpha}^k + \mathbf{F}_{\alpha\alpha}^k\mathbf{q}_{k+1}) + \tau_k^2(\mathbf{P}_{k+1}\mathbf{F}_{\alpha\alpha}^k\Delta\mathbf{F}^k + \mathbf{F}_\alpha^{kT}\mathbf{P}_{k+1}\mathbf{F}_\alpha^k))^{-1}$$

$$(\tau_k(\mathbf{g}_{\alpha x}^k + \mathbf{F}_{\alpha x}^k\mathbf{q}_{k+1} + \mathbf{F}_\alpha^{kT}\mathbf{P}_{k+1}) + \tau_k^2(\mathbf{P}_{k+1}\mathbf{F}_{\alpha x}^k\Delta\mathbf{F}^k + \mathbf{F}_\alpha^{kT}\mathbf{P}_{k+1}\mathbf{F}_x^k)).$$

Comparing the discrete update law for the Hessian to (46) which we obtain through the continuous-time Riccati equation in section 4.3, we notice extra terms involving τ_k^2. For the application to diffeomorphic matching, τ_k is usually not small, so it seems important to track and compute the additional terms of τ_k^2.

The formalism of the discrete problems can be linked to that of the continuous-time models by assumption similar to (79). Take limits as $\tau_k \to 0$ and $L \to \infty$, we assume,

$$\lim_{L\to\infty,\tau_k\to 0,t_k=t} \mathbf{P}_k = \lim_{L\to\infty,\tau_k\to 0,t_k=t} \mathbf{P}(t_k) = \mathbf{P}(t).$$

Thus, move $\mathbf{P}_{k+1}$ to the left hand side and divide the equation by τ_k. Letting $\tau_k \to 0$, we have,

$$-\dot{\mathbf{P}} = \mathbf{g}_{xx} + \mathbf{F}_{xx}\mathbf{q} + 2\mathbf{F}_x^T\mathbf{P}$$

$$- (\mathbf{g}_{\alpha x} + \mathbf{F}_{\alpha x}\mathbf{q} + \mathbf{F}_\alpha^T\mathbf{P})^T(\mathbf{g}_{\alpha\alpha} + \mathbf{F}_{\alpha\alpha}\mathbf{q})^{-1}(\mathbf{g}_{\alpha x} + \mathbf{F}_{\alpha x}\mathbf{q} + \mathbf{F}_\alpha^T\mathbf{P}). \tag{87}$$

For linear quadratic problems, $\mathbf{q}, \mathbf{F}_{xx}$, and $\mathbf{F}_{\alpha\alpha}$ vanish, and this equation is exactly the continuous-time Riccati equation (85). We perform similar process to linear terms $\mathbf{q}$, and Θ, and we have,

$$\mathbf{q}_k = \mathbf{q}_{k+1} + \tau_k(\mathbf{g}_x + \mathbf{P}_{k+1}\Delta\mathbf{F}^k + \mathbf{F}_x^{kT}\mathbf{q}_{k+1}) + \tau_k^2\mathbf{P}_{k+1}\mathbf{F}_x^k\Delta\mathbf{F}^k$$

$$- (\tau_k(\mathbf{g}_{\alpha x}^k + \mathbf{F}_{\alpha x}^k\mathbf{q}_{k+1} + \mathbf{F}_\alpha^{kT}\mathbf{P}_{k+1}) + \tau_k^2(\mathbf{P}_{k+1}\mathbf{F}_{\alpha x}^k\Delta\mathbf{F}^k + \mathbf{F}_\alpha^{kT}\mathbf{P}_{k+1}\mathbf{F}_x^k))^T$$

$$(\tau_k(\mathbf{g}_{\alpha\alpha}^k + \mathbf{F}_{\alpha\alpha}^k\mathbf{q}_{k+1}) + \tau_k^2(\mathbf{P}_{k+1}\mathbf{F}_{\alpha\alpha}^k\Delta\mathbf{F}^k + \mathbf{F}_\alpha^{kT}\mathbf{P}_{k+1}\mathbf{F}_\alpha^k))^{-1}$$

$$(\tau_k(\mathbf{g}_\alpha + \mathbf{F}_\alpha^T\mathbf{q}_{k+1}) + \tau_k^2\mathbf{P}_{k+1}\mathbf{F}_\alpha^k\Delta\mathbf{F}^k),$$

$$\Theta_k = \Theta_{k+1} - \frac{1}{2}(\tau_k(\mathbf{g}_\alpha + \mathbf{F}_\alpha^T\mathbf{q}_{k+1}) + \tau_k^2\mathbf{P}_{k+1}\mathbf{F}_\alpha^k\Delta\mathbf{F}^k)^T$$

$$(\tau_k(\mathbf{g}_{\alpha\alpha}^k + \mathbf{F}_{\alpha\alpha}^k\mathbf{q}_{k+1}) + \tau_k^2(\mathbf{P}_{k+1}\mathbf{F}_{\alpha\alpha}^k\Delta\mathbf{F}^k + \mathbf{F}_\alpha^{kT}\mathbf{P}_{k+1}\mathbf{F}_\alpha^k))^{-1}$$

$$(\tau_k(\mathbf{g}_\alpha + \mathbf{F}_\alpha^T\mathbf{q}_{k+1}) + \tau_k^2\mathbf{P}_{k+1}\mathbf{F}_\alpha^k\Delta\mathbf{F}^k).$$

Letting $\tau_k \to 0$ and $L \to \infty$,

$$\lim_{L\to\infty,\tau_k\to 0,t_k=t} \mathbf{q}_k = \mathbf{q}(t), \qquad \lim_{L\to\infty,\tau_k\to 0,t_k=t} \Theta_k = \Theta(t).$$

Therefore, we obtain the Riccati equations for the linear terms

$$-\dot{\mathbf{q}} = \mathbf{g}_x + \mathbf{F}_x^T\mathbf{q} - (\mathbf{g}_{\alpha x} + \mathbf{F}_{\alpha x}\mathbf{q} + \mathbf{F}_\alpha^T\mathbf{P})^T(\mathbf{g}_{\alpha\alpha} + \mathbf{F}_{\alpha\alpha}\mathbf{q})^{-1}(\mathbf{g}_\alpha + \mathbf{F}_\alpha^T\mathbf{q}), \tag{88}$$

$$-\dot{\Theta} = -\frac{1}{2}(\mathbf{g}_\alpha + \mathbf{F}_\alpha^T\mathbf{q})^T(\mathbf{g}_{\alpha\alpha} + \mathbf{F}_{\alpha\alpha}\mathbf{q})^{-1}(\mathbf{g}_\alpha + \mathbf{F}_\alpha^T\mathbf{q}). \tag{89}$$

Hence, $\mathbf{P}, \mathbf{q}$, and Θ must satisfy the preceding differential equations (87)-(89) with final conditions,

$$\mathbf{P}(1) = \nabla_{xx}\big[\phi(\mathbf{x}(1))\big],$$

$$\mathbf{q}(1) = \nabla_x\big[\phi(\mathbf{x}(1))\big],$$

$$\Theta(1) = 0.$$

5.4 Applications to Diffeomorphic Matching Problem

In this section, we apply the variational approach to diffeomorphic matching problems. For many shape-matching applications, K can be the radial Gaussian kernel K_σ with a suitable scale parameter $\sigma > 0$ where the smooth, symmetric bounded, positive definite kernel K_σ is defined as

$$K_\sigma(x, x') = \frac{1}{(2\pi)^{3/2}\sigma^3}\exp\big(-\frac{\|x - x'\|^2}{\sigma^2}\big), \tag{90}$$

for $x, x' \in \mathbb{R}^d$. The choice σ heavily depends on local mesh size of the reference data set and it is generally kept fixed during the whole computing procedure.

We have introduced the time discretization in (47) as

$$0 = t_0 < t_1 < t_2 < \cdots < t_{L-1} < t_L = 1,$$

and defined the step sizes as $\tau_k = t_{k+1} - t_k$, $k = 0, 1, \cdots, L - 1$.

Following the notation in (19)-(21), we further discretize the problem in time and it is organized as follows.

$$\mathbf{x}_k = x(t_k) = (x_1(t_k), \cdots, x_N(t_k)) \in \mathbb{R}^{Nd}, k = 0, 1, \cdots, L,$$

$$\boldsymbol{\alpha}_k = \alpha(t_k) \in \mathbb{R}^{Nd}, k = 0, 1, \cdots, L - 1.$$

$$\mathbf{K}(\mathbf{x}_k) \in \mathbb{R}^{Nd\times Nd}, \quad (\mathbf{K}(\mathbf{x}_k))_{ij} = K_\sigma(x_i(t_k), x_j(t_k))I_d \in \mathbb{R}^{d\times d}.$$

Moreover, to keep consistent with notations, we still denote the target set as $\{y^j\}_{j=1}^M$ and the reference as $\bar{x}_0 = \{x_i\}_{i=1}^N$, N and M being the cardinal number of the reference and target set. Thus, the optimality problem (24) is

$$\begin{aligned} \min \quad & J(\{\boldsymbol{\alpha}_k\}_{k=0}^{L-1}; x^{(0)}) = \sum_{k=0}^{L-1} g(\mathbf{x}_k, \boldsymbol{\alpha}_k; t_k) + \lambda\phi(\mathbf{x}_L), \\ \text{s.t.} \quad & \mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \boldsymbol{\alpha}_k; t_k), \quad k = 0, \cdots, L - 1, \\ & \mathbf{x}_0 = x^{(0)}. \end{aligned} \tag{91}$$

With the RKHS, the placement function is

$$\mathbf{f}(\mathbf{x}_k, \boldsymbol{\alpha}_k; t_k) = \mathbf{x}_k + \tau_k \mathbf{K}(\mathbf{x}_k)\boldsymbol{\alpha}_k, \tag{92}$$

and the kinetic energy function is

$$g(\mathbf{x}_k, \boldsymbol{\alpha}_k; t_k) = \frac{\tau_k}{2} \boldsymbol{\alpha}_k^T \mathbf{K}(\mathbf{x}_k)\boldsymbol{\alpha}_k. \tag{93}$$

Furthermore, we set $a_i = \frac{1}{N}$ and $b_j = \frac{1}{M}$ in (18) and the measure matching reads

$$\phi(\mathbf{x}_L) = \frac{1}{N^2} \sum_{i,j=1}^{N} K_{\sigma'}(x_i(t_L), x_j(t_L)) - \frac{2}{MN} \sum_{i=1}^{N} \sum_{j=1}^{M} K_{\sigma'}(x_i(t_L), y_j) + \frac{1}{M^2} \sum_{i,j=1}^{M} K_{\sigma'}(y_i, y_j). \tag{94}$$

The scale parameter here in the matching functional, σ', is different from σ we choose in (90). σ' is usually updated dynamically after each continuation iteration which is necessary because the matching functional indicates the distance between the target and the numerical solution and updating σ' gives a more precise evaluation of the matching quality.

6 Application

In this section, we first explain the role that the continuation method plays in the algorithm, as well as initialization of parameters. Then numerical results are presented and discussed.

6.1 The Algorithm with Continuation Method

Consider the cost function in (91), the form of regularization is essentially a trade-off between the kinetic energy (93) and the measure matching functional (94). For λ small, the regularizing effect of the kinetic energy dominates. To obtain a good matching quality, one needs to solve the optimal control problem for fixed and sufficiently large regularization parameter λ. However, the larger λ, the more ill-conditioned the problem is. Moreover, the convergence region of the Newton descent method gets smaller as λ increases and it would not converge if the initializer of control $\boldsymbol{\alpha}$ is not close to a local minimum.

To overcome the obstacle, we introduce an approximate continuation in λ. Initialize λ with a small value, and solve the optimal control problem for fixed λ using the Newton descent. Then at the end of each Newton step, increase λ by some multiplicative factor and use the computed optimal control to initialize the control for new updated regularization parameter λ.

Here we initialize $\boldsymbol{\alpha} = \mathbf{0}$. If more concrete information known at the initial state, we may definitely use other existing initialization. If not, to avoid the initial inversion of excessive computation we can crudely let $\boldsymbol{\alpha} = \mathbf{0}$.

The continuation method reads as follows:

- **Step 1 (Initialization of continuation)**
  Specify a small initial value $\lambda_0 > 0$, set $\nu = 0$.

- **Step 2. (Initialization of the inner iteration)**
  Initialize $\boldsymbol{\alpha}_\nu^{(0)} = \mathbf{0}$ and set $\mu = 0$. compute $\mathbf{x}^{(\nu,0)}$ with $\boldsymbol{\alpha}^{(\nu,0)}$ using (92).

- **Step 3. (Dynamic programming using the second order information)**

  1. Set $\mu = \mu + 1$, and compute α_ν^μ by the second order method.

  2. If the stopping criterion,
  $$\sqrt{-\Theta_0} < \epsilon,$$
  is satisfied, go to step 4. Otherwise go the step 3.

Step 4 (Termination of continuation)
If the matching functional satisfies,
$$\phi((\mathbf{x}_\nu^\mu)_{Nt}) < \text{Err},$$

stop the algorithm.
Otherwise, set $\nu = \nu + 1$, $\boldsymbol{\alpha}_\nu^{(0)} = \boldsymbol{\alpha}_{\nu-1}^\mu$, and increase the regularization parameter

$$\lambda_\nu = \gamma_\nu \lambda_{\nu-1},$$

and go to step 3.

6.2 Numerical Results

6.2.1 Mitral Valve Shape Models

The mitral valve (Figure 1) is a dual-flap valve in the heart that lies between the left atrium and the left ventricle which is typically 4-6 cm^2 in area. It has two leaflets, the anterior leaflet and the posterior leaflet, that guard the opening and the opening is surrounded by a fibrous ring known as the mitral valve annulus.

The mathematical models were generated by image analysis of live 3D-echocardiographic movies ac-
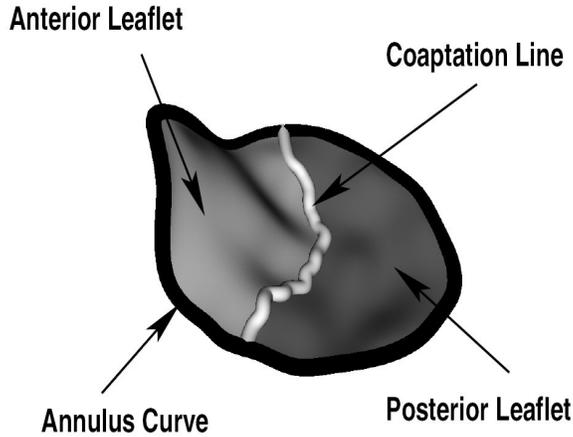
Figure 1: Mitral valve: the middle line is the coaptation line along which the surfaces of the anterior and posterior leaflets meet when the valve is closed. The closed black thick curve is the mitral annulus.

quired by ultrasound technology and each movie includes 27-30 3D frames per heartbeat cycle. The
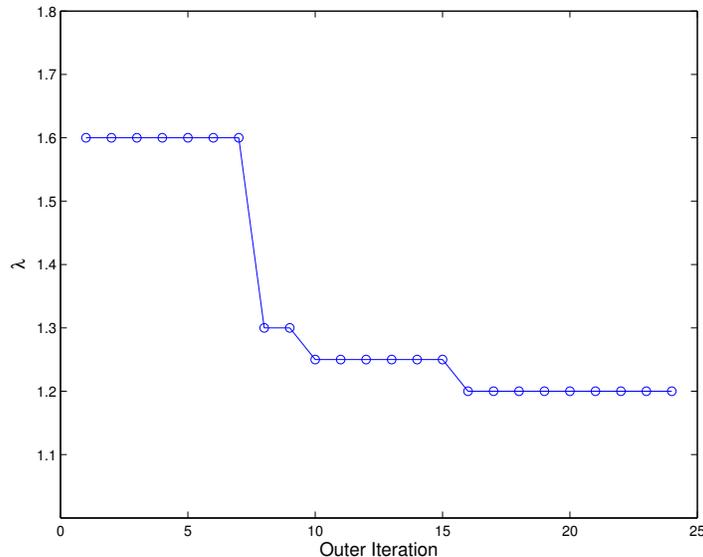
Figure 2: Multipliers of λ for matching anterior leaflets.

static snapshots were obtained by our research group on mathematical image analysis (S. Alexander, J. Freeman, A. Jajoo, S. Jain, and A. Martynenko.) in collaboration with Methodist Hospital, Houston, Texas (S. Ben Zekry, S. Little, and W.Zoghbi, MDs).

Based on NURBS (non-uniform rational B-splines), the mitral valve models are obtained by combining optical flow extraction algorithms with sparse tagging by medical experts.

The mitral valve apparatus can be viewed as a composite deformable built from several smooth deformable shapes, namely a closed curve (the mitral value annulus) and two surfaces (the anterior and posterior leaflet). Accordingly, 3 NURBS models are obtained, one for each leaflet and a third one for the mitral valve annulus.

Based on those models, [6] presented numerical results on the diffeomorphic matching for multiple snapshots, also strain was computed in [8].

6.2.2 Measure Matching for Anterior Leaflets

We present the performance of diffeomorphic matching for 2 snapshots S_0 and S_1 of the anterior leaflet. The scale and termination parameters are shown in Table 1.

The continuation algorithm starts with $\lambda = 400$ and is multiplied by γ. The values of γ are displayed in the Figure 2. Based on our numerical experiments, γ is usually chosen between (1,2) and as λ gets bigger, smaller γ is preferable. If too much increment occurs to λ, the optimal control from the former outer iteration might not be a good initializer.

Figure 3 shows the Newton decrement at each iteration. For each outer iteration, the second order method starts out at the crest value and decreases till the stopping criterion is reached. The lower figure clearly indicates one of the strongest advantages the second order method has. The Newton decrement converges very fast and needs at most six iterations to produce a solution of very high accuracy.

We use Pareto frontiers ([22]) to record and display the tradeoffs between the pair of competing criteria, measure matching and kinetic energy. For fixed regularization parameter λ, we use measure matching versus the kinetic energy to generate a point on the approximate Pareto frontier at the end of each inner iteration of the second order method. Convexity of the approximate Pareto curves is expected and observed numerically in Figure 4.
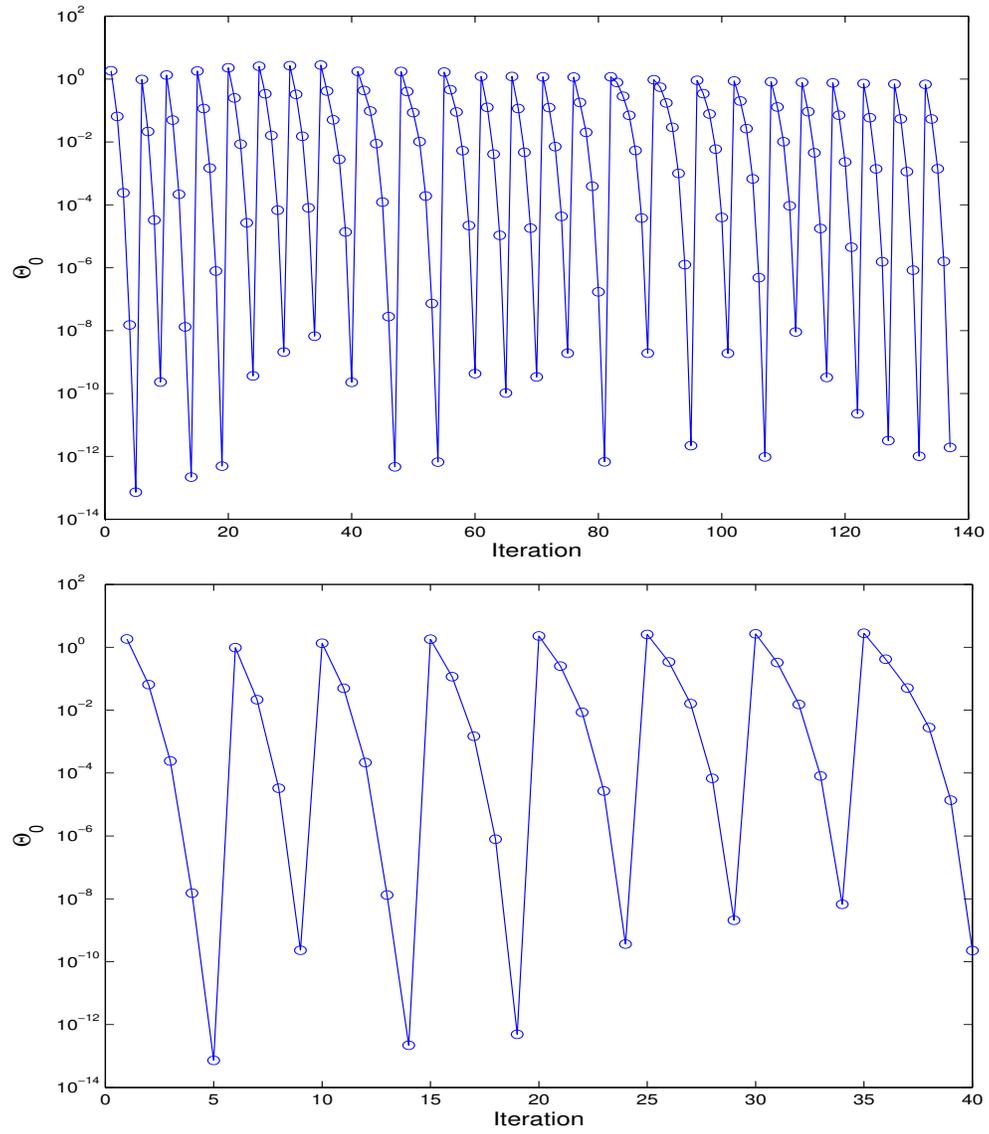
Figure 3: Newton Decrement for matching anterior leaflets: for easier visualization, the lower figure only displays Newton's decrement for iterations 1 to 40.
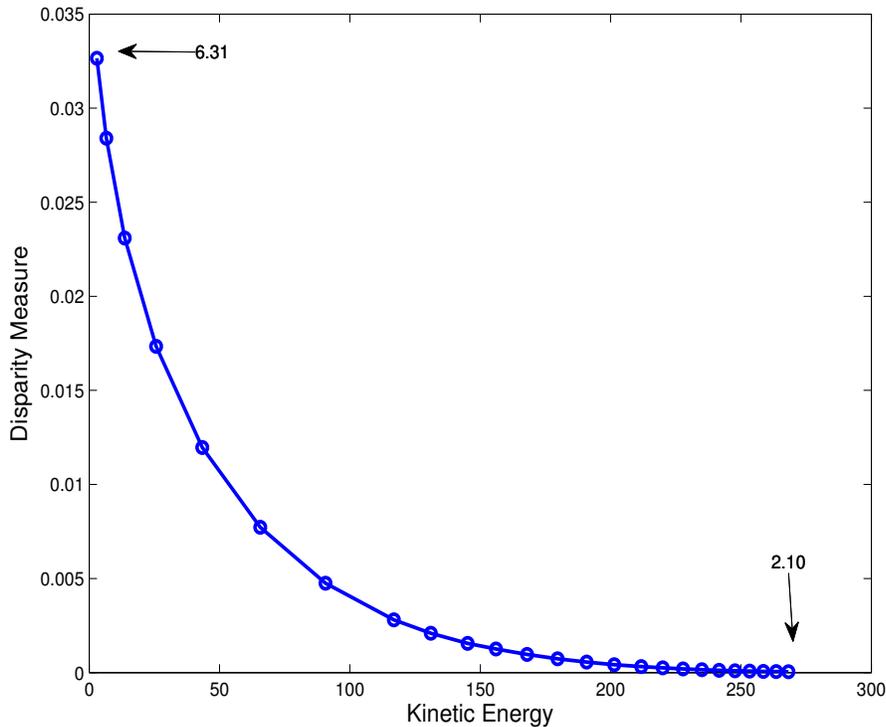
Figure 4: Pareto optimality for matching anterior leaflets.

Figure 5 shows the behaviors of kinetic energy, measure matching and Hausdorff distance in terms of λ. As we expect, kinetic energy goes up as good matching quality is obtained. Moreover, the Hausdorff distance which indicates geometric difference of two shapes also decreases. When λ is around 10^4, even though the measure is still decreasing, the Hausdorff distance stays around 2.5. It is because the resolution is already attained.

To reflect the matching quality, we present geometric distances in Figure 6. At each outer iteration with fixed λ, we compute the point distance of computed deformation $\hat{S}_1$ and the target shape S_1 and plot a box accordingly. For each box in the figure, the central mark is the median, the edges of the box are the 25th and 75th percentiles and outliers are plotted individually. It suggests that as λ increases, the median of overall distance is dramatically reduced. Furthermore, the outliers are brought down and at the end there is no outliers at all.

Parameter	Notation	Value
N	Cardinal number of reference set	83
M	Cardinal number of target set	82
σ	Scale parameter for the Gaussian kernel	4.13
L	4 Time intervals	5
ϵ	Stoping criterion of Newton decrement	1.d-8
Err	Stoping criterion of measure matching	5.d-5

Table 1: Parameters and values for matching two anterior leaflets.
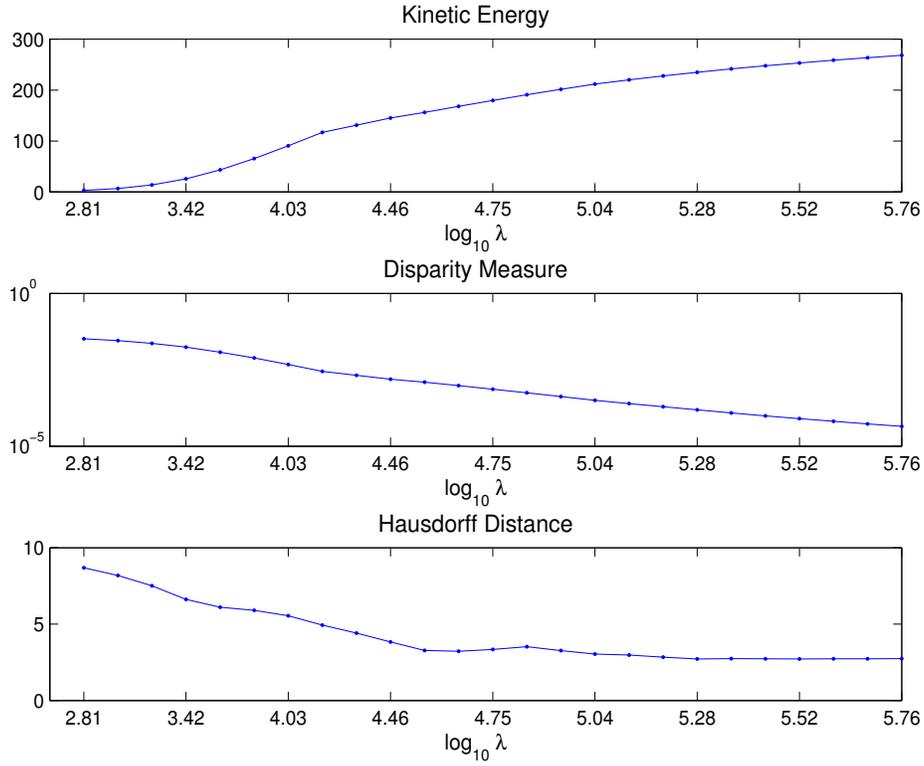
Figure 5: Behaviors of kinetic energy, measure matching, and Hausdorff distance for matching two anterior leaflets.
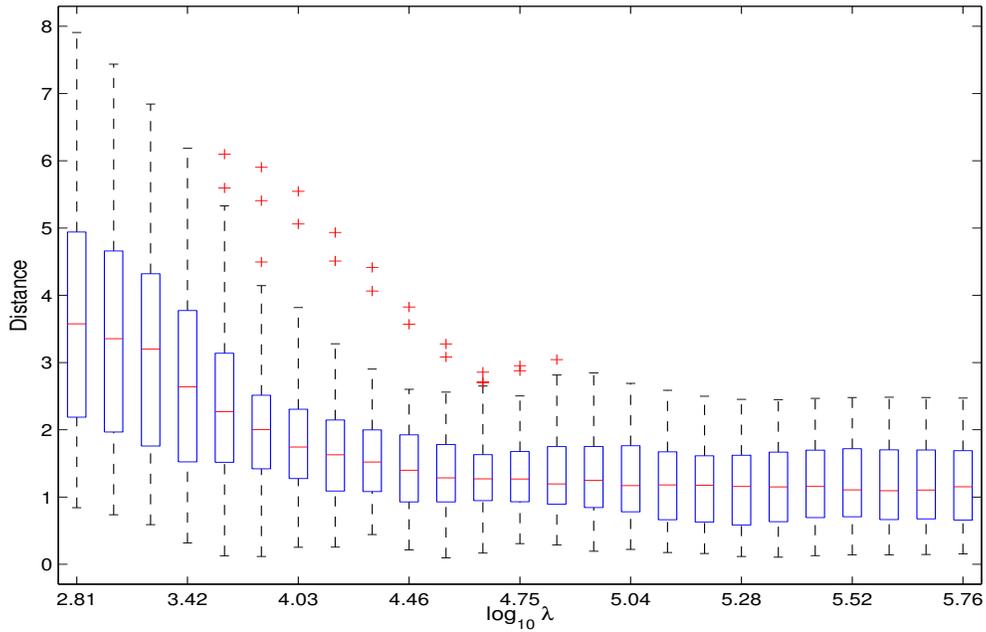
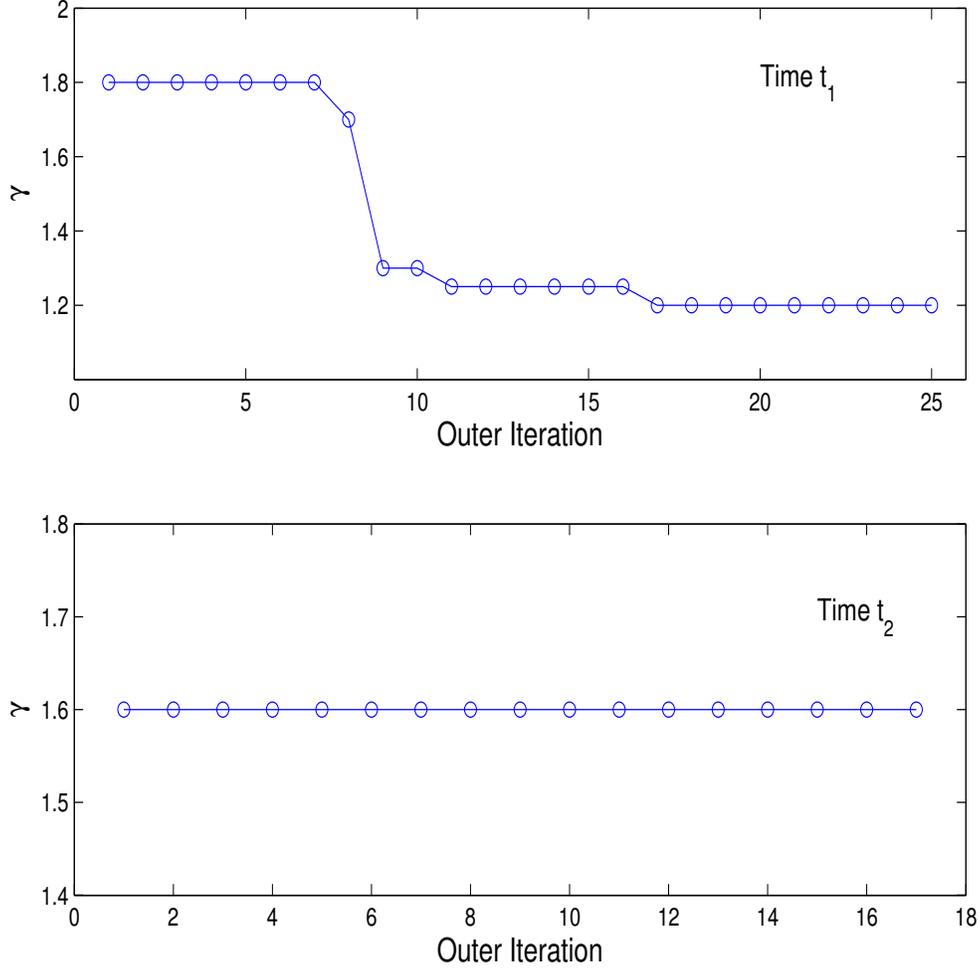Figure 6: Box plot of point distance $\{d(x, S_1), x \in \hat{S}_1\}$: red crosses are outliers.

Figure 7: Multipliers of λ for diffeomorphic matching multiple anterior leaflets at time t_1 and t_2.

6.3 Diffeomorphic Matching for Multiple Anterior Snapshots

We present the performances of diffeomorphic matching for 3 snapshots S_0, S_1, S_2 acquired at time $t_0 = 0$, $t_1 = \frac{4}{9}$ and $t_2 = 1$. Starting at t_0, $\hat{S}_1$ is computed to match S_0 and S_1. Then for the time interval $[t_1, t_2]$, we match $\hat{S}_1$ and S_2 to complete the trajectories. The scale and termination parameters are listed in Table 2.

λ is initialized as 200 and 100 respectively for t_1 and t_2 and values of γ are displayed in the Figure 7 for each outer iteration.

We observe similar results in Figure 8. As λ gets larger, for both deformations computed at t_1 and t_2, kinetic energy increases in sacrifice to higher matching quality.

For computed deformation $\hat{S}_1$ and $\hat{S}_2$, the two graphs in Figure 9 display several level curves for the point matching errors between $\hat{S}_1$ and S_1, and $\hat{S}_2$ and S_2. The coordinate system has been modified isometrically at each snapshot instant in order to display a better projection. From both figures, we notice the errors are smaller around the edges compared to the region inside. Based on our observation of numerical results, we discover that the edges always get matched faster with measure matching.

In Figure 10, the 3 closed curves are 3 successive snapshots of anterior leaflet boundary. The dotted curves are the computed trajectories.
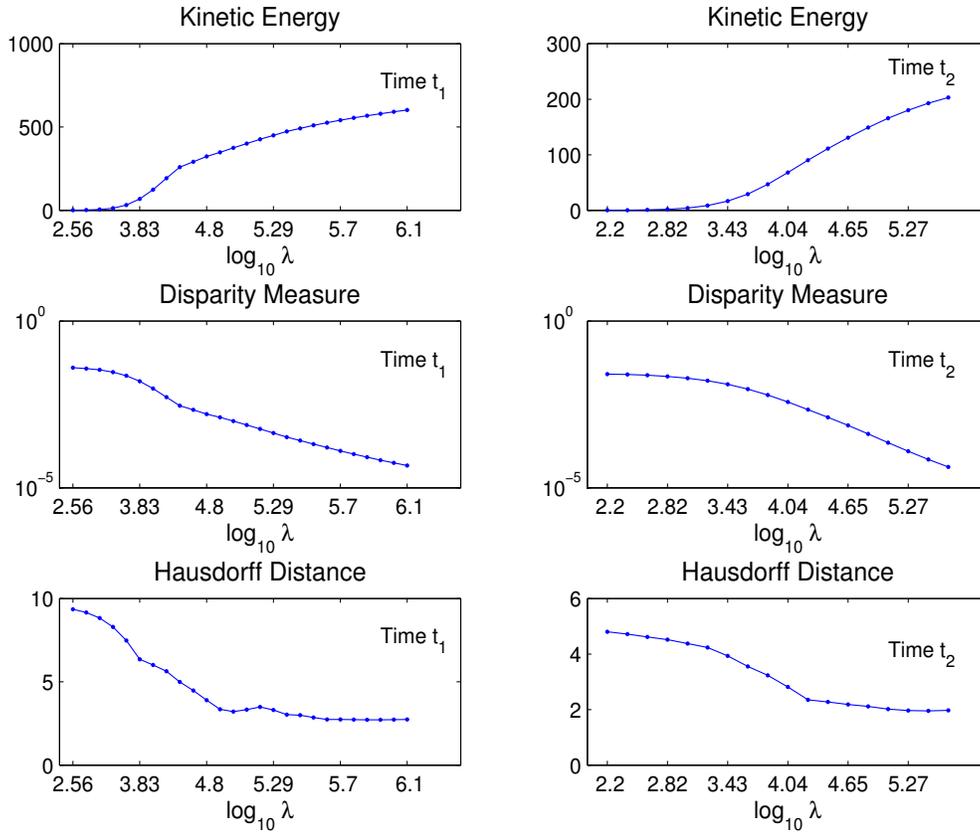
Figure 8: Behaviors of kinetic energy, measure matching, and Hausdorff distance for diffeomorphic matching multiple anterior leaflets at time t_1 and t_2.
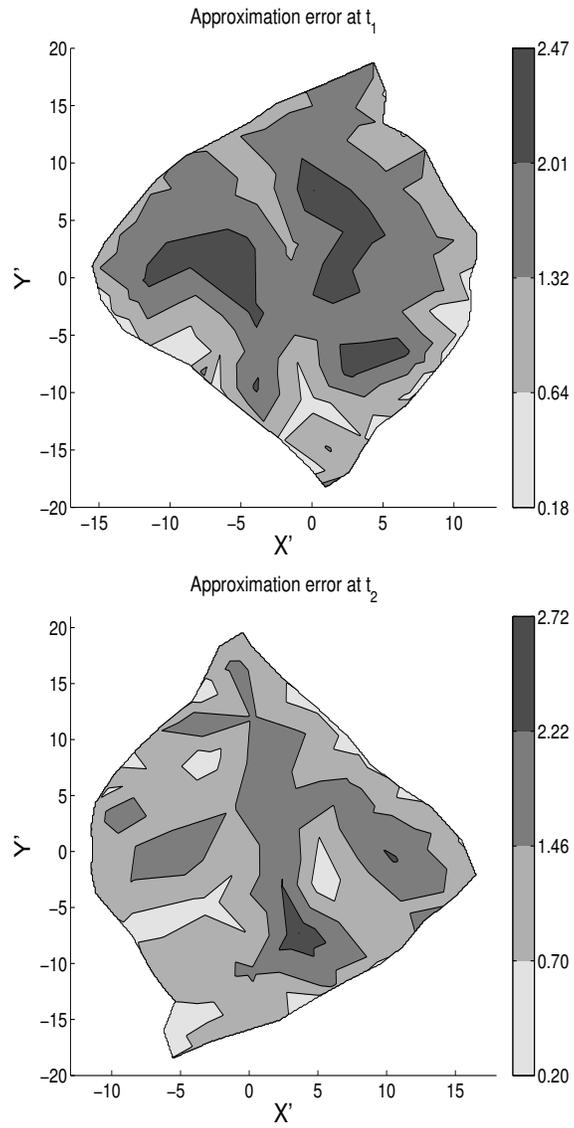
Figure 9: Matching errors between the computed anterior leaflet deformations and the snapshots at t_1 and t_2.
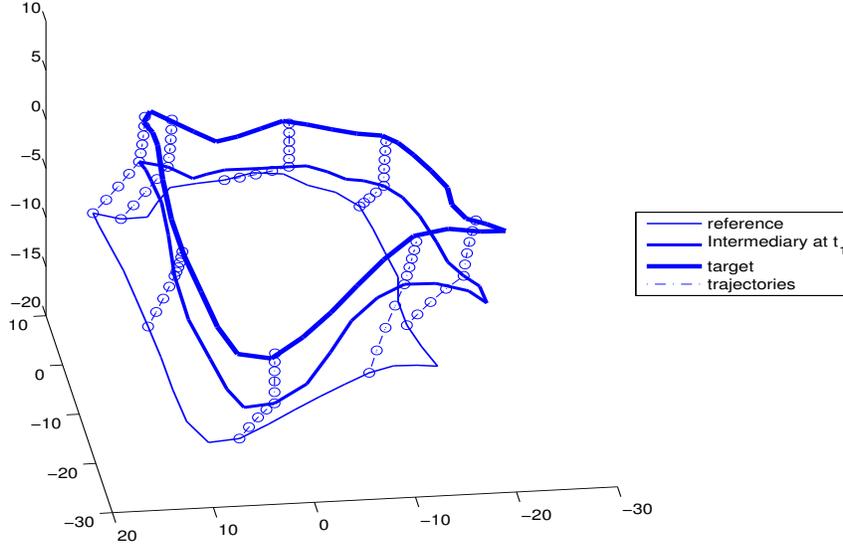
Figure 10: Computed deformations of the anterior leaflet boundary for instants t_0, t_1, t_2: we use circles to outline trajectories of a few selected points.

Parameter	Notation	$[t_0, t_1]$	$[t_1, t_2]$
N	Cardinal number of reference set	83	83
M	Cardinal number of target set	82	84
σ	Scale parameter of the dynamic system	4.13	4.13
L	Time steps	5	6
ϵ	Stoping criterion of Newton decrement	1.d-8	1.d-8
Err	Stoping criterion of measure matching	5.d-5	5.d-5

Table 2: Parameters and values for diffeomorphic matching multiple anterior leaflets.

Figure 11 and 12 display the reference, target and computed deformations at instants t_1, t_2.

6.3.1 Measure Matching for Posterior Leaflets

The data computed here is the posterior leaflet of a patient. The reference consists of 727 points while target set consists of 4261 points. We apply the same variational techniques to this data set and obtain very similar results. However, since the data set is rather large, the actual CPU time is 1.66×10^3. The disadvantage of the second order method is now very noticeable due to the computation involving Hessian matrices. Later we develop the multi-scale method in section 7 to improve the efficiency.
The scale and termination parameters are listed in Table 3.
The regularization parameter λ is initialized as 20 and multiplied by 1.6 at each outer iteration.
 Even though the cardinal number is extremely larger than the experiments in the former context, the Newton decrements behave very similarly as in Figure 13. The number of iterations is still around 5 and independent of the size of data set.
 With dramatically increased cardinal number, we still observe that the Pareto curve is perfectly convex as shown in Figure 14.
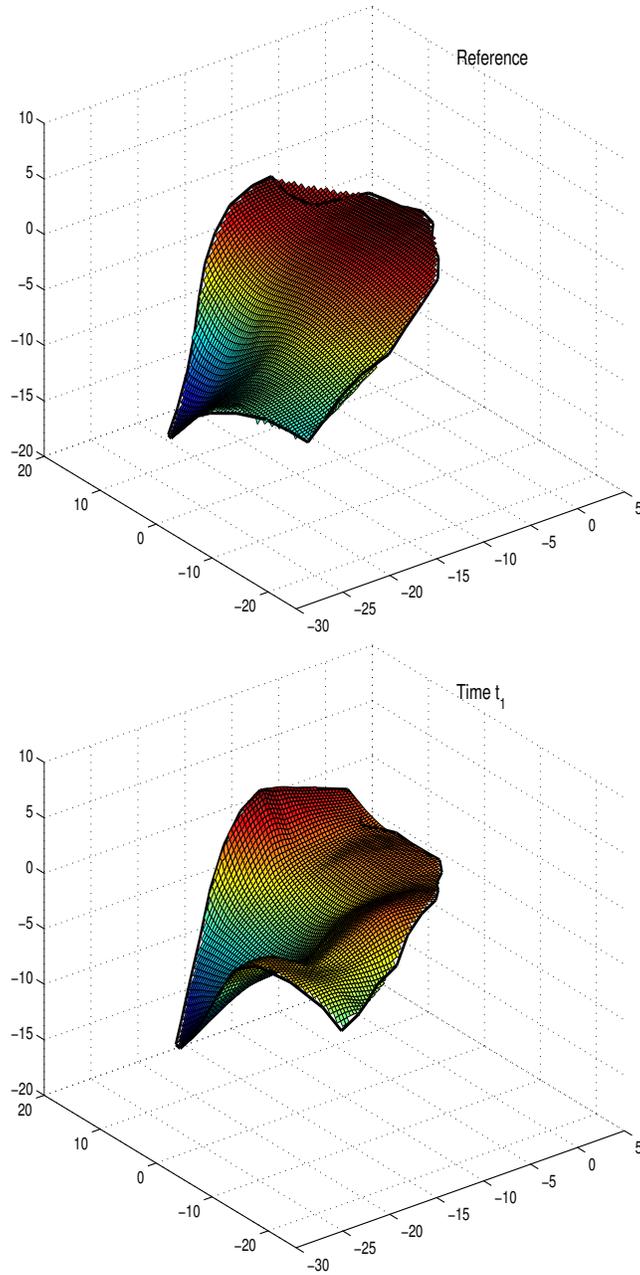 In Figure 15, we still obtain very similar results as before. As we increase the trade-off parameter λ,

Figure 11: Reference shape and computed deformations at time t_1.

Parameter	Notation	Value
N	Cardinal number of reference set	727
M	Cardinal number of target set	4261
σ	Scale parameter of the dynamic system	2.20
L	Time intervals	2
ϵ	Stoping criterion of Newton Decrement	1.d-8
Err	Stoping criterion of measure matching	5.d-5

Table 3: Parameters and values for diffeomorphic matching two posterior leaflets.
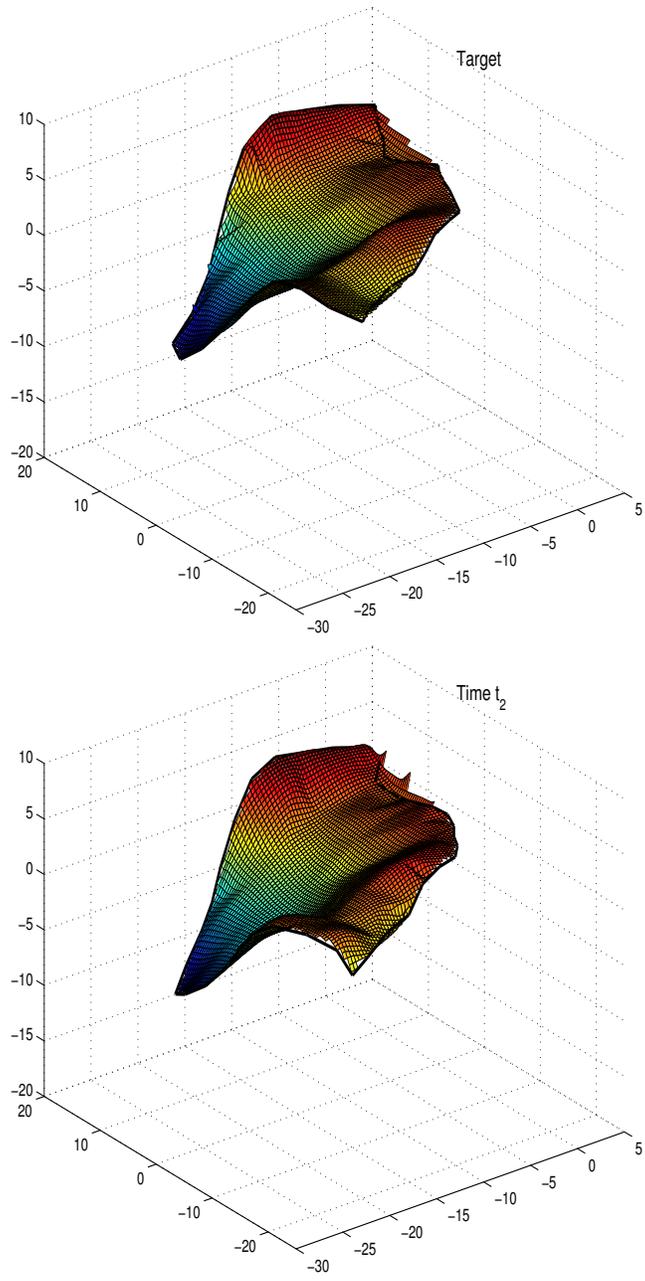
Figure 12: Target shape and computed deformations at time t_2.

41
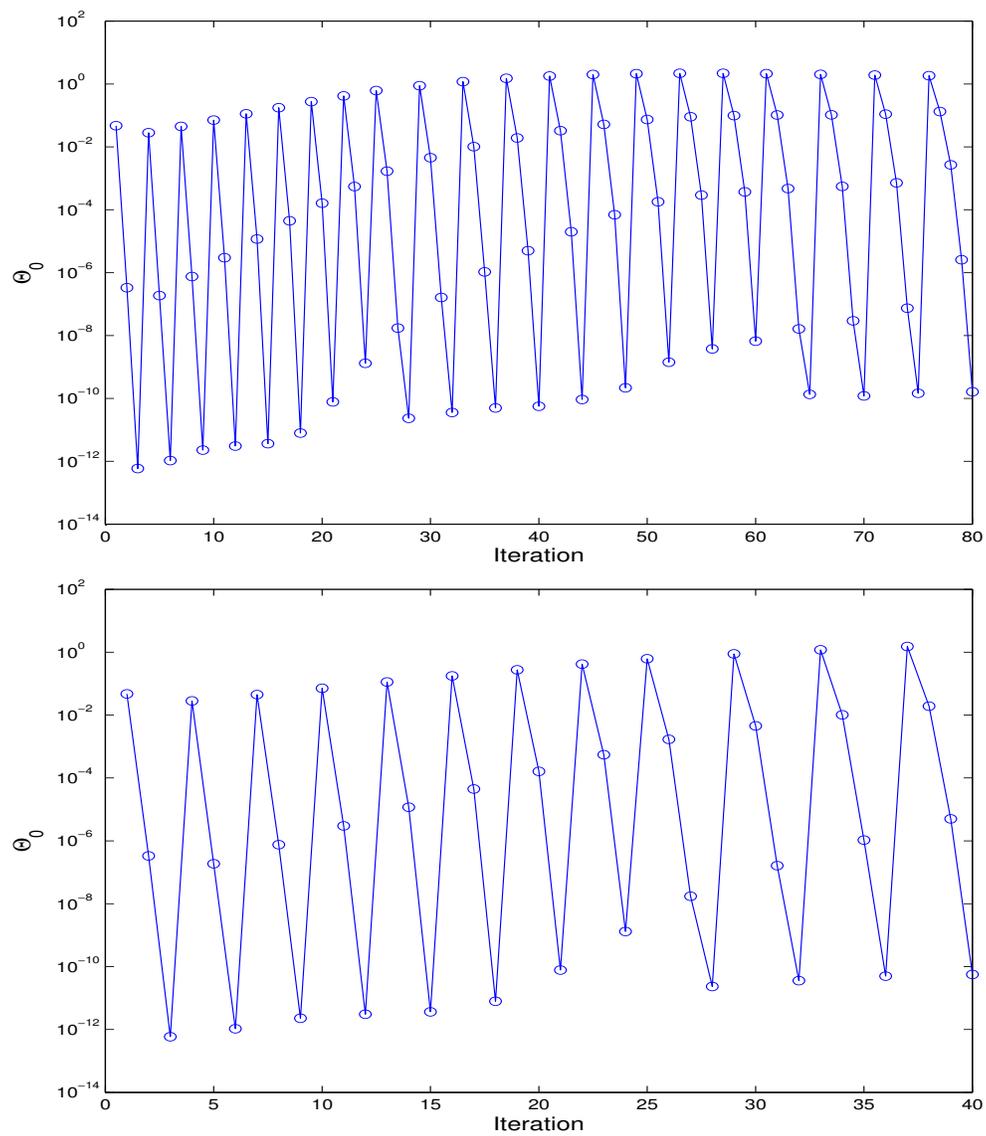
Figure 13: Newton Decrement for diffeomorphic matching two posterior leaflets.
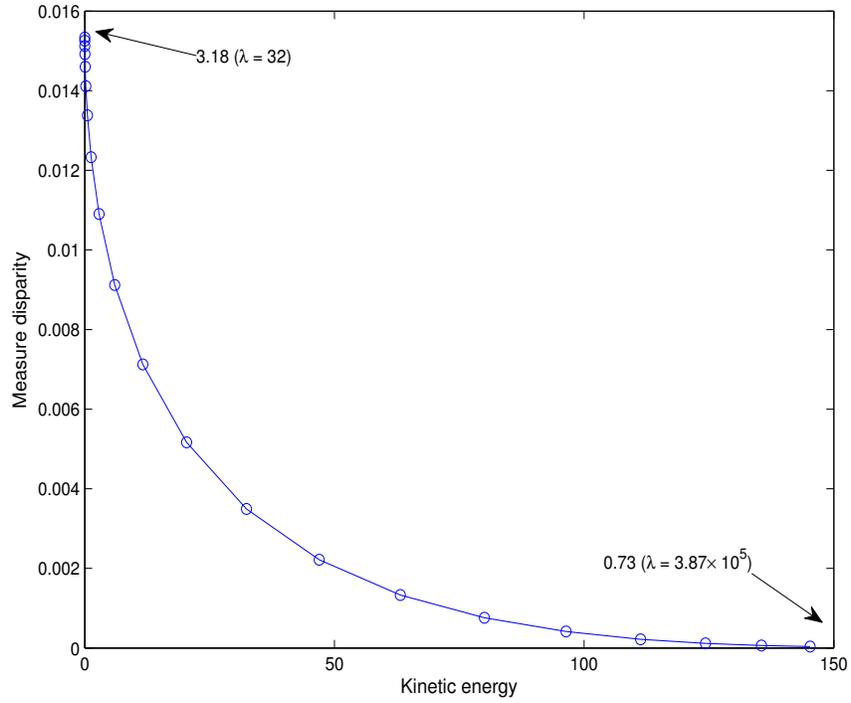
Figure 14: Pareto optimality for diffeomorphic matching two posterior leaflets.
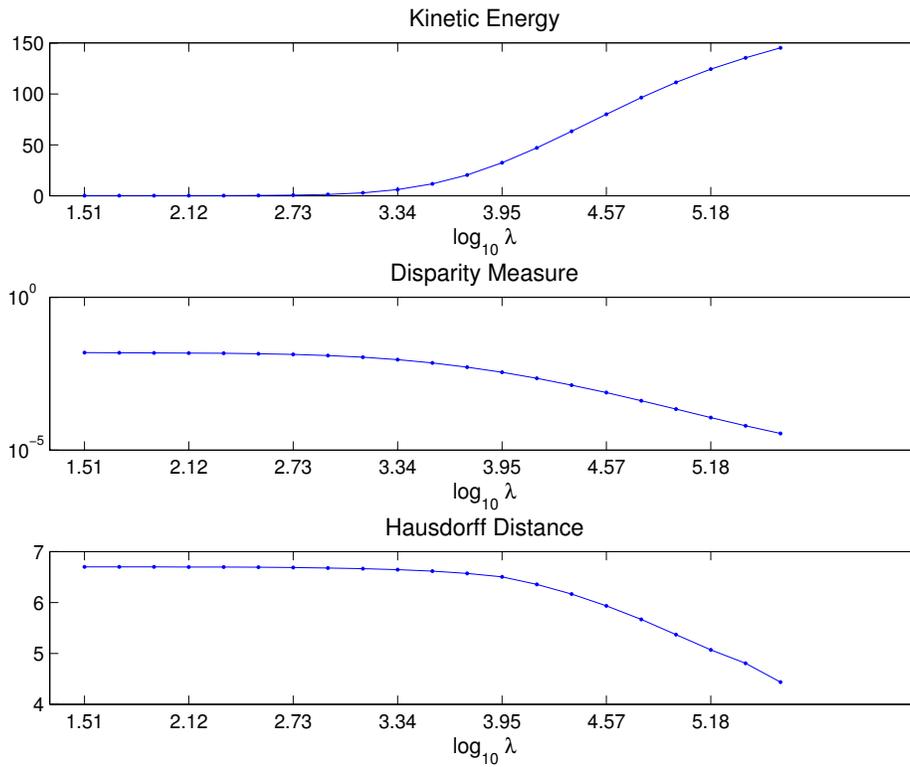

Figure 15: Behaviors of kinetic energy, measure matching, and Hausdorff distance for diffeomorphic matching two posterior leaflets.
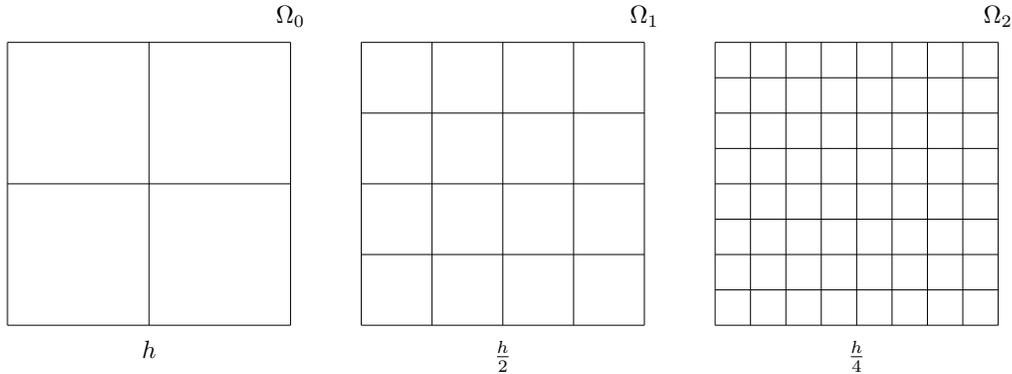
Figure 16: Increasingly finer meshes.

both the measure disparity function and Hausdorff distance are minimized in sacrifice of more kinetic energy.

7 Multi-scale Method

From the first two experiments in Section 6, we discover that the second order method has very strong advantages. First of all, the convergence of the second order method is quadratic near optimal solution and it needs at most six or so iterations to achieve the stopping criterion of high accuracy. Moreover, it performs similarly with different sizes, i.e., iterations needed for convergence do not depend on data sizes.

However, as to large data sets, disadvantages of the second order method become noticeable. Computing and storing Hessian matrices is very expensive, as well as propagations involving Hessian matrices. If the reference data set consists of N points, the number of operations required is on the order of N^3. Besides, the convex convergence region is rather small, so for larger sets, we are under a huge risk of losing positive definite property which is crucial to the method.

To overcome all these disadvantages, we are introducing the multi-scale method. Since at the very beginning, the continuation parameter, λ, is very small and thus the matching quality is not good at all, there is no point in dealing with a humongous data set. Therefore, we want to find a new representation which consists of fewer points and retains the data structure to some degree. Then with finer scales, the original problem can be solved.

7.1 Multi-scale and Representation

The concept of multigrid has been used for the numerical solution of diffusion and convection-diffusion problem [24], and there are various approaches in image analysis problems taking advantage of multigrid relaxation methods [71, 68]. For the multi-scale method to be discussed, we consider a set of increasingly finer meshes (Figure 6.1). We start out with a coarse mesh size h and reduce it by half step by step. For each step, one point is chosen to represent all the points in one block. The essential idea of the method is to refine the grids recursively step by step. In each refining step, a new approximately equivalent optimization problem will be defined, increasing the number of points in the data set by reducing the mesh size to 1/4 or 1/2 of the former mesh size. We construct the coarser problems at the beginning,
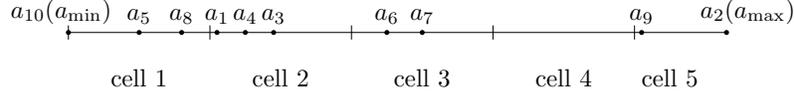
44

Figure 17: An example of the one-dimensional case.

when λ is small and the matching quality does not concern, to avoid unnecessary computing involving large Hessian matrices and hence to speed up the programming.

Various options can be applied for the point interpolation. For example, one might pick the point with the most neighboring points. We now explain our version of representation.

First, let us consider the one-dimensional case, a class of points $\{a_i\}_{i=1}^n \in \mathbb{R}$, and let

$$a_{\max} = \max_i\{a_i\}, \ a_{\min} = \min_i\{a_i\}.$$

For any given mesh size h, define

$$q_i = [\frac{a_i - a_{\min}}{h}] + 1, \quad \text{for any } i = 1, 2, \cdots, n,$$

and thus we locate a_i in the q_i^{th} cell. Furthermore, the total number of cells is

$$Q = [\frac{a_{\max} - a_{\min}}{h}] + 1,$$

where $[\cdot]$ rounds toward the minus infinity.

See Figure 6.2 for an example. 5 cells are formed based on the data set $\{a_i\}_{i=1}^{10}$ with given mesh size h. If there is no point in one cell, e.g., cell 4, then no substitution is needed for that cell.

Now we introduce new representation in 3D with mesh size h enforced. Consider a data set $\{\mathbf{x}_i\}_{i=1}^N \in \mathbb{R}^3$, apply the same strategy to each coordinate. Other than the one-dimensional case, now for $\mathbf{x}_i$, we have $\mathbf{q}_i = (q_i^x, q_i^y, q_i^z)$ as its location and $Q = Q_x \times Q_y \times Q_z$ total cells. Once all the points are properly located, get an average of each cell, which is

$$\hat{\mathbf{x}}_q = \frac{1}{|I_q|} \sum_{i \in I_q} \mathbf{x}_i, \tag{95}$$

where $I_q = \{\text{index } i : \mathbf{x}_i \text{ belongs to cell } q\}$, $q = 1, 2, \cdots, Q$. Thus, we obtain $\Omega_h = \{\hat{\mathbf{x}}_q\}_{q=1}^Q$ as a new representation of data set $\Omega = \{\mathbf{x}_i\}_{i=1}^N$.

Note that the actual number of points in Ω_h could be less than Q considering that some cell might not have any points at all. As an example, consider the data given in Figure 18. The region is divided into 12 cells and each red star represents an average point of each cell. Cell (1,1) is empty and thus no representation is needed.

Ideally a suitable scale parameter $\sigma > 0$ for the Gaussian kernel K_σ in (90) should be introduced to incorporate with new resolution. However, numerical experiments show that fixating σ as the original scale parameter is very acceptable. Keeping σ fixed smooths the way into a finer grid from a coarser one. Besides, at the end, with all the points taken into account, σ is the scale parameter of the original data set anyway.
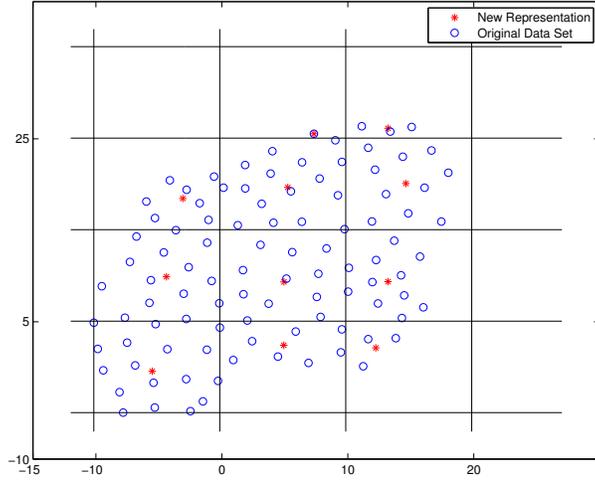
45

Figure 18: New representation and original data of a two-dimensional case.

7.2 Updates Between Two Different Scales

Let $h_1 > h_2 > \cdots > h_R > 0$ be a sequence of increasingly finer mesh sizes and denote $\Omega_r = \Omega_{h_r}$ for $r = 1, 2, \cdots, R$, where h_R is small enough such that Ω_R is nothing but the original data set Ω.

Now consider reference data set $\Omega^{\text{ref}} = \{x_i\}_{i=1}^N \in \mathbb{R}^3$ and target data set $\Omega^{\text{tar}} = \{y_j\}_{j=1}^M \in \mathbb{R}^3$. For a coarser mesh h_r, by (95), generate

$$\Omega_r^{\text{ref}} = \{\mathbf{x}_i^r\}_{i=1}^{N_r}, \quad \Omega_r^{\text{tar}} = \{\mathbf{y}_j^r\}_{j=1}^{M_r},$$

where $r = 1, \cdots, R - 1$. Then the new approximately equivalent optimization problem is diffeomorphic matching of Ω_r^{ref} and Ω_r^{tar}. Since the size of Ω_r^{ref} and Ω_r^{tar} is noticeably smaller than that of Ω^{ref} and Ω^{tar}, the computing time is cut down tremendously, especially at the first couple of rescaling steps. At the same time, the matching quality is not compromised, considering that

$$\Omega_R^{\text{ref}} = \Omega^{\text{ref}}, \quad \Omega_R^{\text{tar}} = \Omega^{\text{tar}},$$

where the original data sets are utilized at the final scaling step R.

Apply the second order method discussed in section 5 to match Ω_r^{ref} and Ω_r^{tar} and it yields an optimal solution

$$\boldsymbol{\alpha}_r^* = (\bar{\alpha}_1^r, \cdots, \bar{\alpha}_{N_r}^r)^T \in \mathbb{R}^{3N_r}, \qquad r = 1, 2, \cdots, R - 1.$$

Instead of initialing null control $\boldsymbol{\alpha}_{r+1}$ blindly for step $r+1$, we want to take advantage of optimal control $\boldsymbol{\alpha}_r^*$ at step r, $r = 1, 2, \cdots, R - 1$. With finer scale size h_{r+1}, we generate

$$\Omega_{r+1}^{\text{ref}} = \{x_i^{r+1}\}_{i=1}^{N_{r+1}}, \quad \Omega_{r+1}^{\text{tar}} = \{y_j^{r+1}\}_{j=1}^{M_{r+1}},$$

46

as the new reference and target set respectively at rescaling step $r + 1$. Also, let

$$\boldsymbol{\alpha}_{r+1} = (\alpha_1^{r+1}, \cdots, \alpha_{N_{r+1}}^{r+1})^T \in \mathbb{R}^{3N_{r+1}}.$$

Now we show how to initialize $\boldsymbol{\alpha}_{r+1}$ with optimal control $\boldsymbol{\alpha}_r^*$ using properties of the velocity field.

Theorem 7.1. *Given optimal solution $\boldsymbol{\alpha}_r^*$ associated Ω_r^{ref} for $r = 1, 2, \cdots, R - 1$, the initializer $\boldsymbol{\alpha}_{r+1}$ follows the update law*

$$\sum_{i=1}^{N_r} \bar{\alpha}_i^r K(x_j^{r+1}, x_i^r) = \sum_{i=1}^{N_{r+1}} \alpha_i^{r+1} K(x_j^{r+1}, x_i^{r+1}), \tag{96}$$

where $K(\cdot, \cdot)$ is the radial Gaussian kernel with scale parameter σ.

Proof. The solution of velocity is restricted to the set of linear combination of $K_{x_1^r}, \cdots, K_{x_{N_r}^r}$, which places us in a very comfortable finite-dimensional situation. Since $\boldsymbol{\alpha}_r^*$ is already the optimal control for scaling step r, we look for v under the form $v = \sum_{i=1}^{N_r} \bar{\alpha}_i^r K_{x_i^r}$, which may also be written as

$$v(x) = \sum_{i=1}^{N_r} \bar{\alpha}_i^r K(x, x_i^r),$$

for any $x \in \mathbb{R}^3$. (Refer to the left of Figure 19.)
Then consider $\Omega_{r+1}^{ref} = \{\mathbf{x}_i^{r+1}\}_{i=1}^{N_{r+1}} \in \mathbb{R}^3$, and the velocity at each position is given by

$$v(x_j^{r+1}) = \sum_{i=1}^{N_r} \bar{\alpha}_i^r K(x_j^{r+1}, x_i^r), \tag{97}$$

for any $j = 1, \cdots, N_{r+1}$.
Similarly, we also find the velocity for $\Omega_{r+1}^{ref} = \{x_i^{r+1}\}_{i=1}^{N_{r+1}}$ with finer mesh as

$$v(x) = \sum_{i=1}^{N_{r+1}} \alpha_i^{r+1} K(x, x_i^{r+1}), \quad x \in \mathbb{R}^3,$$

(Refer to the right of Figure 19.) except that α_i^{r+1} here are unknown. Furthermore, the velocity at any individual point is

$$v(x_j^{r+1}) = \sum_{i=1}^{N_{r+1}} \alpha_i^{r+1} K(x_j^{r+1}, x_i^{r+1}), \tag{98}$$

for $j = 1, \cdots, N_{r+1}$.
  Both equation (97) and (98) describe the velocity and they are approximately the same. Thus, for $j = 1, \cdots, N_{r+1}$,

$$\sum_{i=1}^{N_r} \bar{\alpha}_i^r K(x_j^{r+1}, x_i^r) \doteq \sum_{i=1}^{N_{r+1}} \alpha_i^{r+1} K(x_j^{r+1}, x_i^{r+1}).$$
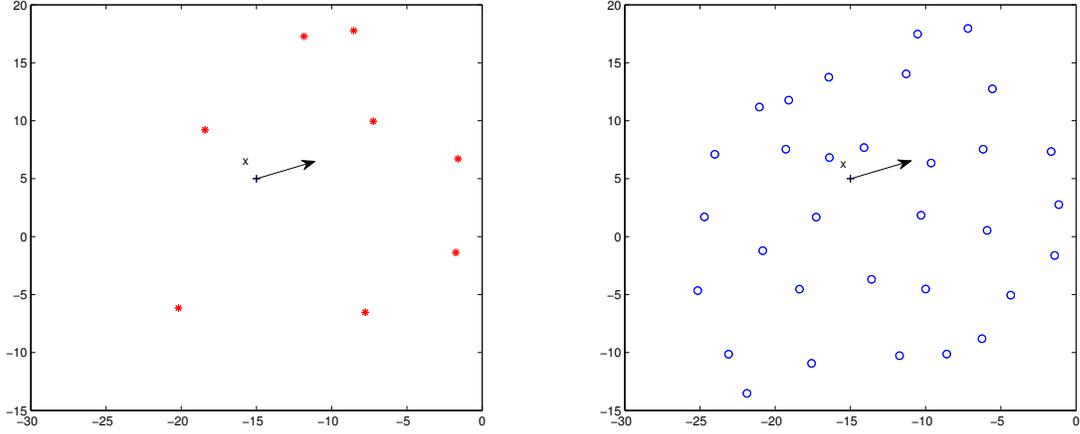
$\square$

Figure 19: Velocity fields with respect to finer mesh size (the left) and coarser mesh size (the right).

Introduce new notations,

$$(\mathbf{K}_r)_{ij} = (K(x_i^r, x_j^r)), \quad (\mathbf{K}_{r,r+1})_{ij} = K(x_i^r, x_j^{r+1}),$$

and the update law (96) can be written as

$$\mathbf{K}_{r,r+1}\boldsymbol{\alpha}_r^* = \mathbf{K}_r\boldsymbol{\alpha}_{r+1}.$$

Furthermore, since $\mathbf{K}_r$ is a positive definite matrix, we have

$$\boldsymbol{\alpha}_{r+1} = \mathbf{K}_r^{-1}\mathbf{K}_{r,r+1}\boldsymbol{\alpha}_r^*, \quad r = 1, \cdots, R-1. \tag{99}$$

7.3 Programming

We have the algorithm laid out as follows

Step 1 (Initialization of scaling)

Specify $\beta_0 \in (0,1)$. Set $r = 0$, $h_r = \beta_0 \text{Disp}(\mathbf{x}, \mathbf{y})$, and then obtain new presentation $\Omega_r^{\text{ref}}, \Omega_r^{\text{tar}}$.

Step 2 (Initialization of continuation)

Initialize $\lambda_0 > 0$ and set $\nu = 0$.

Step 3 (Initialization of Newton's method)

$\boldsymbol{\alpha}_\nu^{(0)} = \mathbf{0}$ and $\mu = 0$.

Step 4 (The second order method)

Set $\mu = \mu + 1$ and compute $\boldsymbol{\alpha}_\nu^\mu$ by the second order method. If the stopping criterion

$$\sqrt{-\Theta_0} < \epsilon,$$

is satisfied, go to step 5.

Step 5 (Termination of continuation)

If the termination criterion $\mu > \text{nMax}$ or $\phi(\mathbf{x}_\nu^\mu) < \epsilon$ is satisfied, go to step 6.

Otherwise, set $\nu = \nu + 1$, $\boldsymbol{\alpha}_\nu^{(0)} = \boldsymbol{\alpha}_\nu^{(\mu)}$, and increase the regularization parameter,

$$\lambda_\nu = \gamma_\nu \lambda_{\nu-1},$$

and go to step 4.

Step 6 (Rescaling)

If $r > \text{rMAX}$, stop algorithm.

Otherwise, set $r = r + 1$, $h_r = \beta h_{r-1}$. Get $\Omega_r^{\text{ref}}, \Omega_r^{\text{tar}}$ and obtain $\boldsymbol{\alpha}_0^{(0)}$ by (99). Set $\lambda_0 = \lambda_\nu$. Then, reset $\mu = \nu = 0$ and go back to step 4.

We use the Hausdorff distance as the disparity function $\text{Disp}(\cdot, \cdot)$ in **step 1**. However, other disparity function could substitute as long as the disparity function chosen displays the geometric distance between the reference and target set. As to the rescaling parameter β, theoretically any number belongs $(0, 1)$ is suitable, but for the best, it is preferable that the number of points with the new scale increases not too slowly. However, if the size of the new data set increases too much, the initializer obtained by (99) would not do too much good because the data structure changes too much. Based on our experiments, it is ideal to double or triple the data size until the size of the original data sets is reached.

In **step 6**, if rescaling is needed, the continuation parameter λ is inherited from previous programming and our numerical experiments suggest it is very important to do so. For each scaling process, the optimization problem is slightly different but ultimately similar. With the continuation method built inside, it is even harder to jump from one scale to another. Fixating λ allows the optimization problem staying consistent and stabilized.

7.4 Numerical Results

7.4.1 Diffeomorphic Matching for Posterior Leaflets

We apply the multi-scale method to diffeomorphic matching of 2 posterior leaflet snapshots S_0 and S_1 acquired at instants t_0 and t_1. Cardinals for S_0 is 406 and that for S_1 is 1702. The scale parameter σ of the Gaussian kernel is 2.57. The continuation parameter λ is initialized at 10 and increased by multiplying 1.6 at each outer iteration. The parameters β are 0.5, 0.5 and 0.1 respectively for each

rescaling step. Till reaching the resolution of the original data set, 3 rescaling steps are used with CPU time 249 seconds.

Figure 20 displays the Pareto optimality curves. For the first two reformatted data, cardinals of them are 88 and 237 and the original data is utilized at the final rescaling step. Convexity is observed of each Pareto frontier generated at each rescaling step.

We display several level curves for the point matching errors between $\hat{S}_1$ and the given posterior leaflet S_1 in Figure 21.
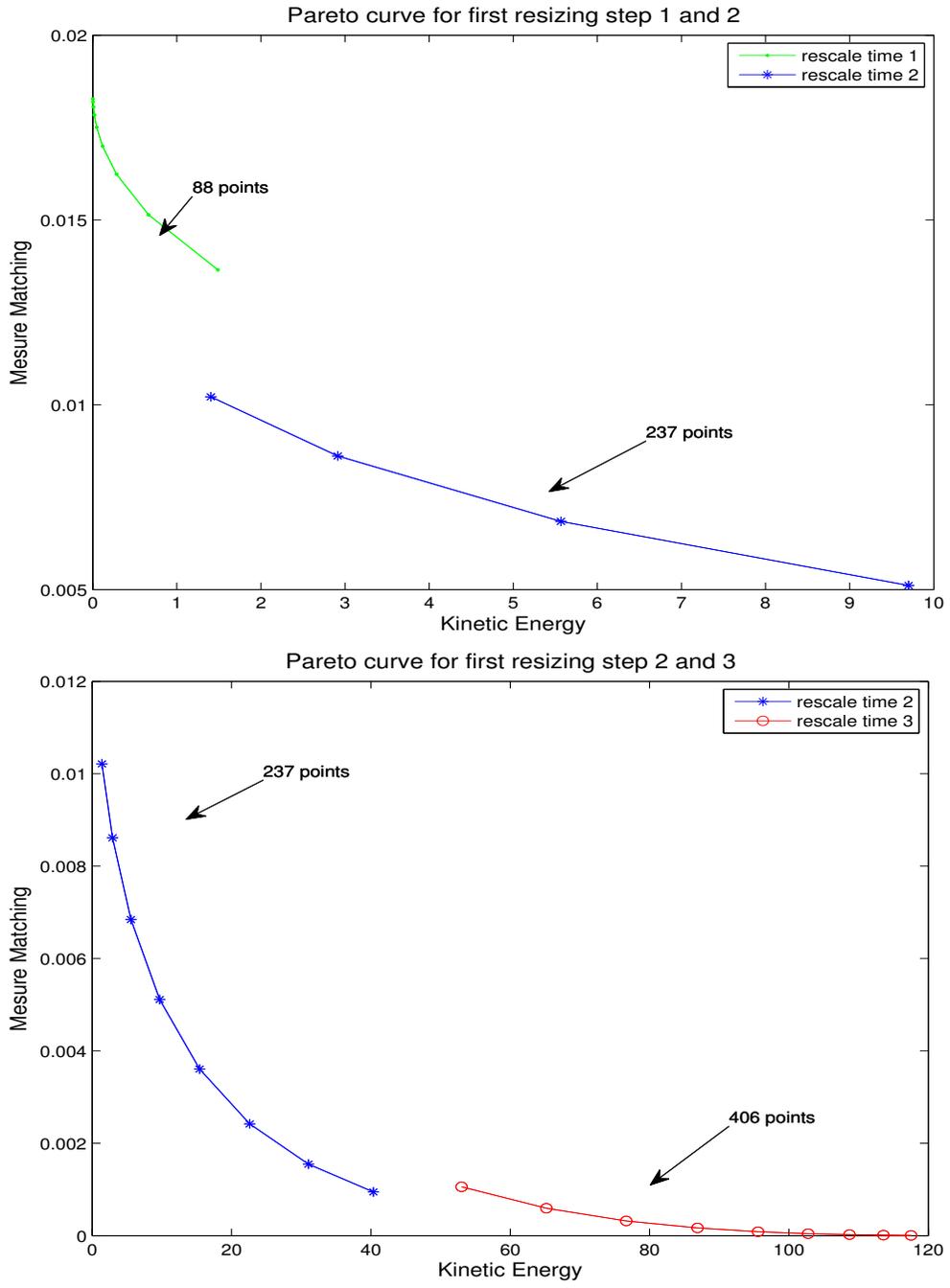
Figure 20: Pareto optimality curves using the multi-scale method: for easier visualization, the iterations are displayed separately at rescaling step 2.
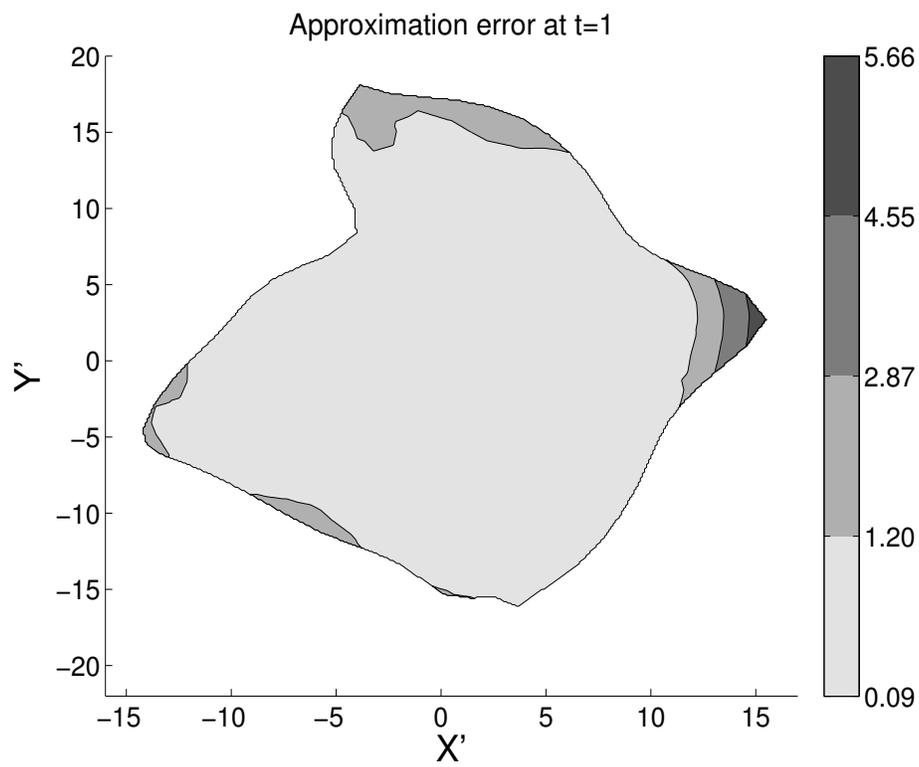
Figure 21: Matching errors between the computed posterior leaflet deformation $\hat{S}_1$ and the target shape S_1.

Parameter	The second order Method	Multi-scale method
σ	2.20	2.20
Time intervals	1	1
Measure matching	3.49×10^{-5}	2.94×10^{-5}
Kinetic energy	145	166
λ	2.42×10^5	4.40×10^5
Number of outer iterations	21	23
CPU time	1660	489

Table 4: Parameter and values of the second order method and the multi-scale method.

7.4.2 Improvement Compared to the Second Order Method

Now we apply the multi-scale method to the same data in section 6.3.1 and compare the numerical results. The reference consists of 727 points while target set consists of 4261 points. The continuation parameter λ_0 is initialized at 20 for both methods.

Table 4 demonstrates that to attain similar matching quality, the multi-scale method uses less than one-third the time as with the same setting as the second order method. The number of outer iterations for the multi-scale method is slightly more than that of the second order method. However, since the original data set is only utilized for the last four outer iterations, computing time is tremendously reduced.

In Figure 22, the approximate Pareto frontiers displays measure matching as functions of the kinetic energy and we observe convexity of these curves. It also shows that most of the outer iterations works on data with coarser meshes.

As shown in Figure 23 and 24, the geometric distance is very comparable.

8 Matching with Elasticity Energy

In our previous experiments, we discover that some area of a given data set tends to stretch too much to obtain a good matching. However, in real life, human tissue can not tolerate such enormous stress and ultimate enough stretching causes tissue to break. With those facts taken into account, we introduce the elasticity energy in the hope of reflecting the stiffness and stretchiness of human tissue.

8.1 Stretching Fraction and Elasticity Energy

Definition 8.1. Imagine two points $\mathbf{x}^{(0)}$ and $\mathbf{y}^{(0)}$ are marked on an elastic rope, and they relocate at new positions $\mathbf{x}^{(1)}$ and $\mathbf{y}^{(1)}$ respectively as the rope stretches (Figure 7.1). Define the constant

$$\frac{|\mathbf{x}^{(1)} - \mathbf{y}^{(1)}|}{|\mathbf{x}^{(0)} - \mathbf{y}^{(0)}|},$$

as the stretching fraction.

For surfaces in 3D, one point is not isolated and the stretching at one point heavily depends on the movement of its neighborhood. Neighborhood is one of the basic concepts in a topological space which is closely related to concepts of open set and interior. In our case, since the movement of the closest points reflects the stretching level at a point, the neighborhood is designed to reveal the fact.

With mesh cells, we define $\mathcal{N}(\mathbf{x}_i)$, the neighborhood of $\mathbf{x}_i$, as the points in the closest cells to the cell
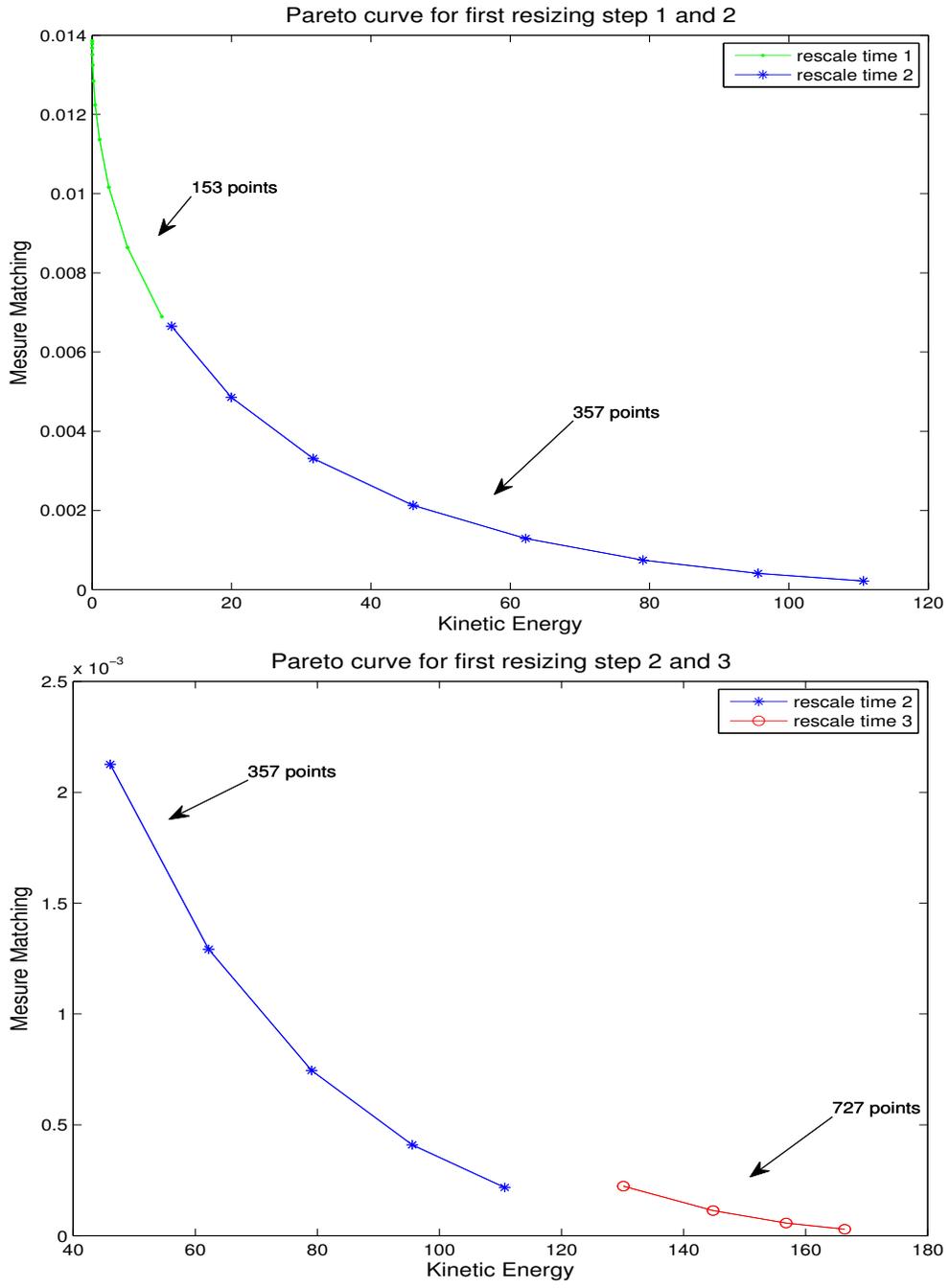
Figure 22: Pareto optimality curves for two posterior leaflets using the multi-scale method: for easier visualization, the iterations are displayed separately at rescaling step 2.
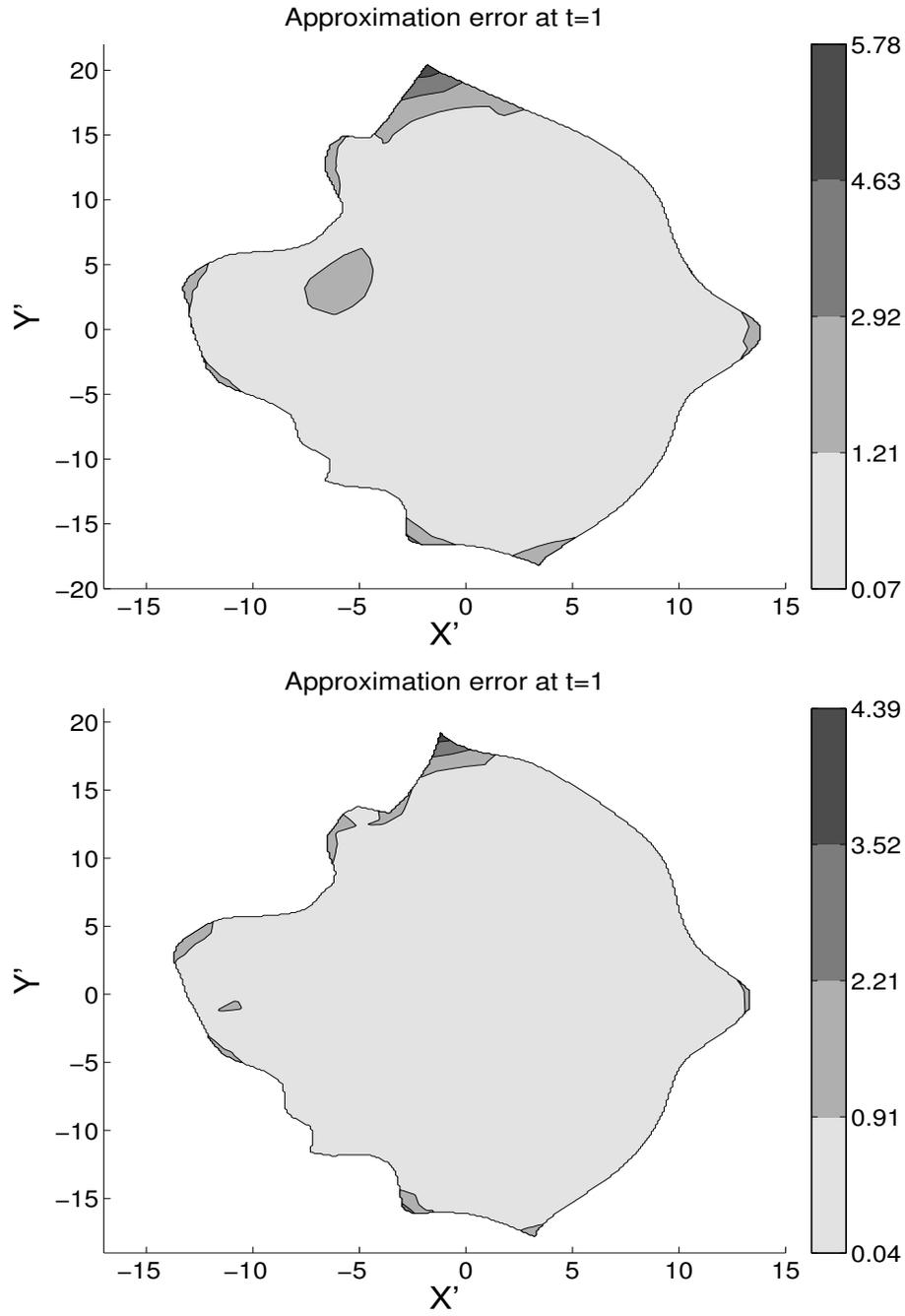
Figure 23: Matching errors between the computed anterior leaflet deformation $\hat{S}_1$ and S_1 using the multi-scale method (top) and the second order method (bottom).
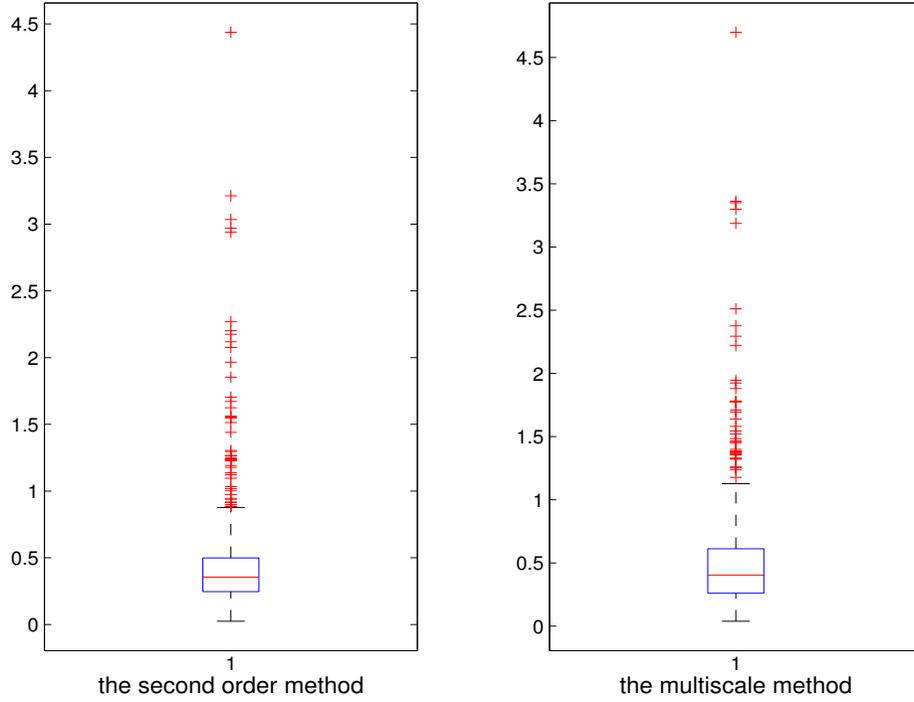
Figure 24: Box plot of point matching errors between the computed anterior leaflet deformation $\hat{S}_1$ and S_1: for each box in the figure, the central mark is the median, the edges of the box are the 25th and 75th percentiles and outliers are plotted individually.
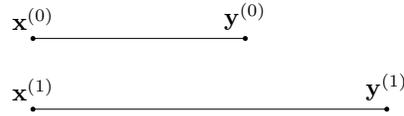

Figure 25: Definition of stretching fraction.

where $\mathbf{x}_i$ belongs. By using this definition, if $\mathbf{x}_i \in \mathcal{N}(\mathbf{x}_j)$, $\mathbf{x}_j \in \mathcal{N}(\mathbf{x}_i)$. Furthermore, the elasticity energy is well posed considering that the closest points are taking into account (Figure 7.2).

Definition 8.2. Let $\{x_i(t)\}_{i=1}^N$ be displacements of a collective data set at time t, and introduce

$$E(t) = \sum_{i=1}^{N} \left(\sum_{x_j(t) \in \mathcal{N}(x_i(t))} \frac{|x_i(t) - x_j(t)|^2}{|x_i(0) - x_j(0)|^2} \right) \tag{100}$$

to be the elasticity energy where $\mathcal{N}(x_i(t))$ stands for the neighborhood of point $x_i(t)$.
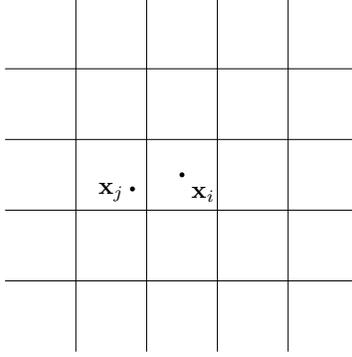
56

Figure 26: Neighborhood of $\mathbf{x}_i$, $\mathcal{N}(\mathbf{x}_i)$, in two-dimensional space.

8.2 Variational Forms

To simplify (100), introduce the function

$$I^t_{\mathcal{N}_i}(j) = \{ \begin{array}{ll} 1 & x_j \in \mathcal{N}(x_i(t)) \\ 0 & x_j \notin \mathcal{N}(x_i(t)) \end{array} ,$$

and then rewrite the elasticity energy as

$$E(t) = \sum_{i=1}^{N} \sum_{j=1}^{N} I^t_{\mathcal{N}_i}(j) \frac{|x_i(t) - x_j(t)|^2}{|x_i(0) - x_j^{(}0)|^2}. \tag{101}$$

For the sake of simplicity, we fix time and derive the variational forms only at time t. In order to apply the second order method, we now derive the variational forms of (101) where

$$\delta E = \sum_{i=1}^{N} \sum_{j=1}^{N} \frac{2}{|x_i(0) - x_j(0)|^2} I_{\mathcal{N}_i}(j)(x_i - x_j)^T(\delta x_i - \delta x_j)$$

$$= \sum_{i=1}^{N} \sum_{j=1}^{N} \frac{2}{|x_i(0) - x_j(0)|^2} I_{\mathcal{N}_i}(j)(x_i - x_j)^T \delta x_i - \sum_{i=1}^{N} \sum_{j=1}^{N} \frac{2}{|x_i(0) - x_j(0)|^2} I_{\mathcal{N}_i}(j)(x_i - x_j)^T \delta x_j.$$

Consider the second term in the formula above and we have

$$\sum_{i=1}^{N} \sum_{j=1}^{N} \frac{2}{|x_i(0) - x_j(0)|^2} I_{\mathcal{N}_i}(j)(x_i - x_j)^T \delta x_j$$

$$= \sum_{i=1}^{N} \sum_{j=1}^{N} \frac{2}{|x_i(0) - x_j(0)|^2} I_{\mathcal{N}_j}(i)(x_j - x_i)^T \delta x_i$$

$$= -\sum_{i=1}^{N} \sum_{j=1}^{N} \frac{2}{|x_i(0) - x_j(0)|^2} I_{\mathcal{N}_i}(j)(x_i - x_j)^T \delta x_i.$$

Thus,

$$\delta E = 4 \sum_{i=1}^{N} \sum_{j=1}^{N} \frac{1}{|x_i(0) - x_j(0)|^2} I_{\mathcal{N}_i}(j)(x_i - x_j)^T \delta x_i \tag{102}$$

which implies

$$(\nabla E)_i = 4 \sum_{j=1}^{N} \frac{1}{|x_i(0) - x_j(0)|^2} I_{\mathcal{N}_i}(j)(x_i - x_j). \tag{103}$$

Furthermore,

$$\left((\nabla E)_i\right)_x = 4 \sum_{j=1}^{N} \frac{1}{|x_i(0) - x_j(0)|^2} I_{\mathcal{N}_i}(j)(\delta x_i - \delta x_j)$$

$$= (4 \sum_{j=1}^{N} \frac{1}{|x_i(0) - x_j(0)|^2} I_{\mathcal{N}_i}(j)I)\delta x_i - 4 \sum_{j=1}^{N} \frac{1}{|x_i(0) - x_j(0)|^2} I_{\mathcal{N}_i}(j)I\delta x_j \tag{104}$$

where I is the identity matrix.

8.3 Dynamic Programming

To appropriately mimic the motion of human tissue, we include the elasticity energy into the minimization function and now the discrete dynamic system becomes

$$\begin{aligned} \min \quad & J(\{\boldsymbol{\alpha}_k\}_{k=0}^{L-1}; \bar{\mathbf{x}}_0) = \sum_{k=0}^{L-1} g(\mathbf{x}_k, \boldsymbol{\alpha}_k; t_k) + b \sum_{k=1}^{L} E(\mathbf{x}_k; t_k) + \lambda \phi(\mathbf{x}_L) \\ \text{s.t.} \quad & \mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \boldsymbol{\alpha}_k; t_k), \quad k = 0, \cdots, L-1, \\ & \mathbf{x}_0 = \bar{\mathbf{x}}_0, \end{aligned} \tag{105}$$

where

$$\mathbf{f}(\mathbf{x}_k, \boldsymbol{\alpha}_k; t_k) = \mathbf{x}_k + \tau_k \mathbf{K}(\mathbf{x}_k)\boldsymbol{\alpha}_k,$$

$$g(\mathbf{x}_k, \boldsymbol{\alpha}_k; t_k) = \frac{\tau_k}{2} \boldsymbol{\alpha}_k^T \mathbf{K}(\mathbf{x}_k)\boldsymbol{\alpha}_k,$$

$$\phi(\mathbf{x}_L) = \frac{1}{N^2} \sum_{i,j=1}^{N} K_{\sigma'}(x_i(t_L), x_j(t_L)) - \frac{2}{MN} \sum_{i=1}^{N} \sum_{j=1}^{M} K_{\sigma'}(x_i(t_L), y_j) + \frac{1}{M^2} \sum_{i,j=1}^{M} K_{\sigma'}(y_i, y_j).$$

To apply the second order method to the modified discrete dynamic system (105), we first rewrite the objective function as

$$J(\{\boldsymbol{\alpha}_k\}_{k=0}^{L-1}; \bar{\mathbf{x}}_0) = \sum_{k=0}^{L-1} g(\mathbf{x}_k, \boldsymbol{\alpha}_k) + b(\sum_{k=1}^{L-1} E(\mathbf{x}_k)) + bE(\mathbf{x}_L) + \lambda \phi(\mathbf{x}_L), \tag{106}$$

where $b \geq 0$ and it balances the kinetic energy and the elasticity energy. Therefore the optimal cost-to-go at stage k is

$$V_k(\mathbf{x}_k) = \min_{\{\boldsymbol{\alpha}_l\}_{l=k}^{L-1}} J(\{\boldsymbol{\alpha}_l\}_{l=k}^{L-1}; \bar{\mathbf{x}}_k).$$

The Bellman's principle of optimality implies

$$V_k(\mathbf{x}_k) = \min_{\boldsymbol{\alpha}_k}\{g(\mathbf{x}_k, \boldsymbol{\alpha}_k) + bE(\mathbf{x}_k) + V_{k+1}(\mathbf{x}_{k+1})\},$$

with finial condition

$$V_L(\mathbf{x}_L) = bE(\mathbf{x}_L) + \lambda\phi(\mathbf{x}_L).$$

8.4 Application

By using the differential dynamic programming, the final state is

$$\mathbf{P}_L = b\nabla_{xx}[E(\bar{\mathbf{x}}_L)] + \lambda\nabla_{xx}[\phi(\bar{\mathbf{x}}_L)], \tag{107}$$

$$\mathbf{q}_L = b\nabla_x[E(\bar{\mathbf{x}}_L)] + \lambda\nabla_x[\phi(\bar{\mathbf{x}}_L)], \tag{108}$$

$$\bar{V}(\bar{\mathbf{x}}_L) = bE(\bar{\mathbf{x}}_L) + \lambda\phi(\bar{\mathbf{x}}_L),$$

$$\Theta_L = 0. \tag{109}$$

The feedback control law is

$$\delta\boldsymbol{\alpha}_k = \mathbf{z}_k - \mathbf{K}_k\delta\mathbf{x}_k, \tag{110}$$

$$\mathbf{K}_k = \mathbf{C}_k^{-1}\mathbf{B}_k, \tag{111}$$

$$\mathbf{z}_k = -\mathbf{C}_k^{-1}\mathbf{d}_k \tag{112}$$

for $k = L - 1, \cdots, 0$, with updated information

$$\mathbf{P}_k = \mathbf{A}_k - \mathbf{B}_k^T\mathbf{C}_k^{-1}\mathbf{B}_k, \tag{113}$$

$$\mathbf{q}_k = \mathbf{e}_k + \mathbf{B}_k^T\mathbf{z}_k, \tag{114}$$

$$\Theta_k = \Theta_{k+1} + \frac{1}{2}\mathbf{d}_k^T\mathbf{z}_k, \tag{115}$$

where

$$\mathbf{A}_k = \mathbf{g}_{xx}^k + b\mathbf{E}_{xx}^k + \mathbf{h}_{xx}^k + \mathbf{f}_x^{kT}\mathbf{P}_{k+1}\mathbf{f}_x, \tag{116}$$

$$\mathbf{C}_k = \mathbf{g}_{\alpha\alpha}^k + \mathbf{h}_{\alpha\alpha}^k + \mathbf{f}_\alpha^{kT}\mathbf{P}_{k+1}\mathbf{f}_\alpha, \tag{117}$$

$$\mathbf{B}_k = \mathbf{g}_{\alpha x}^k + \mathbf{h}_{\alpha x}^k + \mathbf{f}_\alpha^{kT}\mathbf{P}_{k+1}\mathbf{f}_x, \tag{118}$$

$$\mathbf{e}_k = \mathbf{g}_x^k + b\mathbf{E}_x^k + \mathbf{h}_x^k \tag{119}$$

$$\mathbf{d}_k = \mathbf{g}_\alpha^k + \mathbf{h}_\alpha^k. \tag{120}$$

Since the elasticity energy only applies to time stage at $k = 1, 2, \cdots, L-1$,

$$\nabla_x \mathbf{E}_0 = 0, \quad \nabla_{xx} \mathbf{E}_0 = 0. \tag{121}$$

8.5 Numerical Results

We present the numerical results of diffeomorphic matching anterior leaflets including elasticity energy in the objective function. From our initial experiments, we discover that maximum stretching happens mostly on edges. Since there are less neighboring points for boundary points and therefore less control exists over the stretching. Besides due to the nature of measure matching, edges match faster than interior points, so as λ increases, boundary points tend to be dragged along with the points inside. To fix this problem, we split data sets into two parts, interior points and boundary points, and match them separately. In fact, specifically directing movement of shape points allows better matching quality as well as more evenly stretched shapes. This approach is also medically feasible since no matter how human tissue deforms, the edge is confined to stay at the very outside not being able to travel inside.
Now the optimization problem is reformatted as follows

$$
\begin{aligned}
\min \quad & J(\{\boldsymbol{\alpha}_k\}_{k=0}^{L-1}; \bar{\mathbf{x}}_0) = \sum_{k=0}^{L-1} g(\mathbf{x}_k, \boldsymbol{\alpha}_k; t_k) + b \sum_{k=1}^{L} E(\mathbf{x}_k; t_k) \\
& \qquad\qquad + \lambda(a_1 \phi_{int}(\mathbf{x}_L) + a_2 \phi_{bnd}(\mathbf{x}_L)) \\
\text{subject to} \quad & \mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \boldsymbol{\alpha}_k; t_k), \quad k = 0, \cdots, L-1, \\
& \mathbf{x}_0 = \bar{\mathbf{x}}_0,
\end{aligned} \tag{122}
$$

where $\phi_{int}(\mathbf{x}_L)$ and $\phi_{bnd}(\mathbf{x}_L)$ are the positive measures of interior points and boundary points respectively. We use fixed parameters $a_1, a_2 > 0$ to balance the matching quality of interior and edge. Only when both measures decrease at a similar speed, would the stretching be at the same level. We usually choose negative values for b when too much contraction happens and positive values for too much stretching.
The choice of σ for the Gaussian kernel is slightly different from previous experiments. The aim now is to control the stretching but still attain similar matching quality, so we allow the points to deform slightly freely by relaxing the value of σ a little bit.

8.5.1 Diffeomorphic Matching for Anterior Leaflets in Elastic Model

We present the numerical results of diffeomorphic matching for 2 snapshots S_0, S_1 of the anterior leaflet using the elastic model. The reference S_0 consists of 83 points with 36 boundary points and the target set S_1 consists of 82 points with 36 boundary points. The scale parameter σ of the Gaussian kernel is 2.25. Continuation parameter λ for different values of b are all initialized at 300 and increased accordingly with each continuation iteration. The trade-off parameters for measures of interior and boundary are $(a_1, a_2) = (1.1, 1)$. The continuation method stops when both of the measures are less than 5×10^{-3}.
Table 5 shows clearly that with very similar matching quality, elasticity energy decreases from 1.10 to 0.884 as b increases from 0 to 400. Accordingly, the kinetic energy keeps increasing which is the price paid to control the stretching.
In Figure 27, we plot the kinetic energy for fixed λ with different values of b. Even though b varies, the general shape of the curve is very similar. Kinetic energy actually differs after the very first iteration and the difference is carried on for the rest iterations.

b	λ	$\phi_{int}(\mathbf{x}_L)$	$\phi_{bnd}(\mathbf{x}_L)$	Hausdorff distance	Kinetic energy	Elasticity energy
0	1.66×10^4	3.7×10^{-3}	4.4×10^{-3}	3.60	279	1.10
100	1.70×10^4	3.8×10^{-3}	3.0×10^{-3}	2.39	279	1.02
200	1.70×10^4	3.9×10^{-3}	3.1×10^{-3}	2.42	284	0.968
300	1.70×10^4	3.8×10^{-3}	3.8×10^{-3}	2.47	295	0.923
400	1.70×10^4	4.0×10^{-3}	4.7×10^{-3}	2.45	308	0.884

Table 5: Parameters and values for diffeomorphic matching two anterior leaflets with elasticity energy.
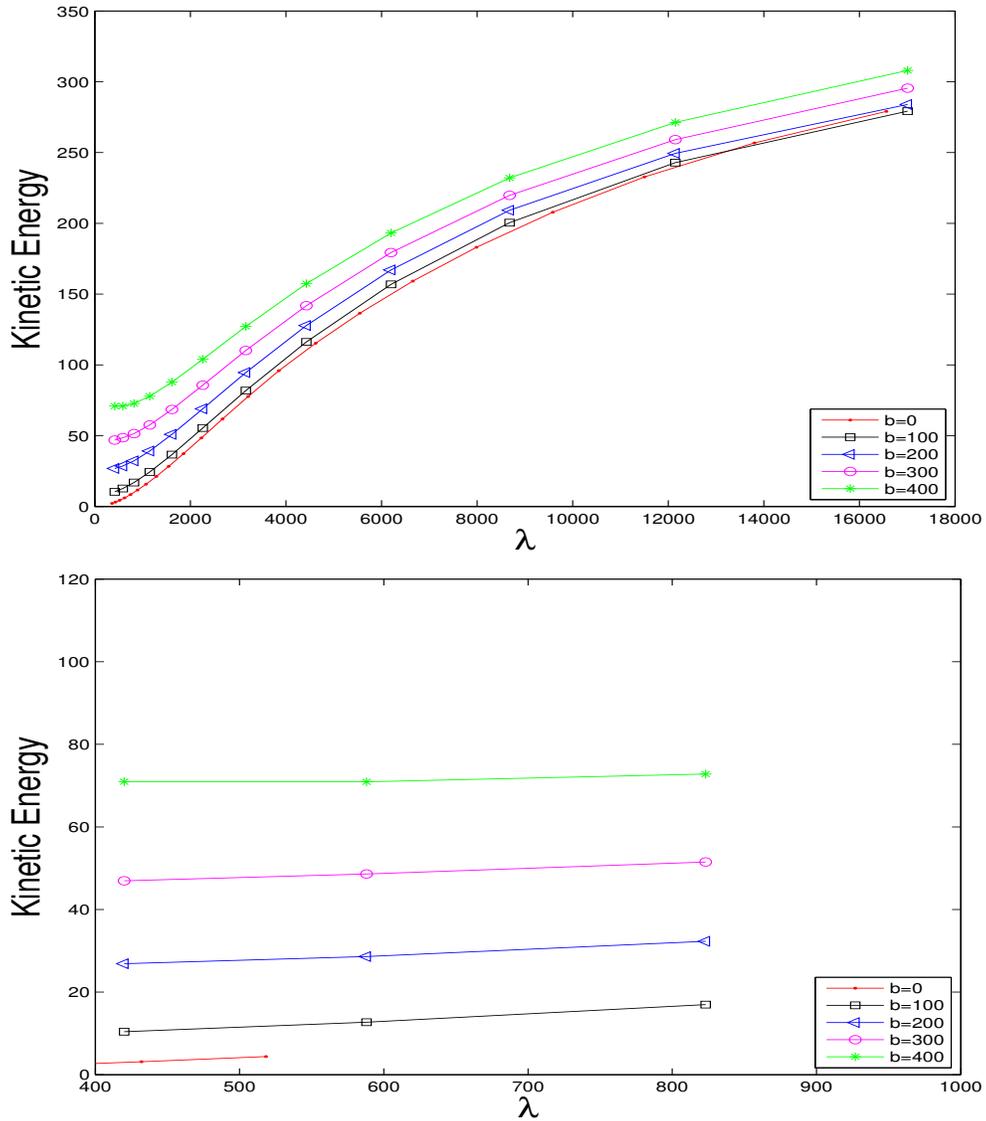

Figure 27: Kinetic energy with respect to regularization parameter λ for diffeomorphic matching two anterior leaflets with elasticity energy: for easier visualization, we plot the first three iterations only in the bottom figure.

b	λ	$\phi_{int}(\mathbf{x}_L)$	$\phi_{bnd}(\mathbf{x}_L)$	Hausdorff distance	Kinetic energy	Elasticity energy
0	1.03×10^4	2.4×10^{-3}	2.9×10^{-3}	3.25	160	1.02
100	1.03×10^4	2.6×10^{-3}	2.8×10^{-3}	3.19	163	0.965
200	1.03×10^4	2.7×10^{-3}	2.9×10^{-3}	3.15	169	0.924
500	1.54×10^4	2.3×10^{-3}	2.1×10^{-3}	2.68	222	0.860

Table 6: Parameters and values for diffeomorphic matching two posterior leaflets with elasticity energy.

Figure 28 displays elasticity energy as a function of penalty parameter λ. We discover similar results as in Figure 27. After the first iteration, the smaller b is, the lower elasticity energy starts.

In Figure 29, for each point of the computed deformation, we calculate the average stretching fraction and then for each percentage plot the corresponding percentile. In the bottom figure, we observe the distortion is less intense for greater value of b. Especially when $b = 400$, the overall stretching fraction is below 1.25 which satisfies the maximum stretching the mitral valve tissue tolerates.

For computed deformations with $b = 0, 100, 200, 300, 400$, Figure 30 displays several level curves of the average distortion fraction. The coordinate system has been modified isometrically at each snapshot instant in order to display a better projection.

8.5.2 Diffeomorphic Matching for Posterior Leaflets with Contraction

We apply the same algorithm to diffeomorphic matching of posterior leaflet snapshots S_0, S_1 as in section 8.5.1. The reference S_0 consists of 260 points with 65 boundary points and the target set S_1 consists of 282 points with 80 boundary points. The scale parameter σ of the Gaussian kernel is 3.34. Continuation parameter λ for different values of b are all initialized at 400 and increased dynamically with each outer iteration. The continuation iteration stops when both measures are less than 3×10^{-3} and $(a_1, a_2) = (1, 1.8)$.

According to Table 6, similar behaviors are discovered as in the section 8.5.1. With very similar matching quality, elasticity energy decreases from 1.02 to 0.860 as b increases from 0 to 500. Responding to value changes of b, the kinetic energy keeps increasing.

In Figure 31, the top graph shows that percentile curves get lower with greater values of b which is similar to Figure 29. The right bottom figure indicates that the highest stretching is very acceptable once b kicks in. However, we notice that the contraction is extraordinarily high with greater values of b. In particular, at $b = 500$ the least stretching fraction is much less than 0.75 which is infeasible to the nature of human tissues. Instead of controlling the high stretching, the task now is to reduce the excessive contraction. To fulfill that, we place extra weighting parameters w to the elasticity energy for 5 percent points with least stretching.

Consider the optimal control problem

$$
\begin{aligned}
\min \quad & J(\{\boldsymbol{\alpha}_k\}_{k=0}^{L-1}; \bar{\mathbf{x}}_0) = \sum_{k=0}^{L-1} g(\mathbf{x}_k, \boldsymbol{\alpha}_k; t_k) + b \sum_{k=1}^{L} \left(wE^1(\mathbf{x}_k; t_k) + E^2(\mathbf{x}_k; t_k) \right) \\
& \qquad + \lambda(a_1 \phi_{int}(\mathbf{x}_L) + a_2 \phi_{bnd}(\mathbf{x}_L)) \\
\text{subject to} \quad & \mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \boldsymbol{\alpha}_k; t_k), \quad k = 0, \cdots, L-1, \\
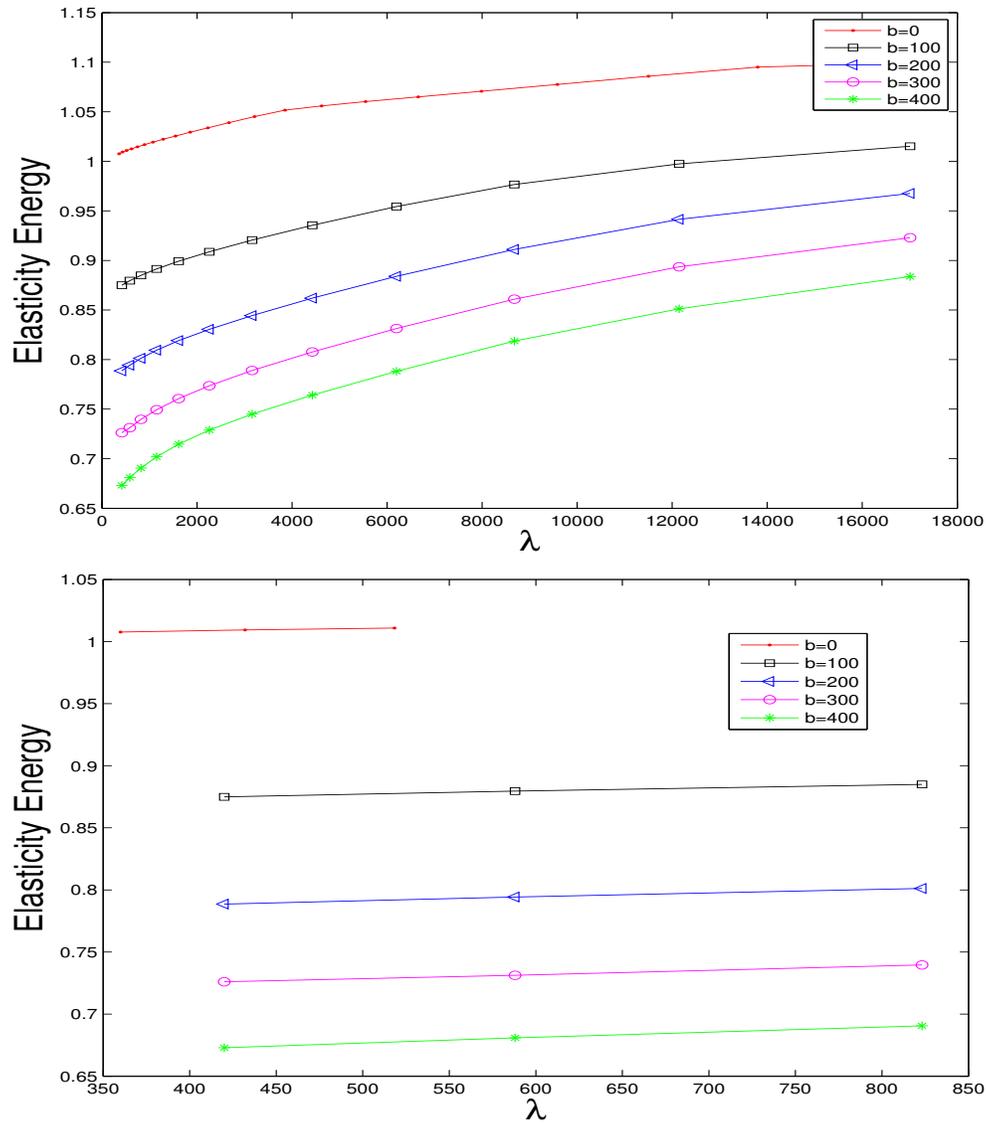& \mathbf{x}_0 = \bar{\mathbf{x}}_0,
\end{aligned}
\tag{123}
$$

Figure 28: Elasticity energy with respect to regularization parameter λ for diffeomorphic matching two anterior leaflets with elasticity energy: for easier visualization, we plot the first three iterations only in the bottom figure.
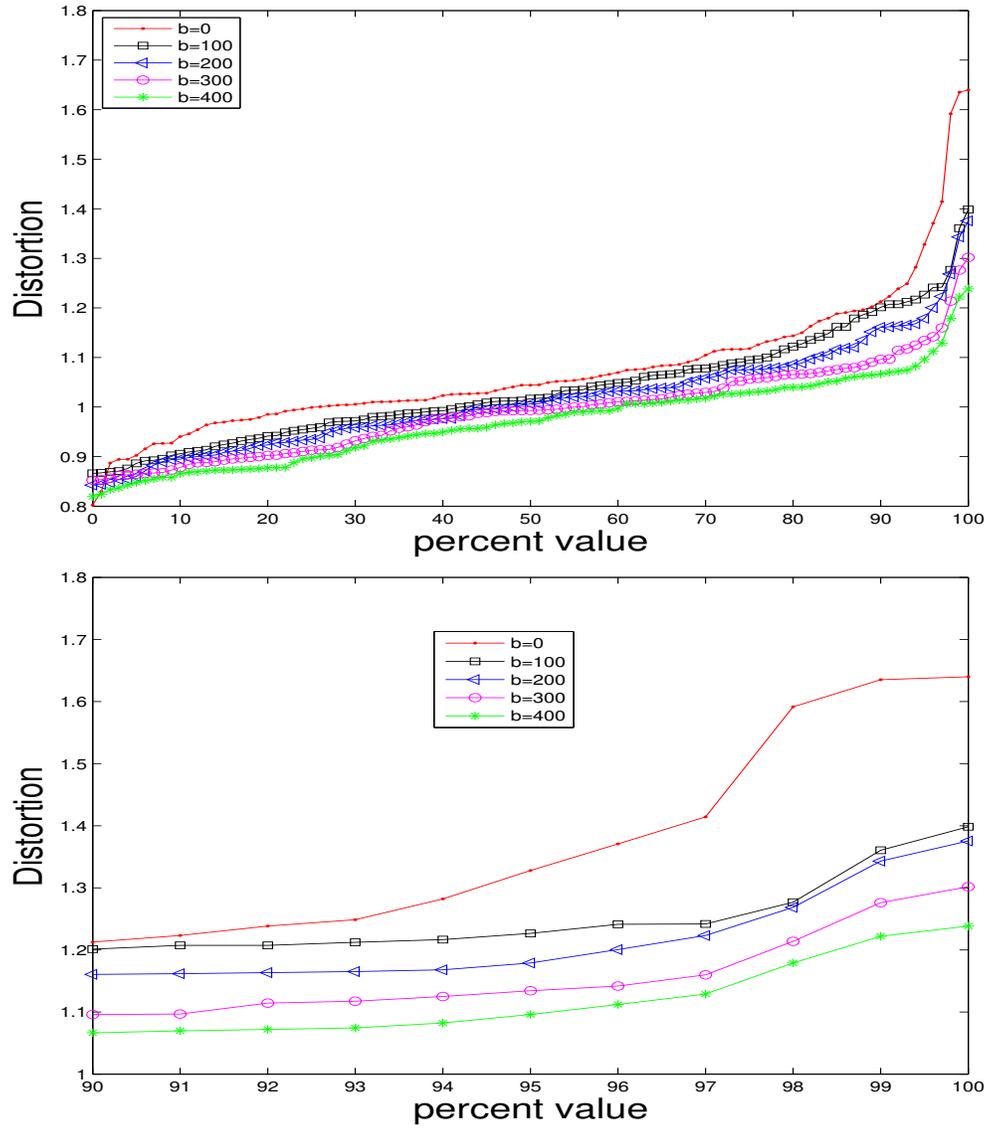
Figure 29: Percentile curves of distortion for diffeomorphic matching two anterior leaflets with elasticity energy: for easier visualization, we plot percentile 90 to 100 separately.
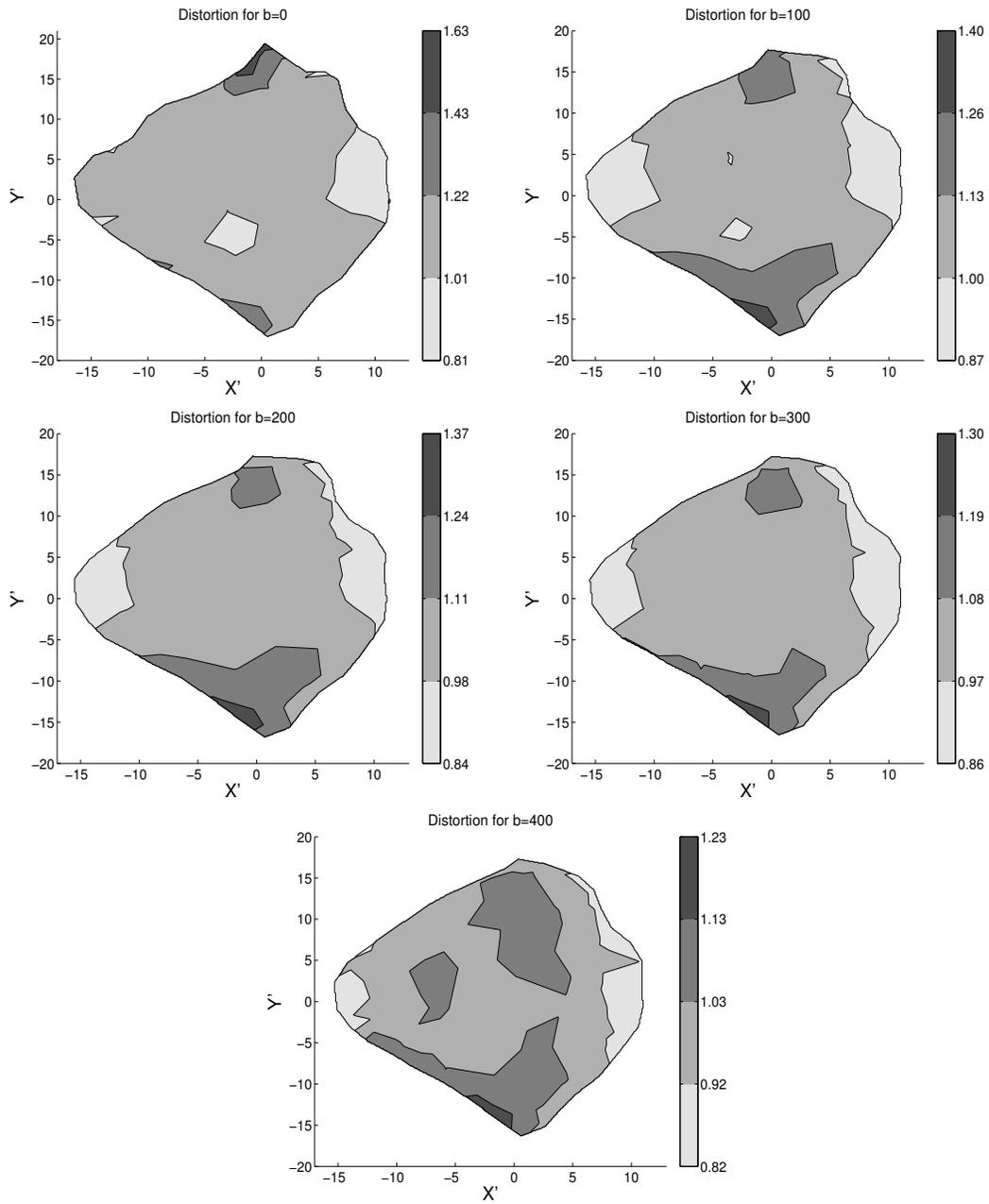
Figure 30: Distortion for computed anterior leaflets for $b = 0,\ 100,\ 200,\ 300,\ 400$.
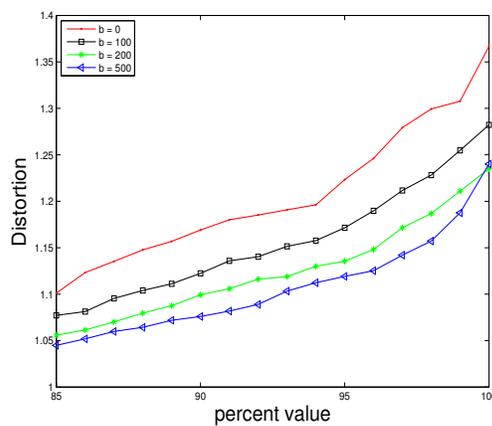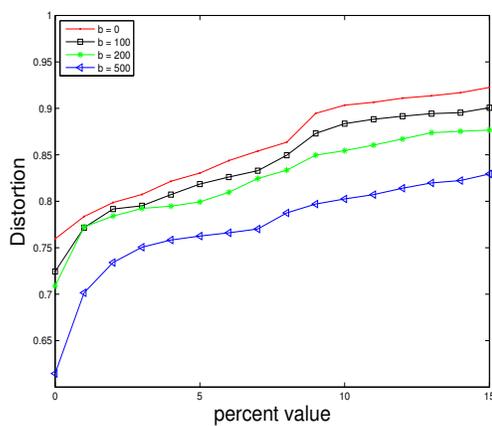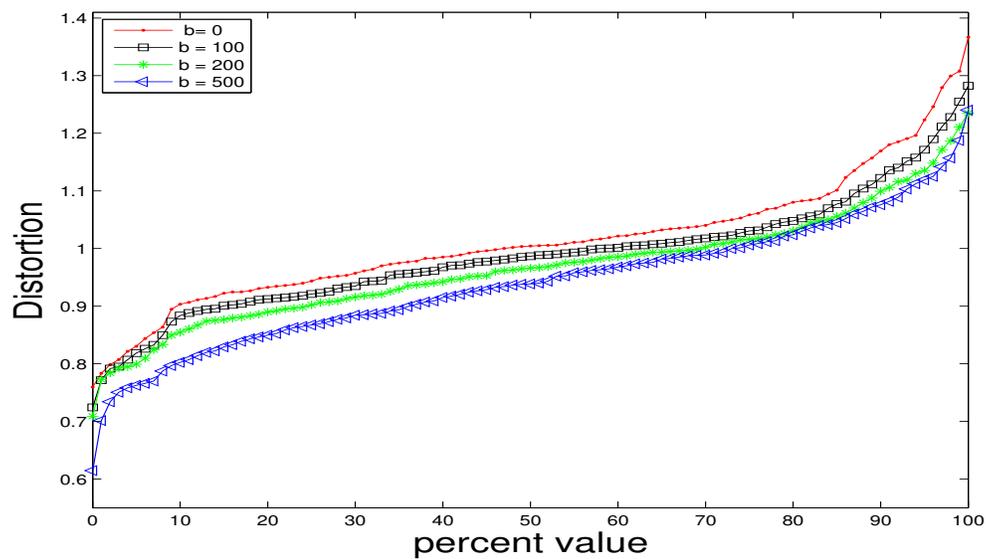
Figure 31: Percentile curves of distortion for diffeomorphic matching two posterior leaflets with elasticity energy: for easier visualization, we plot percentile 0 to 15 and 85 to 100 separately in the bottom row.

66

w	λ	$\phi_{int}(\mathbf{x}_L)$	$\phi_{bnd}(\mathbf{x}_L)$	Hausdorff distance	Kinetic energy	Elasticity energy
1	1.03×10^4	2.7×10^{-3}	2.9×10^{-3}	3.15	169	0.924
0	1.03×10^4	2.7×10^{-3}	2.9×10^{-3}	3.15	169	0.925
-0.5	1.37×10^4	2.7×10^{-3}	2.9×10^{-3}	3.15	169	0.927
-1	1.03×10^4	2.7×10^{-3}	2.9×10^{-3}	3.15	170	0.928

Table 7: Parameters and values for diffeomorphic matching with contraction control.

where E^1 is the elasticity energy of the first 5 percent of points with least stretching and E^2 is that of the remaining points at time t_k. To increase the stretching fraction of those points, we choose $w < 1$ and for better performance negative values are much preferable.

We now fix $b = 200$ and adjust w with the rest parameters staying the same.

As w decreases, the overall elasticity energy increases but the difference is very minor. Table 7 also shows a very slight increment in kinetic energy.

Figure 32 presents the percentiles curves for $w = 1$, $w = 0$, $w = -0.5$, $w = -1$ when b is fixed at 200. Since we only modify the weights of elasticity energy for the most contracted points, it does not affect the rest regions too much as suggested in the bottom right figure. In the bottom left figure, it is clear that the least contraction is above 0.75 and coincides with behaviors of human tissue.

For computed deformations with $b = 200$, Figure 33 displays several level curves of the average distortion fraction for $w = 1$, $w = 0$, $w = -0.5$, $w = -1$. The coordinate system has been modified isometrically at each snapshot instant in order to display a better projection.

9 Conclusions and Future Work

In this paper, we have focused on finding optimal solutions to diffeomorphic matching surfaces in 3D medical imaging. The main goal is to find a time-dependent diffeomorphism F_t and reconstruct the trajectories of leaflets. The variational approach we use has been actively explored in [10, 27, 33, 34]. This paper can be recognized as a continuation work of [6, 8]. However, the key difference here is that we have developed a fast computing method using the second order information. By our numerical experiments in section 6, the beauty of Newton steps is revealed. It only needs at most six iterations to achieve the stopping criterion of high accuracy. Additionally, iterations needed for convergence do not depend on data sizes.

In practical applications, to precisely capture the local properties of objects, a shape with high resolution is much preferable. The multi-scale method we develop actually help overcome the computational challenges dealing with finer mesh points. Not only the computing time has been reduced, the matching quality is not compromised.

Finally, by adding the elasticity energy and splitting data into interior and boundaries, overall stretching is well controlled and satisfies the medical standards. To cope with unexpected contraction, we use negative weighting parameters to modify local intense contraction and the results are quite satisfying.

In this paper, we only regenerate the deformations of anterior and posterior leaflets with given snapshots separately. However, the mitral valve apparatus comprises two leaflets (anterior and posterior) and a fibrous ring called annulus. Our future plan is to deform all parts of the mitral valve as one object and hence to reconstruct the deformation of the mitral valve apparatus between given snapshots.

We have one example in section 3 dealing with three snapshots. However, we actually split the [0,1] interval into two and apply the second order method to each time interval individually. So we further
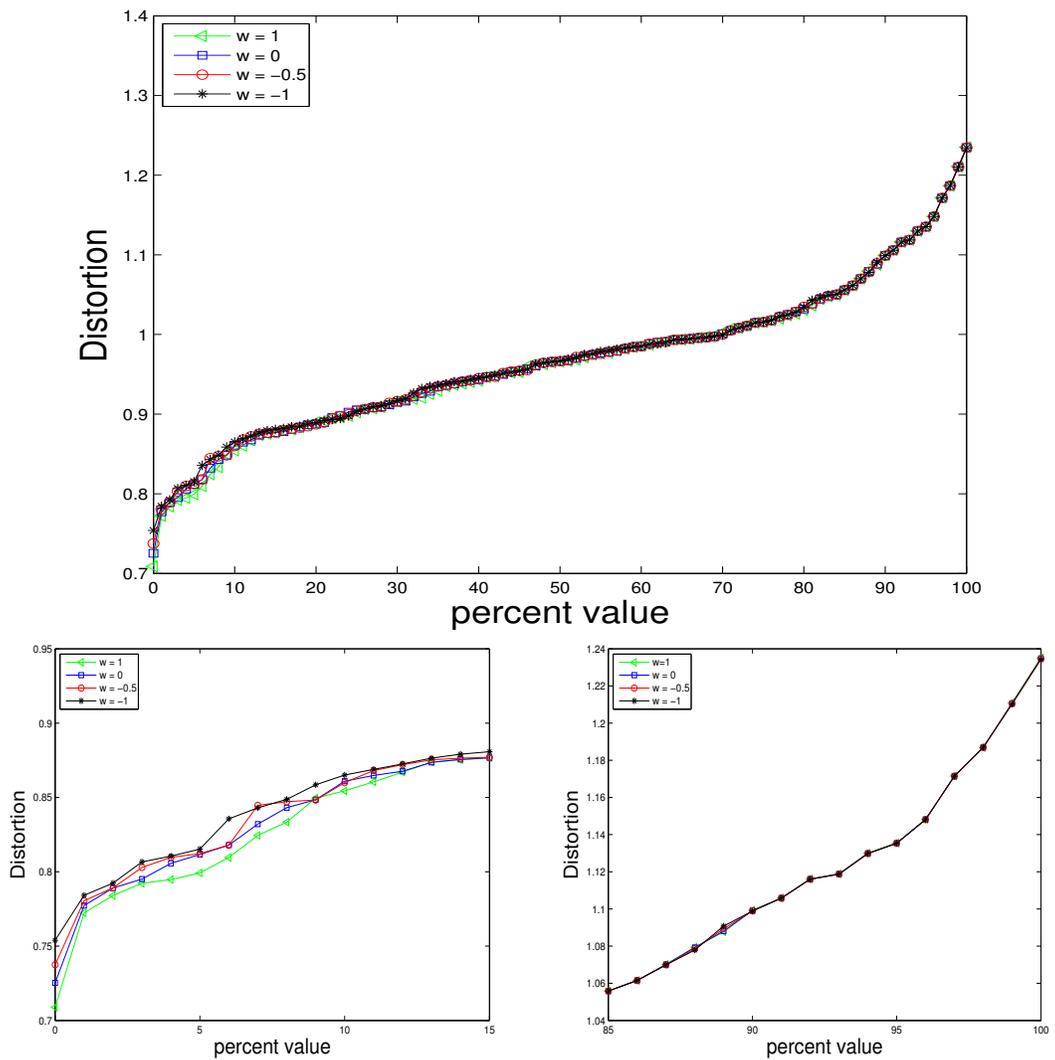
Figure 32: Percentile curves of distortion with contraction control for various w values: for easier visualization, we present percentile 0 to 15 and 85 to 100 separately in the bottom row.
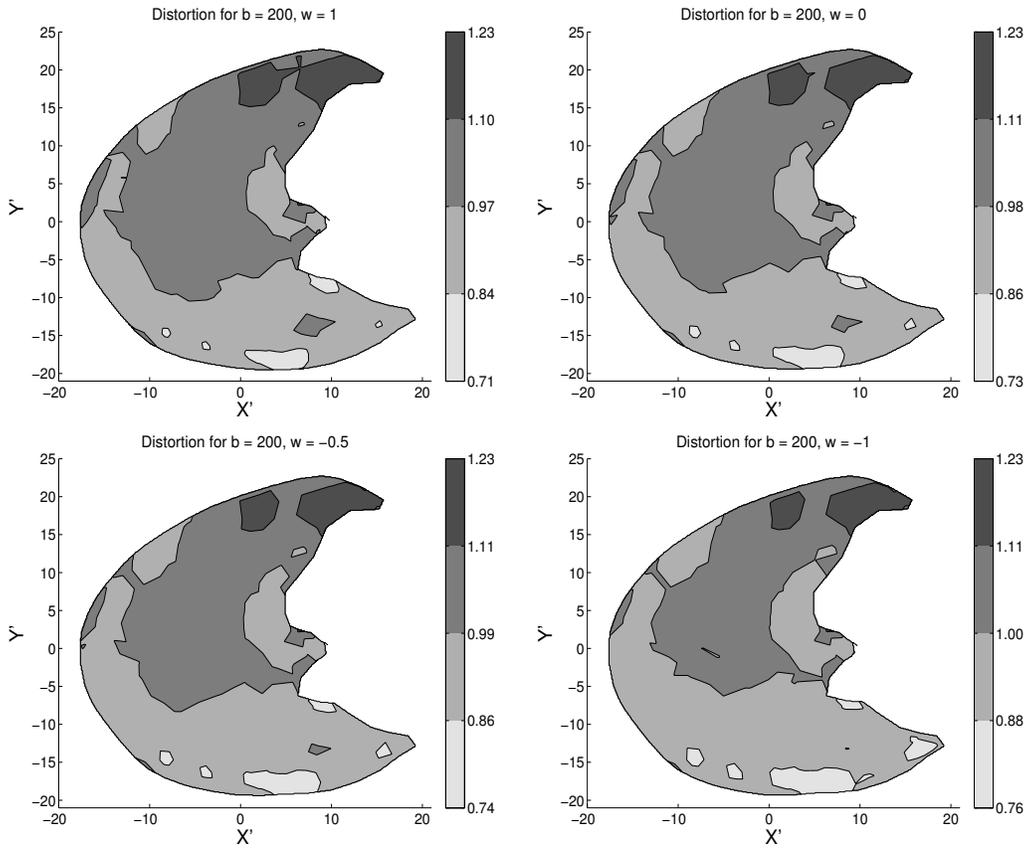
Figure 33: Distortion for computed posterior leaflets at $b = 200$ for $w = 1$, $w = 0$, $w = -0.5$, $w = -1$.

plan to find optimal matching for multiple sub-manifolds as in [6] and recover the full motion.

In the paper, we use the pure Newton method where the step size is fixed at 1. Sometimes it could be a little ambitious and hitting a saddle point could jeopardize the convergence, so we plan to define the step size by backtracking line search.

Last but not the least, redefine the elasticity energy term to consider biological prior knowledge. Because of tissue's low level of viscoelasticity, modeling the valve tissue as an elastic material is currently an accepted practice. So one remedy is to investigate the linear elasticity model and formulate the strain tensor.

References

[1] F. Abraham, M. Behr, and M. Heinkenschloss, *The effect of stabilization in finite element methods for the optimal boundary control of the Oseen equations,* Finite Elments in Analysis and Design, 41(2004), pp. 229-251.

[2] V.I. Arnold, *Ordinary Differential Equations,* MIT Press, Cambridge, Massachusetts, 1973.

[3] V.I. Arnold, *Sur la geometrie differentielle des groupes de Lie de demesion infinte et ses applications l'hydrodynamique des fluids parfaits,* Ann. Inst. Fourier(Grenoble), 1(1996), pp. 319-361.

[4] N. Aronszajn, *Theory of reproducing kernels,* Transactions of the American Mathematical Society, 68(1950), pp. 337-404.

[5] F. Arrate, J.T. Ratnanather, and L. Younes, *Diffeomorphic active contours,* SIAM J. Imaging Science, 3(2010), pp. 176-198.

[6] R. Azencott, R. Glowinski, J. He, R.H.W. Hoppe, A. Jajoo, Y. Li, A. Martynenko, S. Benzekry, S.H. Little, and W.A. Zoghbi, *Diffeomorphic matching and dynamic deformable surfaces in 3D medical imaging,* Comput. Meth. in Appl. Math., 2010, pp. 235-274.

[7] R. Azencott, R. Glowinski, and A.M. Ramos, *A controllability approach to shape identification,* Applied Math. Letters, 21(2008), pp. 861-865.

[8] R. Azencott, A. Jajoo, S. Jain, J. He, Y. Li, A. Martynenko, S. Benzekry, G. Lawrie, S.H. Little, and W.A. Zogbhi, *Comparitive evaluation of mitral valve strain by deformation tracking in 3D-echocardiography*, Cardiovascular Engineering and Technology, 3(2012), pp. 402-412.

[9] M. Bardi and I. Capuzzo-Dolcetta, *Optimal Control and Viscosity Solutions of Hamilton-Jacobi-Bellman Equations*, Birkhauser, 1997.

[10] M. F. Beg, M.I. Miller, A. Trouve and L. Younes, *Computing large deformations metric mappings via geodesic flows of diffeomorphisms,* Int. J. Comp. Vis., 61(2005), pp. 139-157.

[11] R.E. Bellman, *Dynamic Programming*, Princeton University Press, 1957.

[12] R.E. Bellman, *Eye of the Hurricane: An Autobiography*, World Scientific, Singapore, 1984.

[13] D.P. Bertsekas, *Dynamic Programming*, Prentice-Hall Inc., 1987.

[14] D.P. Bertsekas, *Dynamic Programming and Optimal Control*, Athena Scientific, 2005.

[15] J. Betts, *Practical Methods for Optimal Control Using Nonlinear Programming,* SIAM, Philadelphia, USA, 2001.

[16] J.T. Betts and S.L. Campbell, *Mathematics for Industry: Challenges and Frontiers,* SIAM, 2005.

[17] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.

[18] L.G. Brown, *A survey on image registration techniques,* ACM Computing Surveys, 24(1992), pp. 326-376.

[19] C.J.C. Burges, *A tutorial on support vector machines for pattern recognition,* Data Mining and Knowledge Discovery, 2(1998), pp. 121-167.

[20] V. Camion and L. Younes, *Geodesic interpolating splines,* in EMMCVPR 2001, M. Figueiredo, J. Zerubia, and K. Jain, A. eds., vol 2134 of Lecture Notes in Computer Science, Springer 2001, pp. 513-527.

[21] Y. Cao, M.I. Miller, R. Winslow, and L. Younes, *Large deformation metric mapping of vector fields,* IEEE Trans. Med. Imaging, 24(2005), pp. 1216-1230.

[22] A. Chinchuluun, P.M. Pardalos, A. Migdalas, and L. Pitsoulis, *Pareto Optimality, Game Theory and Equilibria,* Springer, Berlin-Heidelber-New York, 2008.

[23] G. Christensen, *Deformable shape models for anatomy,* PhD Thesis, Dept. of Electrical Engineering, Sever Institute of Technology, Washington Univ., St. Louis, MO, 1994.

[24] P.M. De Zeeuw, *Matrix-dependent prolongations and restrictions in a blackbox multigrid solver,* J. Comput. Appl. Math. 33(1990), pp. 1-27.

[25] P. Deufhard, *Newton Methods for Nonlinear Problems: Affine Invariance and Adaptive Algorithms*, Springer, Berlin-Heidelberg-New York, 2004.

[26] J. Dieudonné, *Foundations of Modern Analysis*, Academic Press, 1960.

[27] P. Dupuis, U. Grenander and M. Miller, *Variational problems on flows of diffeomorphisms for image matching*, Quarterly of Applied Math., 56(1998), pp. 587-600.

[28] S.R. Eddy, *What is dynamic programming?*, Nature Biotechnology, 22(2004), pp. 909-910.

[29] L.C. Evans, *An Introduction to Mathematical Optiaml Control Theory Version 0.2*, http://math.berkeley.edu/∼evans/control.course.pdf, Univ. of Carlifornia, Berkeley, 2008.

[30] K. Ghobadi, N.S. Nedialkov and T. Terlaky, *Discretize and optimize approaches for PDE-based optimization problems,* http://individual.utoronto.ca/kimia/KimiaGhobadi_CompOptPaper.pdf, University of Toronto, 2012.

[31] C.A. Glasbey, *Two-dimensional generalizations of dynamic programming for image analysis*, Statistics and Computing, (19)2009, pp. 49-56.

[32] J. Glaunès, A. Qiu, M.I. Miller, and L. Younes, *Large deformation diffeomorphic metric curve mapping,* Int. J. Comp. Vision, 80(2008), pp. 317-336.

[33] J. Glaunès, A. Trouvé and L. Younes, *Diffeomorphic matching of distributions: A new approach for unlabelled point-sets and sub-manifolds matching*, CVPR, IEEE Computer Society, 2004, pp. 712-718.

[34] J. Glaunès, M. Vaillant and I. Miller, *Landmark matching via large deformations diffeomorphisms on the sphere,* J. of Math. Imaging and Vision, 20(2004), pp. 179-200.

[35] R. Glowinski and J.L. Lions, *Exact and Approximate Controllability for Distributed Parameter Systems,* Cambridge University Press, Part I: Acta Numerica, 1994, pp.269-378, Part II: Acta Numerica, 1995, pp. 159-333.

[36] U. Grenander and M.I. Miller, *Computational anatomy: an emerging discipline*, Quart. App. Math., 56(1998), pp. 617-694.

[37] H. Guo, A. Rangarajan and S. Joshi, *Diffeomorphic point matching,* Handbook of Mathematical Models in Computer Vision, Springer, Berlin-Heidelberg-New York, 2006, pp. 205-219.

[38] R.M. Haralick and L.G. Shapiro, *Image segmentation techniques*, Comput. Vis. Graph. Image Proc., 29(1985), pp. 100-132.

[39] P.K. Jain and K. Ahmad, *Functional Analysis*, New Age International, 1995.

[40] A. Jajoo, *Diffeomorphic matching and dynamic deformable shapes*, PhD Thesis, University of Houston, 2011.

[41] S. Joshi, E. Klassen, A. Srivastava and I. Jermyn, *A novel representation for Riemannian analysis of elastic curves in r^n,* In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2007, pp. 1-7.

[42] S. Joshi and M. Miller, *Landmark matching via large deformation diffeomorphisms,* IEEE Transactions in Image Processing, 9(2000), pp. 1357-1370.

[43] S. Kameswaran and L.T. Biegler, *Advantages of nonlinear-programming-based methodologies for inequality path-constrained optimal control problems - a numerical study,* SIAM Journal on Scientific Computing, 30(2008), pp. 957-981.

[44] M. Kass, A. Witkin, and D. Terzopoulos, *Sankes: active contour models,* Int. Journal of Computer Visions, 1(1987), pp. 321-331.

[45] D.G. Kendall, *Shape manifolds, procursteam metrics and complex projective spaces*, Bull. London Math. Soc., 16(1984), pp. 81-121.

[46] D.G. Kendall, D. Barden, T.K. Crane, and H. Le, *Shape and Shape Theory*, Wiley, 1999.

[47] N. Khaneja, M.I. Miller, and U. Grenander, *Dynamic programming generation of curves on brain surfaces*, Pattern Analysis and Machine Intelligence, 11(1998), pp. 1260-1265.

[48] S. Kichenassamy, A. Kumar, P. Olver, A. Tannenbaum, and A. Yezzi, *Gradient flows and geometric active contours,* Proc. ICCV, June 1995, pp. 810-815.

[49] K. Kim, M.O. Franz, and B. Scholkopf, *Iterative kernel principal component analysis for image modeling*, IEEE Trans. Pattern Anal. Mach. Intell., 27(2005), pp. 1351-1366.

[50] E. Klassen, A. Srivastava, W. Mio and S.H. Joshi, *Analysis of planar shapes using geodesic paths on shape spaces*, IEEE Trans. Pattern Anal. Mach. Intell., 26(2004), pp. 372-383.

[51] R. Klette, K. Schlüns, and A. Koschan, *Computer Vision: Three-dimensional Data from Images,* Berlin: Springer, 1998.

[52] A. Mennucci and A. Yezzi, *Metrics in the space of curves,* tech. rep., arXiv:math. DG/0412454 v2, 2005.

[53] J. Mercer, *Functions of positve and negative type and their connections with the theory of integral equations*, Philos. Trans. Roy. Soc. London, 209(1909), pp. 415-446.

[54] M.I. Miller, A. Trouve, and L.Younes, *Computing large deformations via geodesic flows of diffeomorphisms,* Int. J. of Comput. Vision, 61(2005), pp. 139-157.

[55] M.I. Miller, A. Trouvé, and L. Younes, *On the metrics and Euler-Lagrange equations of computational anatomy,* Ann. Review of Biomedical Engineering, 4(2002), pp. 375-405.

[56] M.I. Miller and L. Younes, *Group action, diffeomorphism and matching: A general framework,* Int. J. Comp. Vis., 41(2001), pp. 61-84.

[57] E.H. Moore, *On properly positive Hermitian matrices,* Bull. Amer. Math. Soc., 23(1916), pp. 59.

[58] T.S. Newman and A.K. Jain, *A survey of automated visual inspection*, Comput Vision Image Underst, 61(1995), pp. 231-262.

[59] E. Parzen, *Time Series Analysis Papers*, Holden-Day, San Francisco, CA, 1967.

[60] V.I. Paulsen, *An introduction to the theory of reproducing kernel Hilbert space*, http://www.math.uh.edu/∼vern/rkhs.pdf, Univ. of Houston, 2006.

[61] F. Pernkopfa and P. OLeary, *Image acquisition techniques for automatic visual inspection of metallic surfaces*, NDT&E International, 36(2003), pp. 609-617.

[62] D.L. Pham, C. Xu, and J.L. Prince, *Current methods in medical image segmentation,* Annual Review of Biomedical Engineering, 2(2000), pp. 315-337.

[63] M.H. Quang, S.H. Kang, and T.M. Le, *Image and Video Colorization Using Vector-Valued Reproducing Kernel Hilbert Spaces,* Springer Science+Business Media, 2010.

[64] G.M. Quénot, *Image matching using dynamic programming: application to stereovision an image interpolation*, Image Communication, Bordeaux, France, 1996, pp. 2657-2660.

[65] R.T. Rockafellar and R.J.B. Wets, *Variational Analysis,* Springer-Verlag, 2005.

[66] I.J. Schoenberg, *Metric spaces and positive definite functions*, Trans. Am. Math. Soc. 44(1938), pp. 522-536.

[67] B. Schökopf and A. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*, MIT Press, Cambridge, 2002.

[68] E. Sharon, A. Brandt and R. Basri, *Fast multiscale segmentation,* IEEE Computer Vision and Pattern Recognition, 1(2000), pp. 70-77.

[69] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*, Cambridge University Press, Cambridge, 2004.

[70] H. Takeda, S. Farsiu, and P. Milanfar, *Kernel regression for image processing and reconstruction*, IEEE Tran. Im. Proc., 16(2007), pp. 349-366.

[71] D. Terzopoulos, *Image analysis using multigrid relaxation methods,* IEEE Transactions on Pattern Analysis and Machine Intelligence, 8(1986), pp. 129-139.

[72] A. Trouvé, *An infinite dimensional group approach for physics based models in patterns recognition*, Technical Report, John Hopkins University, 1995.

[73] A. Trouvé, *Diffeomorphisms groups and pattern matching in image analysis,* Int. J. of Comput. Vision, 28(1998), pp. 213-221.

[74] C. Twinnings, S. Marsland, and C. Taylor, *Measuring geodesic distances on the space of bounded diffeomorphisms,* in British Machine Vision Conference, 2002, pp. 847-856.

[75] K. Yao, *Applications of reproducing kernel Hilbert spaces-bandlimited signal models*, Information and Control, 11(1967), pp. 429-444.

[76] L. Younes, *Computable elastic distances between shapes,* SIAM J. Appl. Math, 58(1998), pp.565-586.

[77] L. Younes, *Optimal matching between shapes via elastic deformations,* Image and Vision Computing, 17(1999), pp. 381-389.

[78] L. Younes, P. Michor, J. Shan, and D. Mumford, *A metric on shape spaces with explicit geodesics,* Rend. Lincei Mat. Appl., 9(2008), pp. 25-57.

[79] P.A. van de Elsen, E.D. Pol, and M.A. Viergever, *Medical image matching - a review with classification*, IEEE Engineeing in medicine and biology, 12(1993), pp. 26-39.

[80] B. Zitovà and J. Flusser, *Image registration methods: a survey,* Image and Vision Computing, 21(2003), pp. 977-1000.