

NUMERICAL COMPUTATION OF WEIL-PETERSON GEODESICS IN THE UNIVERSAL TEICHMÜLLER SPACE

MATT FEISZLI AND AKIL NARAYAN

ABSTRACT. We propose an optimization algorithm for computing geodesics on the universal Teichmüller space $T(1)$ in the Weil-Petersson (WP) metric. Another realization for $T(1)$ is the space of planar shapes, modulo translation and scale, and thus our algorithm addresses a fundamental problem in computer vision: compute the distance between two given shapes. The identification of smooth shapes with elements on $T(1)$ allows us to represent a shape as a diffeomorphism on S^1 . Then given two diffeomorphisms on S^1 (i.e., two shapes we want connect with a flow), we formulate a discretized WP energy and the resulting problem is a boundary-value minimization problem. We numerically solve this problem, providing several examples of geodesic flow on the space of shapes, and verifying mathematical properties of $T(1)$. Our algorithm is more general than the application here in the sense that it can be used to compute geodesics on any other Riemannian manifold.

1. INTRODUCTION AND BACKGROUND

Representation and comparison of shapes is a central problem in computer vision. In the past several decades, many approaches to represent, compare, and classify shapes have been presented (See [21], [14] for review and discussion). The space of 2D shapes is inherently nonlinear; this poses fundamental difficulties in computer vision when attempting object recognition and statistics. In [17], Mumford and Sharon describe a construction based on conformal mapping which makes the space of simple closed plane curves into a Riemannian metric space. The space itself is in fact the universal Teichmüller space with the Weil-Petersson metric; in this paper, we describe a numerical solver for geodesics in this space. In Sections 1.1 and 1.2 we describe the optimization problem we wish to solve: minimization of an energy functional with boundary value constraints. We note that a minimization algorithm proposed in [17] was applied to only relatively simple shapes because of numerical difficulties. In this work we aim to apply our algorithm to more general, complicated shapes. Our work is competitive with a recent approach based on shooting for the analogous boundary value problem [9].

The outline of this paper is as follows: Section 2 describes the mathematics of computations for the Weil-Petersson metric using discrete samples of a velocity field. Section 3 discusses the fully discrete algorithm along with our technique for satisfying the boundary constraints. Finally, Section 4 presents numerical results that illustrate the effectiveness of the algorithm.

1.1. Conformal Welding. The construction in [17] is classical in Teichmüller theory. Given a simply-connected planar region Ω bounded by a smooth Jordan curve (this is our definition of a “shape”), one constructs a pair of conformal maps $\Phi_+ : \Delta \rightarrow \Omega_+$ and $\Phi_- : \Delta \rightarrow \Omega_-$, which map the exterior and interior of the unit disk to the exterior and interior of Ω , respectively. The exterior map is normalized to fix ∞ and have real derivative there. The interior map is only defined up to right multiplication by the three-parameter

A. Narayan was supported by National Science Foundation Awards DMS-1318427 and DMS-1552238.

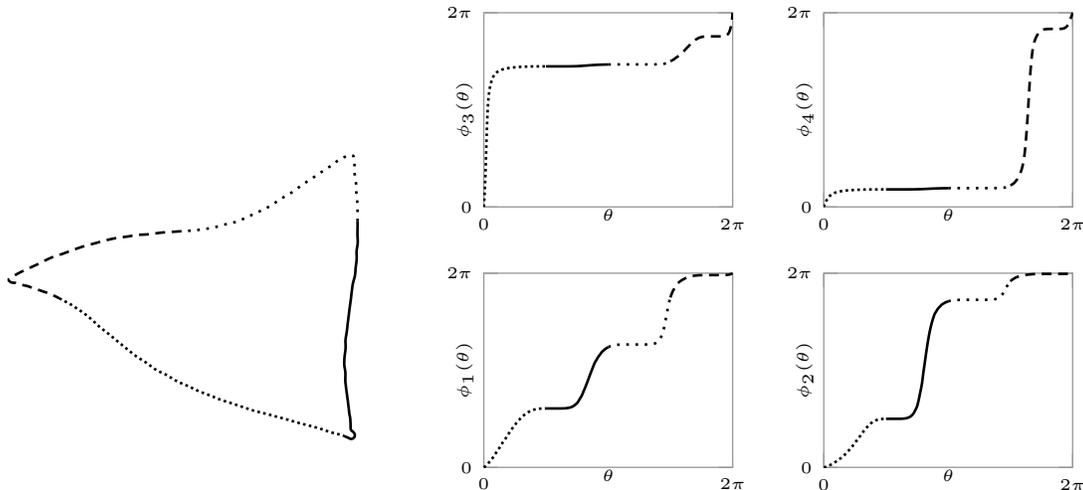


Figure 1. A triangular region Ω (left) transformed into an equivalence class of welding maps ϕ (right). Four members of the equivalence class are shown. Four segments of $\partial\Omega$ are drawn with differing linestyles to highlight their corresponding segment on each of the welding maps.

Möbius group $PSL_2(\mathbb{R})$ of conformal self-maps of the unit disk. Both Φ_+, Φ_- extend continuously to the boundary S^1 , and their composition $\Phi_-^{-1} \circ \Phi_+$, restricted to the boundary, is a map $\Psi : S^1 \rightarrow S^1$.

A remarkable result (see [2], for example) is that this result is almost an isomorphism: the space of shapes, modulo translation and scale, is isomorphic to the group $\mathbf{Diff}(S^1)$, modulo conformal self-maps of the disk. This provides an elegant way of making the space of shapes into a metric space: we take an element of the coset space $PSL_2(\mathbb{R}) \backslash \mathbf{Diff}(S^1)$, i.e. an equivalence class of diffeomorphisms of the circle, as a representation of a shape. This space is known as the universal Teichmüller space and was initially studied in the context of Riemann surfaces [2, 8] it also arises in string theory [4, 15]. In [17] elements of $PSL_2(\mathbb{R}) \backslash \mathbf{Diff}(S^1)$ are called “fingerprints”; in the mathematical literature, elements of the broader class of quasymmetric homeomorphisms of S^1 are commonly known as “welding maps” and each has an associated quasicircle. We illustrate an example of an equivalence class of welding maps in Figure 1 for a simple shape.

1.2. The Weil-Petersson Riemannian metric. The Lie algebra to $PSL_2(\mathbb{R}) \backslash \mathbf{Diff}(S^1)$ is the space of vector fields on the circle modulo the subspace spanned by 1, \cos and \sin . If $v(\theta) = \sum_{n=-\infty}^{\infty} a_n e^{in\theta} \frac{\partial}{\partial \theta}$ is a real-valued vector field on the circle, the Weil-Petersson norm is

$$(1) \quad \|v\|_{WP}^2 = \sum_{n=2}^{\infty} (n^3 - n) |a_n|^2 = \frac{1}{2} \left(\|v^{(3/2)}\|^2 - \|v^{(1/2)}\|^2 \right)$$

where $\|\cdot\|$ denotes the L^2 norm.¹ This may be rewritten as

$$\|v\|_{WP}^2 = \langle Lv, v \rangle$$

¹This formula, while explicit, is troublesome in numerical computations; the boundary values of conformal maps have so much high-frequency content, even for curves with real-analytic boundaries, that the number of Fourier coefficients required for an accurate global representation is prohibitive.

for the positive semidefinite, self-adjoint operator

$$(2) \quad L = -\mathcal{H}(\partial^3 - \partial)$$

where \mathcal{H} is the Hilbert transform. Note that L has a kernel which is exactly the span of $1, \cos$ and \sin . As mentioned above, the corresponding vector fields in the span of $\{\sin \theta \frac{\partial}{\partial \theta}, \cos \theta \frac{\partial}{\partial \theta}, \frac{\partial}{\partial \theta}\}$ are infinitesimal Möbius maps; i.e. they span the Lie algebra $psl_2(\mathbb{R})$ of the Möbius group $PSL_2(\mathbb{R})$. Hence, our norm is indeed a norm on the quotient space.

By right-translations we extend this norm to the entire space; that is, if $\phi(\theta, t) \equiv \phi_t(\theta)$ is a curve in $PSL_2(\mathbb{R}) \setminus \mathbf{Diff}(S^1)$, then any any time t we pull back the velocity $\dot{\phi}_t$ under the derivative of right translation to obtain

$$v_t(\theta) = \dot{\phi}_t \circ \phi_t^{-1}$$

The length of a curve $\phi : [0, 1] \times S^1 \rightarrow S^1$ in $\mathbf{Diff}(S^1)$ is obtained by integrating

$$L[\phi] = \int_0^1 \|\dot{\phi}_t \circ \phi_t^{-1}\|_{WP} dt$$

The resulting Riemannian metric is known as the Weil-Peterson metric. The classical definition was in terms of Beltrami differentials of quasiconformal self-maps of the hyperbolic plane and was originally studied in the context of Teichmüller theory. The formulation we present here occurs naturally in string theory where it was discovered as the unique Kähler metric on $PSL_2(\mathbb{R}) \setminus \mathbf{Diff}(S^1)$ [16]. See [15] for a thorough discussion of how this Kähler metric agrees with the classical WP metric.

It is a fact from variational calculus that minimizing length is equivalent to minimizing energy: the path of least energy is a constant-speed parametrization of the path of least length. Thus given a pair of points $h_0, h_1 \in \mathbf{Diff}(S^1)$ we find the geodesic by finding the path ϕ of minimum energy

$$(3) \quad E[\phi] = \int_0^1 \|\dot{\phi}_s \circ \phi_s^{-1}\|_{WP}^2 ds, \quad \phi_0 = h_0, \quad \phi_1 = h_1$$

Finding this minimizing ϕ is the problem we are concerned with. It is a fact that all sectional curvatures of the WP metric are negative [20], which in finite dimensions would imply that minimizing geodesics are unique. While the existence and uniqueness of minimizing geodesics is a subtle issue in infinite dimensions, both questions were recently answered in the affirmative in [7]: Any two points in $PSL_2(\mathbb{R}) \setminus \mathbf{Diff}(S^1)$ are joined by a unique geodesic, and further, solutions to the geodesic equation exist for all time and the resulting welding maps are Sobolev H^s for any $s < 3/2$, where the inequality is sharp.

It is an essential fact that the WP metric we have described here is, by construction, *right-invariant*. That is, given any h_0, h_1 , define the map g by $h_0 = g \circ h_1$ and let $\phi(\theta, s)$ be the shortest path from g to the identity. Right-invariance of the metric means that the length of $\phi(\theta, s)$ is the same as length of $\phi(h_1(\theta), s)$, which immediately implies that the shortest path from h_0 to h_1 is in fact $\phi(h_1(\theta), s)$. Hence, computing geodesics between two welding maps reduces to the case where one of the maps is the identity. Therefore in what follows we consider only paths ending at the identity.

1.3. Path energy and its gradient. In order to compute the gradient of path energy, we first need to put a metric on the space of admissible paths. We can then take the gradient in this metric (as opposed to simply taking the gradient in our particular choice of coordinate). Choosing a good metric greatly improves convergence and stability. This is sometimes called the “natural” gradient, particularly in the machine learning community.

Let \mathcal{P} be the space of smooth paths on $PSL_2(\mathbb{R}) \setminus \mathbf{Diff}(S^1)$. Let $\phi(s, \theta, t) = \phi_s(\theta, t)$ be a smooth curve in \mathcal{P} . Variable θ is position on S^1 , t parametrizes a path for given s , and s parametrizes the curve of paths. Then for any s the path of a particle $q(s, \theta, \cdot)$ is obtained by integrating the Eulerian vector field $v_s(\theta, t)$ defined by:

$$\phi_s(\theta, t) = \phi_s(\theta, 0) + \int_0^t v_s(\phi_s(\theta, \xi), \xi) d\xi$$

Denote the variation as

$$(4) \quad w_s(\theta, t) = \frac{\partial}{\partial s} \left(\dot{\phi}_s \circ \phi_s^{-1} \right) (\theta, t)$$

and introduce the norm

$$(5) \quad \|w_s\|_{\mathcal{P}}^2 \equiv \int_0^1 \|w_s\|_{WP}^2 dt$$

where the WP norm is taken in θ . It is in this metric on path space that we shall take the gradient.

We make one further remark: there is a natural identification between paths ϕ which end at the identity and their velocity fields $v_t = \dot{\phi}_t \circ \phi_t^{-1}$. In what follows, we prefer to take velocity fields $v(t, \theta)$ as coordinates on path space, as opposed to working with paths themselves. We will then compute the natural gradient of energy for our discrete boundary-value problem in two steps: we first compute the gradient for the unconstrained problem; i.e. we compute an update to the velocity field which simply makes the energy smaller, ignoring the boundary conditions at times 0 and 1. We then project this update (in the metric induced by (5)) onto the space of admissible updates; i.e. those updates which preserve the boundary conditions. This two-step trick is equivalent to simply taking the first variation of energy for the fixed-endpoint problem, but is considerably simpler to derive, carries little performance penalty, and, while experimenting, eliminates the requirement to recalculate the constrained gradient when the method for computing the unconstrained gradient is changed. The significant difficulty of performing numerical work in quotient spaces made this last point quite practical.

1.4. Example. At this point it might be helpful to illustrate our method with a simplified example. Since all sectional curvatures of the WP metric are negative, we make our example on the hyperbolic plane: a manifold of constant negative curvature. Since geodesics in this model are explicitly known and computable, this serves as a straightforward test case for the algorithm. We emphasize that in this example we make simple choices for the discretization in order to make the presentation clear. Finally we note that this example does not showcase some of the more nuanced aspects of our algorithm. (E.g., quotient spaces, more sophisticated quadrature, complicated boundary conditions, etc.) These more subtle points will be discussed later in the article when our algorithm is applied to the WP metric.

In the half-plane model of the hyperbolic plane, the set of points

$$H = \{(x, y) \in \mathbb{R}^2 \mid y > 0\}$$

is equipped with the distance element

$$(6) \quad ds^2 = \frac{dx^2 + dy^2}{y^2}$$

Geodesics in this model are circular arcs orthogonal to the x-axis; see Figure 2, left.

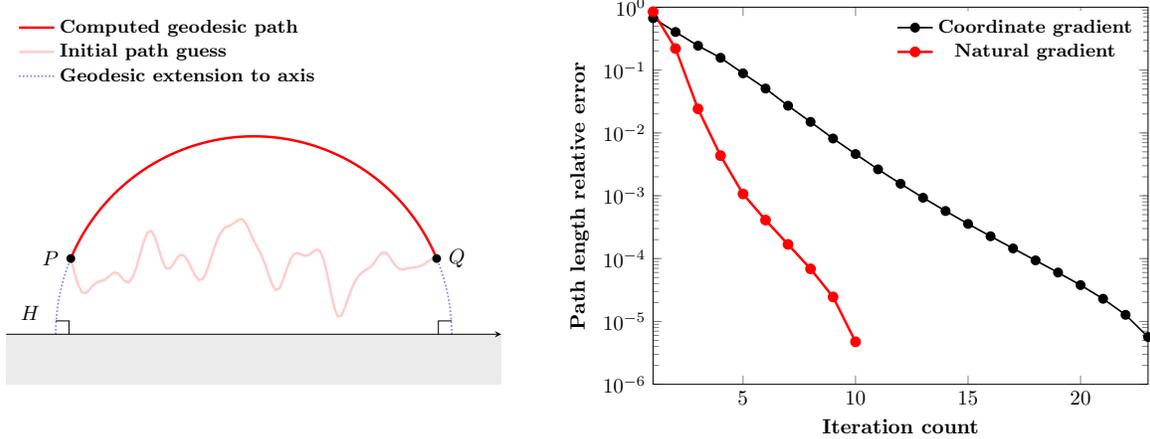


Figure 2. Left: Computed geodesic path in the hyperbolic plane model, along with a randomly-generated initial guess. Right: Path length (versus exact value) for the natural-gradient and coordinate-gradient algorithms, as a function of iteration count. The noise level is $\tau = 0.05$.

Consider two points P and Q in H , and let $((x_0, y_0), (x_1, y_1), \dots, (x_N, y_N))$ denote an ordered set that form any $(N + 1)$ -point initial guess to a geodesic connecting P and Q . We constrain the endpoints:

$$P = (x_0, y_0), \quad Q = (x_N, y_N).$$

For now we assume that the points along the initial path $q_n = (x_n, y_n)$ for $n = 1, \dots, N - 1$ are initialized in any way. A basic discretization involves a velocity field collocated at the midpoint of q_{n-1} and q_n :

$$(7) \quad v_{n-\frac{1}{2}} \triangleq q_n - q_{n-1} \in \mathbb{R}^2, \quad n = 1, \dots, N$$

Our algorithm actually works on the velocity $v_{n-\frac{1}{2}}$ and uses it to define and update the path q_n ². This discretization implies that the path length energy is approximated by summing up discrete versions of (6) collocated at midpoints between the q_n :

$$(8) \quad E \approx E_N = \sum_{n=1}^N ds_{n-\frac{1}{2}}^2 = \sum_{n=1}^N \frac{\|v_{n-\frac{1}{2}}\|^2}{y_{n-\frac{1}{2}}^2},$$

$$y_{n-\frac{1}{2}} = \frac{1}{2} (y_{n+1} + y_n)$$

Since y_n (i.e., q_n) is a function of $v_{n-\frac{1}{2}}$, this implies that the equation above is implicitly a function of $v_{n-\frac{1}{2}}$. Therefore, we can take the gradient of E_N in the $v_{n-\frac{1}{2}}$ coordinates

$\nabla E_N = \left(\frac{\partial E_N}{\partial v_{n-\frac{1}{2}}} \right)_{n=1}^N$. Energy is minimized along geodesics, so we can perform numerical gradient descent with the $v_{n-\frac{1}{2}}$ as variables to minimize E_N .

However, coordinate-dependent gradients, like the one considered above, are not the natural direction of descent for optimization algorithms in non-Euclidean geometries. When

²Note that the transformation $q_n \rightarrow v_{n-\frac{1}{2}}$ corresponds to a size $N - 1$ linear and invertible transformation: There are N total velocity variables $v_{n-\frac{1}{2}}$, and $N + 1$ total position variables q_n . Adding the boundary conditions constrains 1 velocity variable and 2 position variables.

the manifold in question has a known metric (or a suitable metric can be imposed), it is preferable to take the gradient in this metric, and this is sometimes referred to as the *natural* gradient [3]. In our example, the distance element is proportional to $1/y$, so updating the $\{q_n\}$ according to the coordinate gradient permits particles near the boundary to take arbitrarily large steps. In this case, (8) implies that in our $v_{n-\frac{1}{2}}$ coordinate system, a diagonal matrix with entries $G_{n,n} = y_{n-\frac{1}{2}}^{-2}$ will give us a metric tensor on path space which will cause path updates to respect the metric. Therefore, the direction $G^{-1}\nabla E_N$ is the steepest descent direction.

However, this natural gradient direction does not respect the endpoint conditions. (I.e., updating $v_{n-\frac{1}{2}}$ and then computing the path positions q_n via (7) will not in general satisfy both $P = q_0$ and $Q = q_N$. To correct for this, we project $G^{-1}\nabla E_N$ into the space of admissible updates that do satisfy the endpoint constraints. We omit presenting the explicit details of this here, but the general continuous version of the projection is (16) discussed in Section 2.2. The discrete (matrix) version of this projection in the WP metric case is given in (30).

The high-level description of the algorithm can now be completed: at each stage with a given v_n we compute the constrained natural gradient of E_N , and use it to perform steepest descent; we iterate until convergence.

For our numerical examples here, we initialize the geodesic path guess as a random perturbation of a Euclidean straight line:

$$q_n = P + \frac{n}{N+1}(Q - P) + \sigma_n, \quad n = 1, \dots, N-1$$

where $(\sigma_n)_{n=1}^{N-1}$ is treated as a Gaussian process with covariance structure $\text{cov}(\sigma_n, \sigma_m) = \tau \exp\left(-\frac{(n-m)^2}{(N+1)\ell}\right)$, with correlation length ℓ and noise magnitude τ . In our examples below, we set $\ell = \frac{1}{500}$, and τ varies between 0 and 0.1. (See Figure 2 left, for an example realization of this process.)

To illustrate the effectiveness of the natural gradient versus the coordinate gradient we set $\tau = 0.05$ and $N = 100$, and compare the different gradient iteration schemes in Figure 2 right. The natural gradient converges to the true geodesic much more quickly than the coordinate gradient. In fact, in flat spaces it will converge in a single iteration.

The natural gradient is also more robust with respect to discretization error and initial guesses. In Figure 3 we show the required number of iterations until convergence for various values of N and τ . Since our initial guess is a random process, we run a size-50 ensemble of simulations, and the plots show ensemble average iteration counts. The natural gradient algorithm is quite insensitive to both noise level and discretization parameters, whereas the coordinate gradient is quite sensitive and also requires more iterations. In what follows we will use a very similar metric to stabilize updates to our discrete path in shape space.

2. THE DISCRETE PROBLEM

We now return to the problem (3) in the case where $h_0 : S^1 \rightarrow S^1$ is an arbitrary welding map and h_1 is the identity map. This is a boundary-value minimization problem: find $v_s = \dot{\phi}_s \circ \phi_s^{-1}$ subject to initial and terminal conditions $\phi_0 = h_0$ and $\phi_1 = \text{Id}$ such that $\int \|v_s\|_{WP}^2 ds$ is minimized. We introduce $q \equiv \phi$, and given a velocity field v then

$$q(t, x) = q(0, x) + \int_0^t v(\xi, q(\xi, x)) d\xi$$

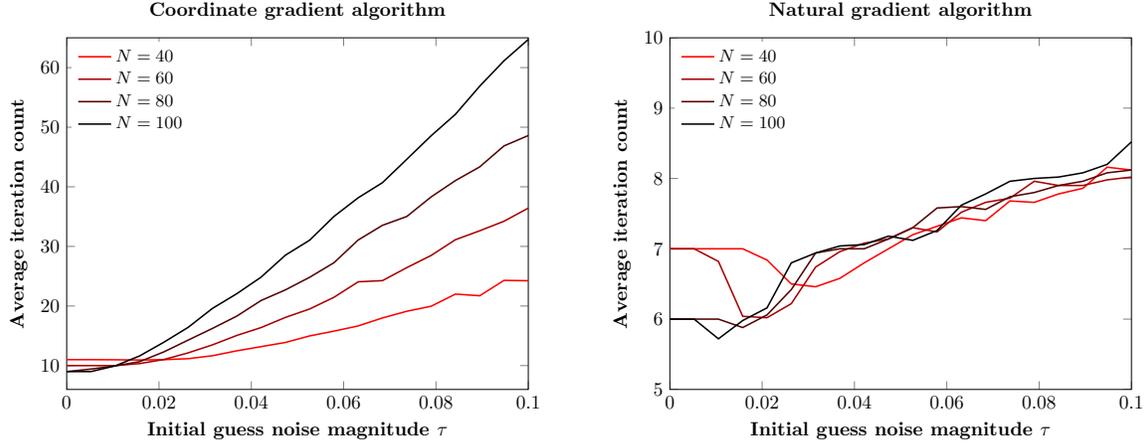


Figure 3. Ensemble-averaged iteration counts required to converge to hyperbolic plane geodesic from a randomly-perturbed Euclidean straight line initial guess (see Figure 2, left). Left: Coordinate gradient algorithm. Right: Natural gradient algorithm.

for all $0 \leq t \leq 1$.

We track M particles on the range of q as they evolve in t . For some fixed x_m on S^1 , $q(t, x_m)$ are the locations of these particles. The velocity field $v(t, \cdot)$ at the locations $q(t, x_m)$ is likewise stored. We store these velocity field values and recover the particle positions $q(t, x_m)$ by integrating rather than directly storing the particle positions. In the next subsection we will describe how we compute the WP norm at each time. We delay discretizing the t variable until Section 3.

2.1. Computing the WP Norm. We first consider the problem of computing the norm of a vector field on S^1 . Subsequent sections will consider computation of path energy and its gradient to find geodesics.

Our discretization provides us with information about the velocity field only at a discrete set of points. Our first task is to interpolate this velocity field to a field defined on all of S^1 and then compute the norm of this lift. The computation proceeds in 3 steps. We first collect some standard results which let us do this when the metric has no kernel. Next we compute the optimal interpolant and its norm when the metric has a kernel; this is done by projecting the interpolation data onto the orthogonal complement of the kernel and computing the norm of the lift. Finally, we generalize the problem to allow an arbitrary basis for the interpolating functions; this allows for more flexibility in the implementation and also suggests a more stable method of computation, which we detail later.

2.1.1. Lifting the velocity field into the orthogonal complement of the kernel. We need to extend a vector field defined on a finite subset of S^1 to a vector field on all of S^1 . Here we describe a method for finding the smoothest possible extension using Green's functions for the operator L . In this section we consider only vector fields orthogonal to the kernel; we extend to more general cases next.

Consider the space of all vector fields on S^1 , and a subspace on which $\|\cdot\|_{WP}$ is a proper norm:

$$V = \{v : \|v\|_{WP} < \infty\}, \quad \tilde{V} = \{v \in V : v \perp \ker_V L\},$$

where, explicitly, $\ker_V L = \text{span}\{1, \sin, \cos\}$. Let \mathcal{L}^M be the M -manifold of configurations of M points $q_1 < q_2 < \dots < q_M$ on the unit circle and write $\mathcal{Q} = \{q_1, q_2, \dots, q_M\}$. The tangent space $T_{\mathcal{Q}}\mathcal{L}^M$ is the M -dimensional space of vector fields $v = (v_1, v_2, \dots, v_M)$ supported on the set \mathcal{Q} . We consider all possible extensions of v to vector fields defined on the entire circle, and induce a norm on $T_{\mathcal{Q}}\mathcal{L}^M$ by

$$(9) \quad \|v\|_{WP(\mathcal{Q})} \equiv \inf \{ \|\tilde{v}\|_{WP} : \tilde{v}(q_m) = v_m, 1 \leq m \leq M \}$$

Conversely, given the basepoint \mathcal{Q} , consider the evaluation map taking a global vector field \tilde{v} , defined on all of S^1 , to $T_{\mathcal{Q}}\mathcal{L}^M$ by

$$\tilde{v} \rightarrow (\tilde{v}(q_1), \tilde{v}(q_2), \dots, \tilde{v}(q_M))$$

This map induces a splitting of any \tilde{v} into two components. The set of vector fields for which $\tilde{v}(q_m) = 0$ for all $1 \leq m \leq M$ is the *vertical subspace* (of the *Lie algebra* of $\mathbf{Diff}(S^1)$: the set of smooth vector fields on S^1), and its WP -orthogonal complement is the *horizontal subspace*; the minimizing extension in (9) of $v \in T_{\mathcal{Q}}\mathcal{L}^M$ is known as a *horizontal lift*.

As the next proposition shows, the horizontal subspace is an M -dimensional subspace spanned by translates of Green's function for the operator $L = -\mathcal{H}(\partial^3 - \partial)$. This Green's function is known explicitly, see [10]:

$$G(\theta) = 2 \sum_{n=2}^{\infty} \frac{\cos(n\theta)}{(n^3 - n)} = (1 - \cos \theta) \log [2(1 - \cos \theta)] + \frac{3}{2} \cos \theta - 1$$

The following results are standard facts which may be proven using the reproducing kernel property of Green's function and the fact that the vector fields $G(\theta - q_m) \frac{\partial}{\partial \theta}$ lie in the horizontal subspace.

Proposition 1 (Computing Horizontal lifts).

(1) *The set of vector fields*

$$\left\{ G(\theta - q_m) \frac{\partial}{\partial \theta} \right\}_{m=1}^M$$

is a basis for the component of the horizontal subspace in \tilde{V} .

(2) *The horizontal lift on \tilde{V} of the tangent vector $(v_1, \dots, v_M) \in T_{\mathcal{Q}}\mathcal{L}^M$ is the vector field $\tilde{v}(\theta) \frac{\partial}{\partial \theta}$ for*

$$\tilde{v}(\theta) = \sum_{i=1}^M G(\theta - q_i) p_i \quad \text{where} \quad p_i = \sum_{j=1}^M G_{ij}^{-1} v_j$$

and G_{ij} is the positive definite symmetric matrix

$$(10) \quad G_{ij} = \langle G(\cdot - q_i), G(\cdot - q_j) \rangle_{WP} = G(q_i - q_j)$$

Further,

$$\|\tilde{v}\|_{WP}^2 = \sum_{i,j} G_{ij}^{-1} v_i v_j$$

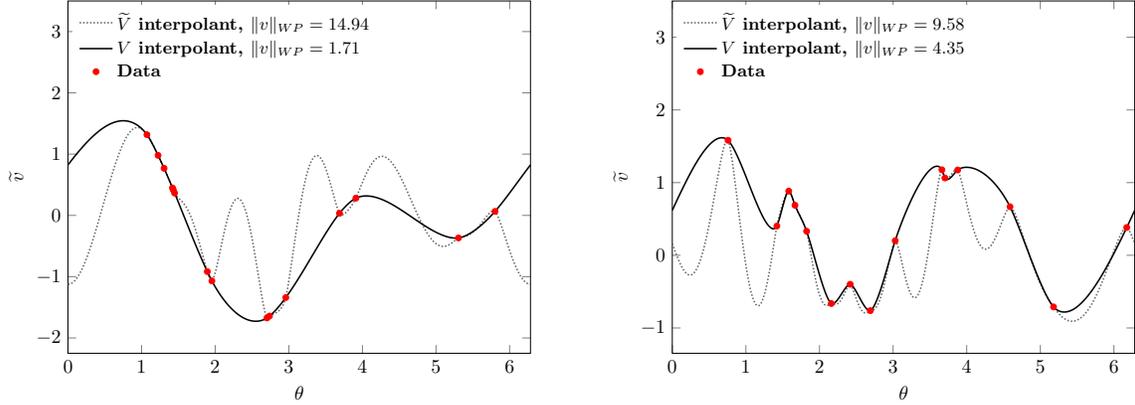


Figure 4. Interpolation procedures from (a) Proposition 1 on the quotient space \tilde{V} , and from (b) Corollary 1 on V . Left: the data is taken from the function $v(\theta) = \sin(2\theta) + \cos(\theta)$ on 15 randomly distributed θ locations. Right: the data is taken as Gaussian random variable perturbations of function values from $v(\theta) = \sin(2\theta)$ on 15 randomly distributed θ locations.

2.1.2. *Lifting the velocity field.* Now we extend these results to compute horizontal lifts which take into account the kernel.

Lemma 1. *Let G be an $M \times M$ positive matrix, B an $M \times K$ full-rank matrix with $K < M$. Then given some $v \in \mathbb{R}^M$, the projection of v onto the G^{-1} -orthogonal complement of the kernel of B^T is given by*

$$P_G^B v \doteq (I - B(B^T G^{-1} B)^{-1} B^T G^{-1})v$$

In our case B is an $M \times 3$ matrix containing point evaluations of a basis for the three-dimensional kernel of the WP norm. Once we have subtracted out the contribution of the kernel, we may compute the lift and its norm. The next result is a corollary of section 2.1.1 and the previous lemma.

Corollary 1. *Fix \mathcal{Q} and let the matrix G be as in Proposition 1. For any $v \in T_{\mathcal{Q}}\mathcal{L}^M$,*

$$(11) \quad \|v\|_{WP(\mathcal{Q})}^2 = v^T G^{-1} P_G^B v$$

Further, the horizontal lift \tilde{v} of v to V is given by

$$\tilde{v} = \sum_{j=1}^3 w_j b_j(\cdot) + \sum_{j=1}^M (G^{-1} P_G^B v)_j G(\cdot - q_j)$$

where

$$w = (B^T G^{-1} B)^{-1} B^T G^{-1} v$$

and $\{b_j\}_{j=1}^3$ is the basis for $\ker v L$ used to construct B : $B_{m,k} = b_k(q_m)$.

Remark. The vector $p = G^{-1} P_G^B v$ contains what are known as the momenta coefficients. Similarly, we call w the kernel coefficients.

The interpolant produced from Corollary 1 is notably different than the horizontal lift from the quotient space in Proposition 1. This is exemplified in Figure 4.

Up to this point, we have taken Green's functions centered at the interpolation nodes as the basis for our interpolating space. As discussed, this produces the norm-minimal lift.

While this optimality is nice, and is in fact what our current implementation ultimately uses, one might wish to choose a different space of interpolating functions. For example, the first N complex exponentials, or a wavelet basis adapted to the problem might be reasonable choices. We address this now; it is only a small modification to the calculations above and admits a stable solution which we present later.

Let \tilde{V}_N be any N -dimensional subspace of \tilde{V} such that the interpolation problem for data collocated at \mathcal{Q} is unisolvent. If $F = \{f_n(\cdot)\}$ is a basis of \tilde{V}_N , define the $M \times N$ matrix λ_F to have entries

$$(\lambda_F)_{m,n} = \langle f_n, G(\cdot - q_m) \rangle_{WP} = f_n(q_m)$$

Let G_F be the $N \times N$ Gram matrix for the f_n :

$$(G_F)_{m,n} = \langle f_m, f_n \rangle_{WP}$$

For $v \in \mathbb{R}^M$ interpolation data on \mathcal{Q} , define $\tilde{V}_N(v)$ as the subset of functions from $\tilde{V}_N \oplus \ker_V L$ that interpolate to v . Then define

$$\|v\|_{WP(\mathcal{Q}), \tilde{V}_N}^2 = \min_{v \in \tilde{V}_N \oplus \ker_V L} \|v\|_{WP}^2$$

The results from the previous section may be rephrased by replacing $G \leftarrow G_F$ where $G_F = \lambda_F G_F^{-1} \lambda_F^T$. Since \tilde{V}_N is total for interpolation, then G_F is invertible. One finds that all formulas carry over essentially up to change-of-basis. This, then, is our most general result for computing lifts, which we state in terms of an operator L_F .

Proposition 2. *In the notation above, set*

$$L_F = G_F^{-1} P_{G_F}^B$$

Then

$$(12) \quad \begin{aligned} \|v\|_{WP(\mathcal{Q}), \tilde{V}_N}^2 &= v^T L_F v \\ &=: v^T p. \end{aligned}$$

The reconstruction $\tilde{v} \in \tilde{V}_N \oplus \ker_V L$ is given by

$$\tilde{v} = \sum_{n=1}^N c_n f_n + \sum_{j=1}^3 w_j b_j$$

where

$$(13a) \quad w = (B^T G_F^{-1} B)^{-1} B^T G_F^{-1} v$$

$$(13b) \quad c = G_F^{-1} \lambda_F^T G_F^{-1} (v - Bw).$$

Note that unlike the result of Corollary 1, the momenta $p \equiv L_F v$ are no longer the reconstruction coefficients for the lift unless \tilde{V}_N is the span of Green's functions centered at \mathcal{Q} . In any case, L_F is a symmetric, rank $M - 3$, positive semidefinite matrix.

Whether using Green's function interpolation or another basis, using (12) and (13) to compute either the norm or any coefficients can be problematic. The matrices in the explicit formula given can easily be ill-conditioned, depending on F and \mathcal{Q} . We do not directly solve this issue here, but instead we reformulate the problem in a format that we observe is more robust with respect to numerical precision issues than the above formulas. This is done in Section 3.

2.2. Path energy and its gradient. Computing the length and energy of a path is relatively straightforward given the work in the previous section. If v^t is a vector of velocity evaluations at time g , we have

$$E = \int_0^1 \|\tilde{v}\|_{WP}^2 dt \int_0^1 v^t L_F^t v^t dt,$$

where L_F^t is the metric on the particle positions q^t .

As mentioned above, we compute the gradient of energy for our discrete boundary-value problem in two steps. We first compute the gradient ignoring the boundary conditions at times 0 and 1. We then project this update onto the manifold of updates which preserve the boundary conditions. Since we have yet to discretize the temporal variable, we wait to compute the unconstrained gradient until Section 3. For now we concentrate on the projection onto the admissible updates.

Define the energy of a velocity field as above. Now consider two manifolds of velocity fields. \mathcal{N} contains all velocity fields defined on M particles at each time, and $\mathcal{M} \subset \mathcal{N}$ contains those which respect the boundary-value conditions.

$$\begin{aligned} \mathcal{N} &= \{\tilde{v} : \{1, \dots, M\} \times [0, 1] \rightarrow \mathbb{R}\} \\ \mathcal{M} &= \left\{ V \in \mathcal{N} \left| q_m(1) = q_m(0) + \int_0^1 \tilde{v}(\xi, q_m(\xi)) d\xi \right. \right\} \end{aligned}$$

We defined the energy $E : \mathcal{N} \rightarrow \mathbb{R}$ as a function on \mathcal{N} , and in the next section we will describe the computation of its gradient ∇E on \mathcal{N} . With this gradient in hand, we obtain the direction of steepest ascent as a projection of ∇E onto the manifold \mathcal{M} by projection in the following metric.

Let \mathcal{P} be the space of smooth paths on $PSL_2(\mathbb{R}) \backslash \mathbf{Diff}(S^1)$. Let $q(t, x, s) = q_{t,s}(x)$ be (a representative of) a smooth curve in \mathcal{P} . Variable x is position on S^1 , t parametrizes a path for given s , and s parametrizes the curve of paths. Recall that the velocity $v_{t,s}$ along the path is defined by

$$q(t, x, s) = q(0, x, s) + \int_0^t v_{\xi,s}(q(\xi, x, s)) d\xi$$

and as described above, we write

$$(14) \quad w_{t,s} = \frac{\partial}{\partial s} (\dot{q} \circ q^{-1})_{t,s}$$

and introduce the norm

$$(15) \quad \|w_{t,s}\|_{\mathcal{P}}^2 \equiv \int_0^1 \|w_{t,s}\|_{WP}^2 dt$$

However, we wish to work with the velocity field manifolds \mathcal{N} and \mathcal{M} as our coordinate on \mathcal{P} . With $q = q(t, x, s)$, $\dot{q} = \frac{\partial}{\partial t} q$, and $q' = \frac{\partial}{\partial x} q$, we have

$$\begin{aligned} \frac{\partial (\dot{q} \circ q^{-1})_{t,r}}{\partial r}(x) &= \left(\frac{\partial \dot{q}}{\partial s} \circ q^{-1} \right)_{t,r} + (\dot{q} \circ q^{-1})'_{t,r} \left(\frac{\partial q^{-1}}{\partial s} \right)_{t,r} \\ &= \left(\frac{\partial v}{\partial s} \right)_{t,r} + v'_{t,r} \left(\frac{\partial q^{-1}}{\partial s} \right)_{t,r} \\ &= \left(\frac{\partial v}{\partial s} \right)_{t,r} - v'_{t,r} (q^{-1})'_{t,r} \left(\frac{\partial q}{\partial s} \circ q^{-1} \right) \end{aligned}$$

Direct computation yields

$$\begin{aligned} \frac{\partial q_s}{\partial s}(t, x) &= \frac{\partial}{\partial s} q(0, x, s) + \frac{\partial}{\partial s} \int_0^t v_{\xi;s}(q(\xi, x, s)) d\xi \\ &= \int_0^t \frac{\partial}{\partial s} v_{\xi;s}(q(\xi, x, s)) d\xi \end{aligned}$$

Therefore

$$\frac{\partial (\dot{q} \circ q^{-1})_{t;r}(x)}{\partial r} = \mathcal{Y} \left[\frac{\partial v}{\partial s} \right],$$

where the linear operator \mathcal{Y} is given by

$$(16) \quad \mathcal{Y}(t, x, s) = \text{Id} - v'(q^{-1})' \int_0^t (\cdot)_\xi d\xi$$

This operator \mathcal{Y} gives a mapping from velocity field updates (i.e. tangent vectors to \mathcal{N} or \mathcal{M}) to tangent vectors $\frac{\partial q}{\partial s}$ to \mathcal{P} . Hence our metric on path space immediately induces a metric on \mathcal{N} and the submanifold \mathcal{M} . We make our projection of the update to the velocity field with respect to this metric.

3. COMPUTATIONS

We now address the more practical issues of implementation of the methods described. We first formulate computation of the *WP* norm as a classical numerical linear algebra problem. Following this is presentation of the temporal discretization; having discretized both the temporal and S^1 variables, we can compute the unconstrained coordinate gradient of the energy. Finally, the submanifold projection for gradient updates is discretized.

3.1. The WP norm. It is not necessary to explicitly form the metric matrix L_F as defined in Proposition 2. We can accomplish the same task in a more stable manner using some established numerical linear algebra results. Given an $M \times N$ matrix A , A^\dagger denotes its Moore-Penrose pseudoinverse. For any $\mathbf{b} \in \mathbb{R}^N$, the vector $A^\dagger \mathbf{b}$ is the minimum-norm, least squares solution to $A\mathbf{x} = \mathbf{b}$. Here, both minimum-norm and least squares refer to the ℓ^2 (Euclidean) metric on \mathbf{x} and \mathbf{b} , respectively.

For computation of the *WP* norm, we develop the ideas for a general subspace V_N of V and then specialize to the minimal-norm lift over all V using Green's functions. Our goal is the interpolation of data $\mathbf{v} \in \mathbb{R}^M$ collocated at \mathcal{Q} with an element from $\tilde{V}_N \oplus \ker_V L$; thus we are essentially trying to find the minimal norm solution to

$$(17) \quad \begin{bmatrix} \lambda_F & B \end{bmatrix} \begin{pmatrix} \mathbf{c} \\ \mathbf{w} \end{pmatrix} =: A\mathbf{x} = \mathbf{v},$$

where the norm over the coefficients \mathbf{c} is given by the positive (Gram) matrix G_F and that over the kernel coefficients \mathbf{w} is zero. Again we assume that \tilde{V}_N is total for interpolation on \mathcal{Q} , implying that $A\mathbf{x} = \mathbf{v}$ has at least one solution. Let G_{F_0} be a block diagonal matrix with G_F in the upper-left block and a 3×3 zero matrix in the lower-right block. It is easy to show that $\ker(B)$ is trivial so long as $M \geq 3$. This implies that $\ker(G_{F_0}) \cap \ker(A)$ is trivial and therefore the minimum norm solution to (17) is unique [6]. This is an easy way to show uniqueness of the lift defined in (9).

Restating (17), we are trying to solve the minimization problem:

$$(18) \quad \text{minimize} \quad \|R\mathbf{x}\|_2 \quad \text{subject to} \quad A\mathbf{x} = \mathbf{v},$$

where R is any matrix square root of G_{F_0} . (I.e. $R^T R = G_{F_0}$.) One of the standard tools for solving least squares equality constrained problems is the generalized singular value decomposition (GSVD) [12]. Since A and $G(F)_0$ have the same number of columns, the GSVD matrix decomposition of (A, R) is given by

$$(19a) \quad A = UCX^{-1}$$

$$(19b) \quad R = VSX^{-1},$$

where U and V are orthogonal matrices, X is invertible, and C and S each have only one non-vanishing diagonal with non-negative entries and satisfy $C^T C + S^T S = I$. (Here, V denotes a matrix in the GSVD decomposition, and not a linear subspace.) The solution to (18) is given by

$$(20) \quad \mathbf{x} = \begin{pmatrix} \mathbf{c} \\ \mathbf{w} \end{pmatrix} = XC^\dagger U^T \mathbf{v},$$

The above matrix mapping \mathbf{v} to \mathbf{x} is a weighted pseudoinverse of A with respect to R . The momenta are $\mathbf{p} = UC^{\dagger T} X^T G_{F_0} \mathbf{x} = L_F \mathbf{v}$. Our experience is that this is a much more stable way to compute momenta than the explicit matrix relations used to define L_F . Note that since C^\dagger is a matrix with only one non-vanishing diagonal, applications of C^\dagger can be accomplished with a simple vector-vector multiply. The norm is $\|\mathbf{v}\|_{WP(\mathcal{Q}), \tilde{V}_N} = \mathbf{c}^T G_F \mathbf{c} = \mathbf{v}^T \mathbf{p}$.

For the horizontal lift from the whole space V , we need only take the space \tilde{V}_N to be the M -dimensional space of Green's functions centered at the q_m . Notable simplifications in this case are that (a) $G_F = G$, with G being the Gram matrix of Green's functions (10) and (b) the momentum and basis coefficients coincide: $\mathbf{p} = \mathbf{c}$.

3.2. Temporal discretization. Since flow along the geodesic is reversible, we use a quadrature method that is symmetric with respect to the endpoints. We rewrite the flow of particles in an equivalent form

$$(21) \quad q_m(t) = \frac{1}{2} \left(q_m(0) + \int_0^t \tilde{v}(\xi, q_m(\xi)) d\xi \right) + \frac{1}{2} \left(q_m(1) - \int_t^1 \tilde{v}(\xi, q_m(\xi)) d\xi \right)$$

Now choose T ordered points s^t on $(0, 1)$; let \mathbf{q}^t and \mathbf{v}^t denote the M particle positions and velocities, respectively, at those times. The endpoint particle positions \mathbf{q}^0 and \mathbf{q}^{T+1} are given. Let h^t denote the quadrature weight associated with data at time s^t so that $\int_0^1 f(s) ds \simeq \sum_{t=1}^T h^t f(s^t)$.

Following (21) we use the quadrature rule to integrate particle positions forward from \mathbf{q}^0 to \mathbf{q}^t , and backwards from \mathbf{q}^{T+1} to \mathbf{q}^t , and average the result. For example if the representation is piecewise-linear, then the symmetric velocity-to-particle map is

$$(22) \quad \mathbf{q}^t = \frac{1}{2} \left(\mathbf{q}^0 + \sum_{r=1}^{t-1} h^r \mathbf{v}^r + \frac{h^t}{2} \mathbf{v}^t \right) + \frac{1}{2} \left(\mathbf{q}^{T+1} - \sum_{r=t+1}^T h^r \mathbf{v}^r - \frac{h^t}{2} \mathbf{v}^t \right), \quad t \geq 1$$

The scheme described above expresses particle positions \mathbf{q}^t linearly with the velocity field values \mathbf{v}^r . Collect all the velocity evaluations into a matrix V , of size $M \times T$; do the same for the particle positions in a matrix Q . Then regardless of the choice of linear temporal quadrature rule, there exists a $T \times T$ matrix Z such that

$$(23) \quad Q = \frac{1}{2} (Q^0 + Q^{T+1}) + VZ,$$

where Q^0 and Q^{T+1} are matrices with the known vectors \mathbf{q}^0 and \mathbf{q}^{T+1} repeated. The entries of the matrix Z depends on the choice of temporal representation and quadrature. For the piecewise-linear choice, it has entries $Z_{r,t} = \frac{1}{2}c_{r,t}h_r$, where

$$(24) \quad c_{r,t} = \begin{cases} 0, & r = t, \\ 1, & r < t, \\ -1, & r > t. \end{cases}$$

In order to evaluate the energy $\int_0^1 \|v_s\|_{WP}^2 ds$, we build the quadrature factor h^t into the norm at each point in time. This can be accomplished by simply replacing $G(F)$ by $h^t G(F)$ at each point in time. Therefore

$$(25) \quad E = \sum_{t=1}^T \|\mathbf{v}^t\|_{WP(\mathcal{Q}), \tilde{V}_N}^2 = \sum_{t=1}^T (\mathbf{v}^t)^T L_F \mathbf{v}^t$$

where L_F is time-varying, depending explicitly on \mathbf{q}^t , the particle locations at time s^t , and also depending proportionally on h^t .

To keep the presentation simple we have described use of a piecewise-linear quadrature. In practice we find it is sometimes more efficient to use a Legendre-Gauss-Lobatto quadrature rule for $t \in [0, 1]$, which enables a high-order polynomial representation for each particle's velocity field. When to use a high-order representation versus a piecewise linear representation depends on whether one expects large t -derivatives in the velocity field.

3.3. The gradient of the WP norm. At some time s^t , we have seen that if $L_F = G_F^{-1} P_{G_F}^B$ then formula (25) defines the norm, and to minimize we must compute variations of this with respect to velocity evaluations. Since the particle positions are influenced by (23), the matrices L_F and therefore the energy change in nontrivial ways when we vary \mathbf{v}^t . To simplify the procedure, we first fix t and consider only variations in $\|\mathbf{v}^t\|_{WP(\mathcal{Q})}$ with respect to \mathbf{v}^r for $r = 1, \dots, T$.

Direct variation of the quadratic form yields

$$(26) \quad \begin{aligned} \frac{\partial}{\partial \mathbf{v}^r} \|\mathbf{v}^t\|_{WP(\mathcal{Q}), \tilde{V}_N}^2 &= 2\delta_{t,r} (\mathbf{v}^t)^T L_F + (\mathbf{v}^t)^T \left(\frac{\partial}{\partial \mathbf{v}^r} L_F \right) \mathbf{v}^t \\ &= 2\delta_{t,r} (\mathbf{p}^t)^T + (\mathbf{v}^t)^T \left(\frac{\partial}{\partial \mathbf{v}^r} L_F \right) \mathbf{v}^t \end{aligned}$$

where $\delta_{i,j}$ is the Kronecker delta. The metric L_F and its components G_F and $P_{G_F}^B$ are generated from \mathbf{q}^t , depending on \mathbf{v}^t . Using simple properties of matrix algebra, a straightforward but messy computation yields a formula for the variation of the objective with respect to the particle positions.

Lemma 2. *Let $\lambda(F')$ be the $M \times N$ matrix with entries $\lambda(F')_{m,n} = f'_n(q_m)$ and B' be the $M \times 3$ matrix with entries $B'_{m,j} = b'_j(q_m)$. For two matrices C and D of the same size, $C \circ D$ denotes the elementwise product. Then*

$$(27) \quad \frac{\partial}{\partial \mathbf{q}^t} \|\mathbf{v}^t\|_{WP(\mathcal{Q}), \tilde{V}_N}^2 = -2 [\mathbf{p}^t \circ (\lambda(F') G(F)^{-1} \lambda(F)^T \mathbf{p}^t + B' \mathbf{w}^t)]^T.$$

In particular, if F is a basis of Green's functions centered at \mathcal{Q} , then

$$(28) \quad \frac{\partial}{\partial \mathbf{q}^t} \|\mathbf{v}^t\|_{WP(\mathcal{Q})}^2 = -2 [\mathbf{p}^t \circ \tilde{v}'(\mathbf{q}^t)]^T,$$

where \tilde{v}' is the derivative of the horizontal lift from V .

The salient result of Lemma 2 is that computing variations of the WP norm with respect to particles is quite easy once we have the momentum coefficients \mathbf{p} and kernel coefficients \mathbf{w} in hand from the GSVD. The only additional difficulty could come in computing entries for $\lambda(F')$. In all the straightforward choices for F we make, it is not a deterrent.

3.4. The gradient of energy. With the gradient of the WP norm computed, computing the energy is now straightforward. Combining (23), (26), and (27), we have

$$(29) \quad \frac{1}{2} \frac{\partial E}{\partial \mathbf{v}^t} = (\mathbf{p}^t)^T - \sum_r Z_{t,r} [\mathbf{p}^t \circ (\lambda(F')G(F)^{-1}\lambda(F)^T \mathbf{p}^t + B' \mathbf{w}^t)]^T$$

where again we recall that the momenta coefficients \mathbf{p}^t already have the scaling factor h^t included.

Although the above equation is an exact formula for the gradient, updating velocity fields with these values will not respect the endpoint constraint $\mathbf{q}^1 = \mathbf{q}^0 + \sum_t h_t \mathbf{v}^t$, and so we project into the space of admissible updates.

The discrete versions of the manifolds presented in Section 2.2 can be defined as follows:

$$N = \{\mathbf{v} : [1, \dots, M] \times [1, \dots, T] \rightarrow \mathbb{R}\}$$

$$M = \left\{ \mathbf{v} \in N \left| \mathbf{q}^1 = \mathbf{q}^0 + \sum_t h_t \mathbf{v}^t \right. \right\}$$

Since the condition $\mathbf{q}^1 = \mathbf{q}^0 + \sum_t h_t \mathbf{v}^t$ is linear, we may write it as

$$\mathbf{q}^1 - \mathbf{q}^0 = A\mathbf{v}$$

for some constraint matrix A . We see that M is an implicit submanifold of N : it is the $\mathbf{q}^1 - \mathbf{q}^0$ level set of the function $A\mathbf{v}$. By the implicit function theorem, the tangent space to M at any point is equal to the nullspace of the differential DA , and we have simply $[DA] = A$. That is, the admissible updates to a given velocity field V are exactly those vectors \mathbf{v} in the nullspace of A .

In order to perform the appropriate projection onto the nullspace of A , we place our update along a curve on the space of paths \mathcal{P} and apply the norm induced by the metric (14) and (15). We then seek a discretization of the semidefinite operator $\mathcal{Y}^\dagger L \mathcal{Y}$ given by (16) and (2), where \mathcal{Y}^\dagger is the L^2 adjoint of \mathcal{Y} .

At a fixed value of s^t , we have already approximated L with L_F depending on \mathbf{q}^t . Therefore we concentrate on \mathcal{Y} : the integral $\int_0^{s^t} (\cdot)_\xi d\xi$ is approximated by the t -th column of Z . Let $\tilde{Z} = Z^T \otimes I_M$, where I_M is the $M \times M$ identity matrix; then $\mathbf{q} = \tilde{Z}\mathbf{v}$. The factor $v'(\mathbf{q})$ can be computed at each t via $v'(\mathbf{q}^t) = \lambda(F')\mathbf{c}^t + B'\mathbf{w}^t$. We collect these factors into a diagonal $MT \times MT$ matrix W with the t -th block diagonal entry $(W^t)_{j,j} = v'(q_j^t)(q^{-1})'(q_j^t)$. The result then is the modified metric on updates $\frac{\partial}{\partial \mathbf{v}} E$:

$$(30) \quad \tilde{L}_F = (I - W\tilde{Z})^T L_F (I - W\tilde{Z}).$$

I.e., the matrix $Y := I - W\tilde{Z}$ is our approximation to the operator \mathcal{Y} . We use this metric both to form the natural gradient and to orthogonally project into the nullspace of the constraint matrix A .

4. ALGORITHM AND RESULTS

With all the necessary derivations complete, a summary of the minimization algorithm is presented in Algorithm Listing 1. One detail we have not yet addressed is a method for computing welding maps – i.e. for determining ϕ given a discrete collection of ordered samples on $\partial\Omega$. A few standard methods exist for numerically computing conformal welds [5, 11, 18, 19], and we settle upon the Zipper algorithm [13]; specifically we use the (simplest) ‘geodesic’ version. We use the Zipper algorithm to (a) construct a representation of a conformal weld given discrete samples on the shape boundary $\partial\Omega$, (b) ‘interpolate’ the welding map to any starting particle locations $\mathbf{q}^0, \mathbf{q}^{T+1}$ of our choosing, and (c) ‘invert’ the weld: compute a representation of a simple closed curve from discrete samples of a homeomorphism of S^1 . There are other possibilities for accomplishing these conformal welding tasks [17].

Algorithm 1 A simple gradient descent algorithm for optimization.

Input: Initial and final particle positions \mathbf{q}^0 and \mathbf{q}^{T+1}
Initialize: $\mathbf{v}^t = \mathbf{q}^{T+1} - \mathbf{q}^0$ for all t
while Not converged **do**
 Compute norm, momenta, and kernel coefficients from (20)
 Compute standard unconstrained gradient ∇E from (29) and (27)
 Form projected natural gradient $\widetilde{\nabla E}_n$ from (16) and (30).
 Find ε such that $E(V - \varepsilon \widetilde{\nabla E}_n) < E(V)$
 Update $V \leftarrow V - \varepsilon \widetilde{\nabla E}_n$, update particles using (23)
end while
Output: Velocity field V

Our stopping criterion for the optimization iteration is defined by monitoring the relative objective decrease at each step in tandem with the Euclidean vector 2-norm of the projected natural gradient (normalized with respect to the size of the vector). At convergence, the former is less than 10^{-8} and the latter is less than 10^{-6} .

One necessary measure of convergence is constancy of the WP norm along the path. For all the results we show, the relative variation measure $(\max_t \|\mathbf{v}^t\|_{WP(\mathcal{Q})}^2) / (\min_t \|\mathbf{v}^t\|_{WP(\mathcal{Q})}^2) - 1$ is no larger than 10^{-3} and is usually $\mathcal{O}(10^{-6})$.

In all the tests we use 150 particles. The full set of particles is obtained by computing 50 uniformly distributed particle samples on the ranges of the two welding maps ϕ_0 and ϕ_1 , and 50 on the domain of the welding maps, and then taking the union of the sets. This encourages inclusion of resolvable features from both the interior and exterior of the welds in the computation.

For the temporal discretization, we have found that using a high-order Gauss-Lobatto quadrature rule works very well for welding maps whose derivatives are not large. When the welding maps exhibit high frequency content (a result of e.g. protrusions in the shape), then a standard piecewise linear choice works well. In this latter case we use 150 equispaced points in the time variable.

4.1. Zero-distance welds. As a first test we compute the energies of the paths connecting welding maps in the same equivalence class (which are expected to vanish). The welding maps are the maps ϕ_i in Figure 1. In Table 5 we show the computed energy at convergence. The shown near-vanishing energies confirms validity of the computation.

Computed path energy between welds in Figure 1.

	ϕ_1	ϕ_2	ϕ_3	ϕ_4
ϕ_1	—	—	—	—
ϕ_2	7.530E-12	—	—	—
ϕ_3	1.249E-9	1.729E-8	—	—
ϕ_4	2.853E-9	6.633E-8	2.069E-10	—

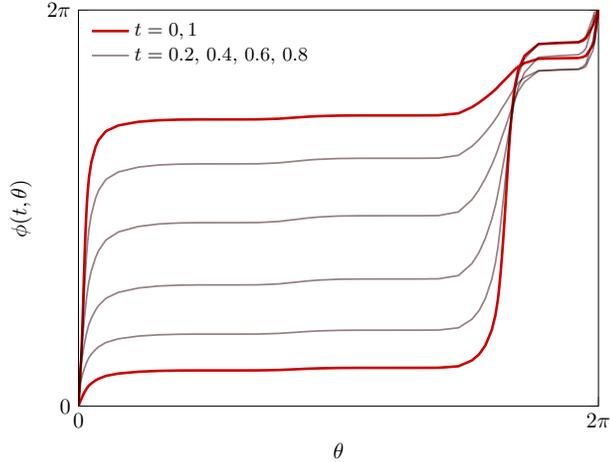


Figure 5. Left: computed path energies between welding maps in the same equivalence class shown in Figure 1. The near-zero value of the energies verifies that the computed paths are accurate geodesics. Right: evolution snapshots of $\phi(t, \theta)$ at $t = 0, 0.2, 0.4, \dots, 1$ where $\phi(0, \cdot) = \phi_3$ and $\phi(1, \cdot) = \phi_4$.

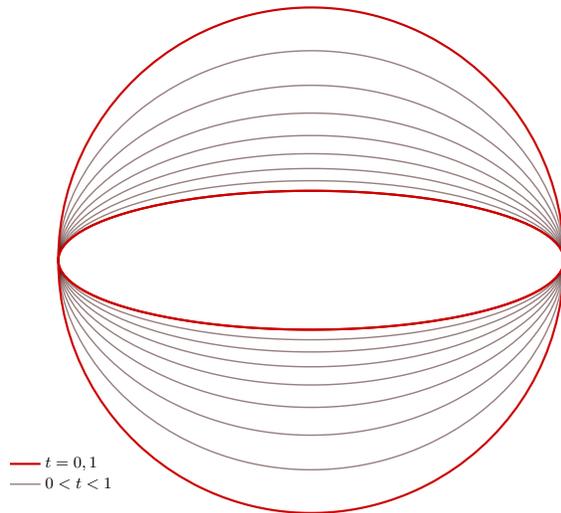
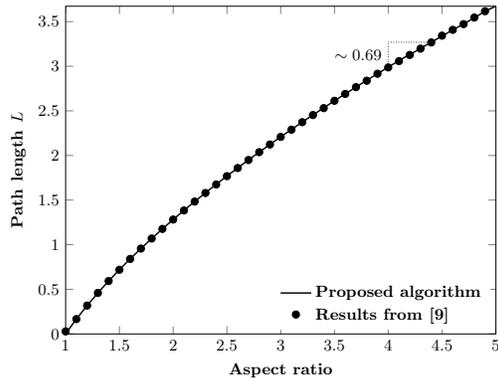


Figure 6. Left: computed geodesic energy for evolving an ellipse of given aspect ratio to a circle using the minimization algorithm (solid line) and using the shooting procedure from [9] (dashed line). (The lines overlap.) Right: evolution of the ellipse shown at equidistant points of the path parameter s .

4.2. Path length vs aspect ratio. Let ψ_t^r be a geodesic where ψ_0^r is the welding map for an ellipse of aspect ratio r , and ψ_1^r is the identity. In Figure 6 we show results of simulations for path lengths $L[\psi^r]$ versus aspect ratio r ranging from 1 to 5. The results suggest that the asymptotic relation is approximately linear, and these results match those obtained in [9] very well.

4.3. Hyperbolicity test. We verify the negative curvature of Teichmüller space $PSL_2(\mathbb{R}) \backslash \mathbf{Diff}(S^1)$. Consider an ellipse of a fixed aspect ratio corresponding to the weld ϕ_1 . For rotations of the ellipse by angles $\pm 2\pi/3$ we generate two more welding maps ϕ_2 and ϕ_3 . We consider

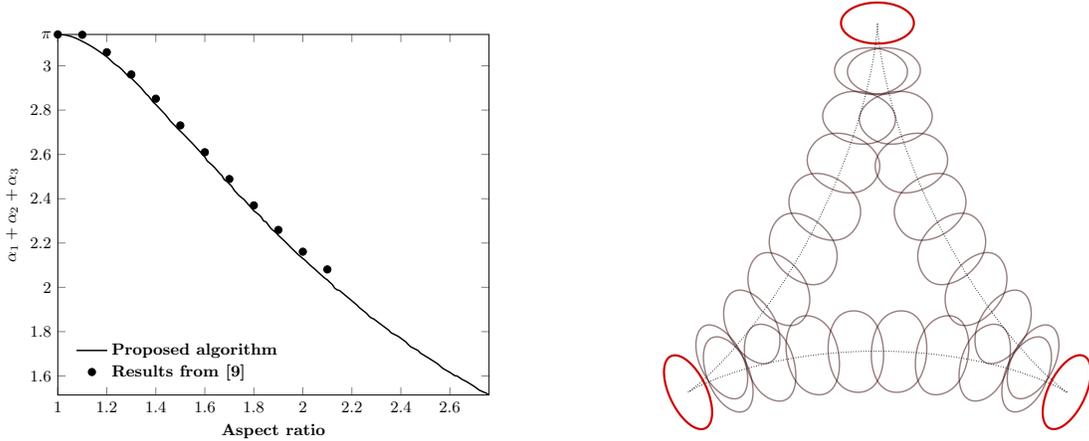


Figure 7. Left: Angle sum versus aspect ratio using the minimization algorithm and the results from [9]. Right: evolution along geodesics for a triplet of rotated ellipses. Each ellipse-like shape is the true shape along the geodesic, but the translational placement along a hyperbolic triangle is a cartoon-like representation of the flow on $T(1)$, suggesting the negative curvature of the space.

these three points in $PSL_2(\mathbb{R}) \backslash \mathbf{Diff}(S^1)$ and form a triangle using ϕ_i as the vertices and corresponding geodesic paths on $PSL_2(\mathbb{R}) \backslash \mathbf{Diff}(S^1)$ as the vertex connections. Let the $t = 0$ velocity field that pushes ϕ_i to ϕ_j at $t = 1$ be denoted $v_{i,j}$. Then the angle at vertex ϕ_i is given by

$$\alpha_i = \frac{\langle v_{i,i\oplus 1}, v_{i,i\ominus 1} \rangle_{WP}}{\|v_{i,i\oplus 1}\|_{WP} \|v_{i,i\ominus 1}\|_{WP}},$$

where \oplus and \ominus denote modular addition and subtraction, respectively, on the set $\{1, 2, 3\}$. Because all sectional curvatures of the WP metric are negative, we expect that $\sum_{i=1}^3 \alpha_i \leq \pi$. We verify this fact in Figure 7. We again compare the results against those computed in [9] and obtain similar results.

4.4. Shapes with corners. Non-smooth shapes do not result in diffeomorphic welds, and so the distance from any element of $T(1)$ to these non-smooth welds in the WP metric is infinite. Our algorithm cannot compute a geodesic of infinite length, so we expect to run into limitations in the algorithm as we attempt to compute geodesics to shapes with sharp corners. We generate a triangular shape with rounded corners in the plane, and parameterize the smoothness of the corner by $\alpha \geq 1$. As shown on the right-hand side of Figure 8, $\alpha = 1$ corresponds to a sharp corner (a triangle in Euclidean space), and increasing α rounds the corners of the triangle. For many values of α , we compute the geodesic from this rounded triangle to the identity and compile how the path length depends on α . The results are shown on the left-hand side of Figure 8.

As we decrease α down to the critical value of 1, we do observe a sharp growth in the path length. When α gets close to $1 + 10^{-3}$, our algorithm begins to reach its limit: at termination of the algorithm, the relative variation of the WP norm along the path increases to about 0.1%. Results from computations with $\alpha - 1$ smaller than 10^{-3} no longer appear to be geodesics. (I.e., the WP norm variation along the path becomes larger.) While this

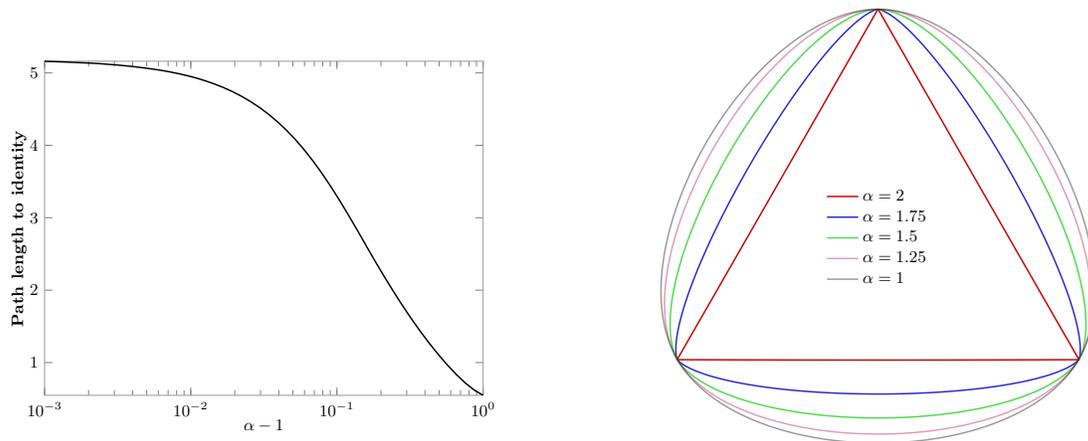


Figure 8. Left: WP distance from a circle (the identity weld) to a triangle with smoothness parameter α . Right: triangular shapes corresponding to values of the α parameter.

illustrates limitations of the algorithm, we are able to verify the expected increase in path length as $\alpha \downarrow 1$.

4.5. Shapes from the MPEG-7 dataset. We conclude our investigation with computation of geodesics from shapes in the MPEG-7 CE-shape-1 collection of planar shapes[1]. Figures 9, 10, and 11 show computed geodesics, and list the path length in each case. Also displayed are snapshots of the welding maps $\phi(t, \theta)$ at equidistant points in time.

We note that many shapes in this database have welding maps whose derivatives are singular to machine precision, or vanish to machine precision.³

When this happens, any algorithm implemented in finite precision for computing geodesics from this welding map will fail. We anticipate that any future work aimed at fixing the problem will first need to address the difficulty in computing conformal welds for these problematic shapes.

5. CONCLUSION

In this paper we have proposed and implemented a numerical solver for computing geodesics on the universal Teichmüller space $T(1)$ in the WP metric. Given a planar shape, a conformal weld in the coset space $PSL_2(\mathbb{R}) \backslash \mathbf{Diff}(S^1)$ uniquely identifies this shape. We prescribe two given welding maps as the endpoints of the path in $PSL_2(\mathbb{R}) \backslash \mathbf{Diff}(S^1)$, and minimize the WP energy of the path, subject to the endpoint constraints. This provides a constructive way to metrize the space of planar shapes, modulo rigid translation and scaling.

We demonstrated the applicability of our method by solving for numerous geodesics; our results compare favorably with the results obtained by a shooting method in [9]. The main difficulty with our procedure stems from the inability to handle welds whose composite conformal maps contain large derivatives; this is in fact a limitation for any procedure of which we are aware. Our future work will be directed at addressing this problem.

³In this case the composite conformal maps of the weld are ‘crowded’; the max-min ratio of the derivative magnitude is large. A robust strategy for finite-precision computation in this situation is an open problem.

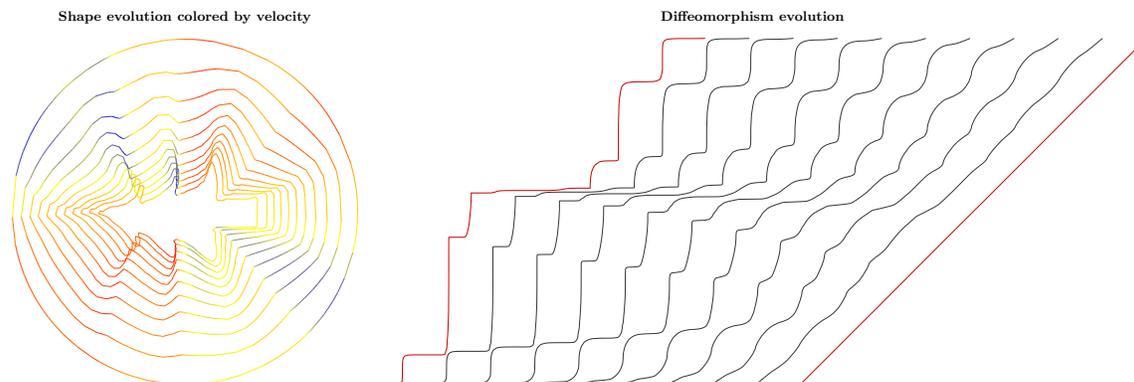


Figure 9. Left: shape evolution for MPEG-7 shape 1333, a rotated tree. Right: Evolution of the fingerprint along the path. The path length on $T(1)$ is 10.25.

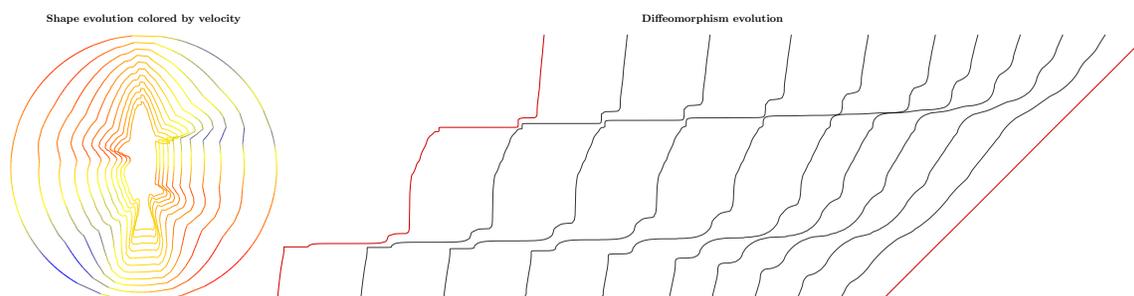


Figure 10. Left: shape evolution for MPEG-7 shape 785, a fish. Right: Evolution of the fingerprint along the path. The path length on $T(1)$ is 8.889.

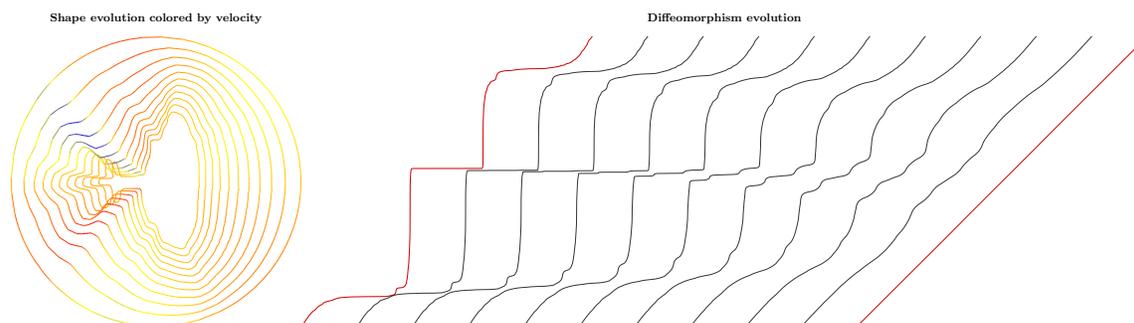


Figure 11. Left: shape evolution for MPEG-7 shape 875, a rotated miter. Right: Evolution of the fingerprint along the path. The path length on $T(1)$ is 7.486.

Our optimization algorithm can, in effect, be used to minimize energy in any Riemannian metric, and our approach for satisfying endpoint constraints on manifolds can likewise be utilized in other applications.

REFERENCES

1. *Shape data for the MPEG-7 core experiment CE-Shape-1*, <http://www.cis.temple.edu/~latecki/TestData/mpeg7shapeB.tar.gz>.

2. Lars V. Ahlfors, *Lectures on quasiconformal mappings*, 2 ed., American Mathematical Society, August 2006.
3. Shun-ichi Amari, *Natural Gradient Works Efficiently in Learning*, *Neural Computation* **10** (1998), no. 2, 251–276.
4. M. J. Bowick and S. G. Rajeev, *String theory as the Kähler geometry of loop space*, *Physical Review Letters* **58** (1987), no. 11, 1158.
5. Tobin A. Driscoll and Lloyd N. Trefethen, *Schwarz-Christoffel mapping*, 1 ed., Cambridge University Press, June 2002.
6. Lars Eldén, *A weighted pseudoinverse, generalized singular values, and constrained least squares problems*, *BIT* **22** (1982), no. 4, 487–502.
7. Francois Gay-Balmaz, Jerrold E. Marsden, and Tudor S. Ratiu, *The geometry of the universal Teichmüller space and the Euler-Weil-Petersson equations*, Tech. report, 2009.
8. John H Hubbard, *Teichmüller theory and applications to geometry, topology, and dynamics*, Matrix Editions, Ithaca, NY, 2006 (English).
9. S. Kushnarev and A. Narayan, *Approximating the Weil–Petersson Metric Geodesics on the Universal Teichmüller Space by Singular Solutions*, *SIAM Journal on Imaging Sciences* **7** (2014), no. 2, 900–923.
10. Sergey Kushnarev, *Teichons: Solitonlike geodesics on universal teichmüller space*, *Experimental Mathematics* **18** (2009), no. 3, 325–336.
11. Bao Cheng Li and Stavros Syngellakis, *Numerical conformal mapping based on the generalised conjugation operator*, *Mathematics of Computation* **67** (1998), no. 222, 619–639, ArticleType: research-article / Full publication date: Apr., 1998 / Copyright 1998 American Mathematical Society.
12. Charles F. Van Loan, *Generalizing the singular value decomposition*, *SIAM Journal on Numerical Analysis* **13** (1976), no. 1, 76–83.
13. Donald E. Marshall and Steffen Rohde, *Convergence of a variant of the zipper algorithm for conformal mapping*, *SIAM Journal on Numerical Analysis* **45** (2007), no. 6, 2577.
14. David Mumford, *Pattern theory: the mathematics of perception*, in ICM, Higher Education Press, 2002, p. 2002.
15. Subhashis Nag and Alberto Verjovsky, *Diff(S^1) and the Teichmüller spaces*, *Communications in Mathematical Physics (1965-1997)* **130** (1990), no. 1, 123–138.
16. Graeme Segal, *The definition of conformal field theory*, *Topology, geometry and quantum field theory*, London Math. Soc. Lecture Note Ser., vol. 308, Cambridge Univ. Press, Cambridge, 2004, p. 421577.
17. E. Sharon and D. Mumford, *2D-Shape analysis using conformal mapping*, *International Journal of Computer Vision* **70** (2006), no. 1, 55–75.
18. George T. Symm, *An integral equation method in conformal mapping*, *Numerische Mathematik* **9** (1966), no. 3, 250–258.
19. ———, *Numerical mapping of exterior domains*, *Numerische Mathematik* **10** (1967), no. 5, 437–445.
20. Leon A. Takhtajan Teo and Lee-Peng, *Weil-Petersson metric on the universal Teichmüller space*, American Mathematical Society, August 2006.
21. Remco C. Veltkamp, *Shape matching: Similarity measures and algorithms*, *Shape Modeling and Applications*, International Conference on (Los Alamitos, CA, USA), IEEE Computer Society, 2001, p. 0188.