DOI: 10.1111/cgf.13255

Eurographics Symposium on Geometry Processing 2017
J. A. Bærentzen and K. Hildebrandt
(Guest Editors)

Volume 36 (2017), Number 5

Fast Planar Harmonic Deformations with Alternating Tangential Projections

Eden Fedida Hefetz Edward Chien Ofir Weber

Faculty of Engineering, Bar-Ilan University, Israel

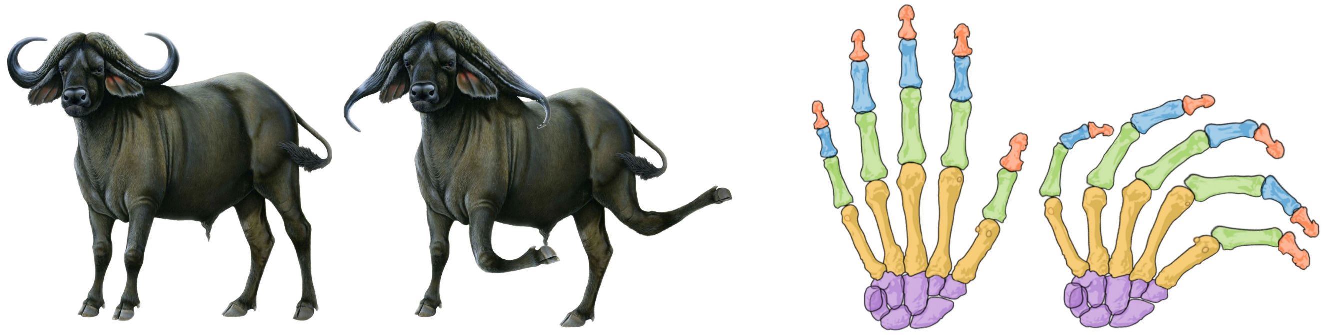


Figure 1: *Example deformations with our alternating tangential projections (ATP) method. The source domains are shown on the left, while the deformations are shown on the right. These deformations were easily constructed via real-time cage-based interaction and user-controlled distortion bounds. The distortion bounds for these images were* $C_k = 0.4$*,* $C_a = 2$*, and* $C_b = 0.5$*.*

Abstract

We present a planar harmonic cage-based deformation method with local injectivity and bounded distortion guarantees, that is significantly faster than state-of-the-art methods with similar guarantees, and allows for real-time interaction. With a convex proxy for a near-convex characterization of the bounded distortion harmonic mapping space from [LW16], we utilize a modified alternating projection method (referred to as ATP) to project to this proxy. ATP draws inspiration from [KABL15] and restricts every other projection to lie in a tangential hyperplane. In contrast to [KABL15], our convex setting allows us to show that ATP is provably convergent (and is locally injective). Compared to the standard alternating projection method, it demonstrates superior convergence in fewer iterations, and it is also embarrassingly parallel, allowing for straightforward GPU implementation. Both of these factors combine to result in unprecedented speed. The convergence proof generalizes to arbitrary pairs of intersecting convex sets, suggesting potential use in other applications. Additional theoretical results sharpen the near-convex characterization that we use and demonstrate that it is homeomorphic to the bounded distortion harmonic mapping space (instead of merely being bijective).

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Geometric algorithms, languages, and systems; Hierarchy and geometric transformations I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation G.1.6 [Numerical Analysis]: Optimization—Convex programming

1. Introduction

The problem of shape deformation and mapping is fundamental in computer graphics, and often one requires that such deformations are locally injective and of low distortion. These guarantees are difficult to achieve and there have been many works in recent years that have aimed to produce deformations or mappings with these guarantees [Lip12, WMZ12, AL13, APL14, PL14, LZ14, KABL14, CW15, KABL15, CLW16, FL16]. Only one recent work [LW16] (and the closely related [CCW16]) has achieved a convex characterization of this problem, for the subspace of harmonic planar mappings, producing useful bijections between the bounded distortion mapping space ($\mathcal{BD}$) and convex or near-convex spaces. None of the aforementioned methods is fast enough for real-time interaction. In this work, we solve this speed problem, producing a cage-based deformation method which produces locally injective bounded distortion harmonic mappings with framerates of 35-45 fps on standard problem sizes.

For this, we utilize the $\mathcal{L}_\nu$ space from [LW16], which characterizes such a mapping with the logarithm of its first Wirtinger derivative $l = \log f_z$, and the second complex dilatation $\nu = \overline{f_{\bar{z}}}/f_z$. To project to this near-convex space, [LW16] convexified and discretized this space, and performed a convex optimization problem which is effective, but too slow for interactive deformation. A key insight in our paper is that this problem may be solved approximately with a local-global strategy [SA07, LZX*08], implemented as an alternating projection method. In the local step, the quantities l and ν are projected to the bounded distortion constraint space, which has a product structure that allows for a fast parallelized projection, implemented with a GPU kernel. The global step, projects to a linear subspace of holomorphic l and ν, which corresponds to solution of a least squares problem, and is also parallelizable. Finally, inspired by [KABL15], we enhance this global step by restricting it to be orthogonal to the previous local step, making it a tangential projection (when the local step reaches a smooth point). We provide a theoretical proof that this algorithm (referred to as ATP) will converge to a bounded distortion harmonic mapping. Experimentally, on standard problem sizes, it results in speedups of 4-5x and 200-400x over the standard alternating projection method and direct convex optimization, respectively, and approximates the geometric projection well (see Fig. 9).

In addition to the contributions of a parallelized, improved, and provably convergent alternating projection method, we note that our cage-based deformation method naturally extracts l and ν values from the cage data, and better reflects user intent than projection-based methods (in addition to being faster). These methods take in a computed mapping that is not bounded distortion and attempt to project it to a bounded distortion space (something our framework is also capable of), but our method circumvents the need for computation of a mapping and directly infers the suitable l and ν values from the user manipulation of the cage (see Fig. 2). In relation to other cage-based barycentric methods for deformation, our approach shares their efficiency due to our optimized methods, but additionally allows for direct distortion control, which these do not have.

Lastly, we have additional theoretical contributions: the proof of convergence for ATP extends beyond the domain of our specific problem to arbitrary pairs of intersecting closed convex sets (Appendix A), and we strengthen the bijection between $\mathcal{BD}$ and $\mathcal{L}_\nu$ found in [LW16] by showing it is also a homeomorphism (Appendix B). The first is important as ATP is a general strategy that may have applications in other problems that take alternating projection solutions, and the second shows that the bijection used in this paper preserves nearness relationships, for which [LW16] provided ample experimental evidence of.

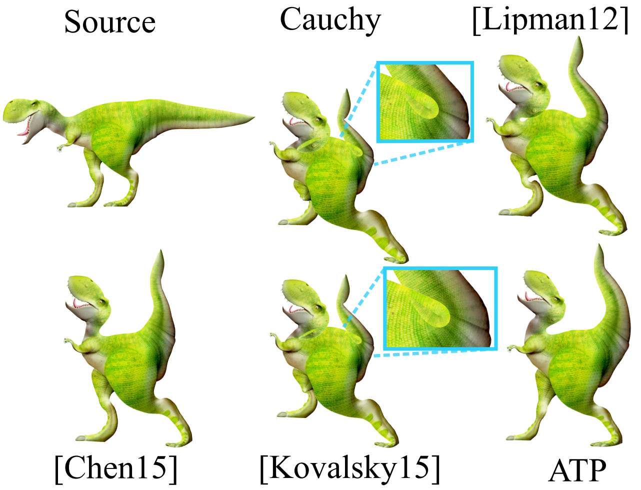


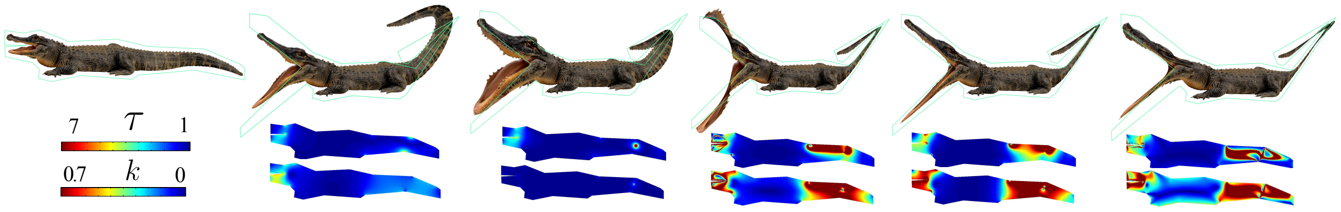
Figure 2: *Comparison to projection methods. We compare ATP to three projection methods: [Lip12], [CW15], and [KABL15]. The input mapping for the projection methods is labelled "Cauchy" and is a linear combination of Cauchy basis functions [WBCG09]. As can be seen, ATP avoids the poor input mapping, better detects user intention, and distorts the source domain to a lower degree. Note also that [KABL15] fails to correct the lack of local injectivity in the input mapping. The distortion bounds used with all methods were* $C_k = 0.4$, $C_a = 2$, *and* $C_b = 0.5$.

2. Previous Work

Barycentric coordinates provide a general tool for cage-based shape deformation. The computation of the coordinates is done in preprocessing making the evaluation of the mapping at runtime very efficient. This is achieved by a dense matrix-vector multiplication, where the matrix is fixed and the vector varies based on the position of the vertices of the so-called polygonal cage. This product is embarrassingly parallel and uses a highly regular memory access pattern which allows for efficient GPU implementation.

The linearity and simplicity of barycentric mappings has its price, as these methods lack control over the metric distortion induced by the mapping. Furthermore, a barycentric mapping between a source and a target polygon is, in general, not a bijection [Jac13]. The majority of barycentric methods possess the Lagrange property, hence they interpolate the target cage [HF06, JMDG07, LKCOL07, WBCGH11]. Methods that relax this requirement by allowing the image of the boundary under the mapping to *approximate* the target cage, typically enjoy preferable quality [LLCO08, WBCG09, WG10], but still lack direct control over distortion. Our method belongs to this class of cage-based deformation methods. It is competitive in terms of efficiency yet provides full control over distortion and injectivity. For a review of a broader class of linear deformation methods, we refer the reader to [BS08].

Conformal deformation methods [WG10, CPS11, VMW15] are locally injective by construction. They prohibit angular distortion completely but lack control over isometric distortion which is unsatisfactory. The larger linear space of harmonic mappings contains conformal mappings and also possesses many of the same useful



(a) Source (b) ATP (c) Cauchy (d) Mean value (e) Harmonic (f) Magic

Figure 3: *Comparison to other cage-based deformations. We compare ATP to several cage-based deformation methods using various kinds of barycentric coordinates: [WBCG09] in (c), [HF06] in (d), [JMDG07] in (e), and [WBCGH11] in (f). Our deformation is of low conformal and isometric distortion (measured with $\tau = \max(\sigma_a, 1/\sigma_b)$), and is qualitatively superior. Additionally, it is the only locally injective deformation. The distortion bounds used with ATP were $C_k = 0.5$, $C_a = 5$, and $C_b = 0.2$.*

properties, making it a popular choice. Harmonic methods have been in use for some time, for problems aiming to achieve local injectivity (or global bijectivity). Tutte [Tut63] presented a seminal method for embedding a planar graph into a convex subset of the plane, and [Flo97] generalized this method and utilized it for globally bijective mesh embedding. Several other works use these mappings as an intermediate for constructing bijections between nonconvex domains [WZ14, APL14] . Chen and Weber [CW15] present necessary and sufficient boundary conditions for a planar harmonic mapping to be locally injective and to have conformal-and-isometric bounded distortion. These conditions, however, are nonconvex and enforcing them is challenging. Levi and Weber [LW16] parametrized the space of locally injective and bounded distortion planar harmonic mappings, and show that it can be characterized via convex alternative spaces.

The bounded distortion method of Poranne and Lipman [PL14] supports general types of smooth planar mappings that can be expressed as linear combinations of radial basis functions. More generally, bounded distortion and/or locally injective methods that restrict the mapping to be continuous piecewise linear are very popular due to their flexibility. These are usually also suitable for handling curved surfaces. Such methods are nonsmooth by construction and are typically orders of magnitude slower than our method. Moreover, none of the existing methods is readily suitable for a GPU implementation. Approaches that are based on a barrier term [SKPSH13, SS15, FLG15, RPPSH17] require a valid initialization to operate successfully. Similarly, methods that convexify the nonconvex bounded distortion space [Lip12, KABL14], rely on proper initialization (frame choices) and often suffer from feasibility issues. Aigerman and Lipman [AL13] project a mapping to the bounded distortion space by *approximating* the bounded distortion space using a convex space in the vicinity of the initial mapping and iteratively adjust the approximation. Kovalsky et al. [KABL15] solve a similar projection problem more efficiently using alternating projections. [AL13, KABL15] are not guaranteed to converge and even upon convergence, they produce orientation preserving mappings which are not necessarily locally injective. Moreover, [KABL15] does not support control over isometric distortion. Our method (ATP) draws inspiration from [KABL15], yet in contrast, is guaranteed to converge to a locally injective mapping with user-specified conformal and isometric distortion bounds due to the convex nature of the space we operate within [LW16]. ATP outperforms the conic solver used in [LW16] by up to 3 orders of magnitude.

2.1. Alternating Projection Methods

In this small subsection, we briefly mention some of the relevant works on alternating projection methods. The study of such methods began with von Neumann [vN50], who was considering linear subspaces, and proved the convergence of alternating geometric projections onto the subspaces to their intersection. Moreover, he showed that the limit point was equal to the geometric projection to the intersection. We refer to this "method of alternating projections" by the acronym MAP. MAP was later extended to arbitrary closed convex sets, and potentially more than two such sets.

Our suggested modification, ATP, which adds tangential projections, is provably convergent for two such sets with nontrivial intersection, so we note that the corresponding proof for MAP is provided in [BB93]. Our proof is a variation of their argument. There is a large body of work around MAP exploring its correctness, convergence speed, and variants, so we refer to the excellent survey by the same authors: [BB96].

As already noted, we took inspiration for our modification from [KABL15] who apply the same method in a setting with nonconvex sets. There is an additional recent work [SS16] where the authors apply this method for the purposes of structured low-rank approximation. In their setting, one of the spaces is not convex, but a smooth variety, and they prove local quadratic convergence. For us, the boundary of one of our sets ($\mathfrak{B}$, see Section 4.2) is not smooth, so their proof does not apply in our case.

As noted in [KABL15], the tangential modification is inspired by Newton's method for root-finding, and by Gauss-Newton methods for nonlinear least squares problems. These ideas have appeared in other places in the graphics literature as well: in [BDS*12] which used alternating projections, and in [TSG*14] which used Gauss-Newton optimization methods. Lastly, we note that many works in graphics springing from [SA07, LZX*08] adopt related alternating energy minimization approaches.

3. Bounded Distortion Harmonic Mappings

In this section, we briefly review some basic definitions, and the construction of the $\mathcal{L}_\nu$ space from [LW16], to set notation. For a more detailed introduction to some of the preliminary material on bounded distortion harmonic mappings and the relevant complex analysis, we refer to Section 3 of [CCW16]. A more complete description of the $\mathcal{L}_\nu$ space is in [LW16].

A *harmonic function* $u : \Omega \to \mathbb{R}$ defined on a domain $\Omega \subseteq \mathbb{R}^2$ is one which satisfies the Laplace equation:

$$\Delta u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0$$

at each point in the domain. The value of a harmonic function is intuitively the average of its nearby values, and thus achieves extrema on its boundary (see the Mean Value Property and Maximum Principle [Dur04]).

A planar *harmonic mapping* is a mapping $f : \Omega \to \mathbb{R}^2$ for which both component functions u and v (where $f = (u, v)$) are harmonic. We may characterize local injectivity and quantify distortion of these mappings in terms of their first and second Wirtinger derivatives at each point, denoted f_z and $f_{\bar{z}}$. These are representative of the closest similarity and antisimilarity matrices to the Jacobian, J_f. In particular, we have the following formulae:

$$\det(J_f) = |f_z|^2 - |f_{\bar{z}}|^2 \tag{1}$$

$$\sigma_a = |f_z| + |f_{\bar{z}}|, \quad \sigma_b = \Big| |f_z| - |f_{\bar{z}}| \Big| \tag{2}$$

$$k = \frac{|f_{\bar{z}}|}{|f_z|}. \tag{3}$$

Equation (1) implies that local injectivity is equivalent to $|f_z| > |f_{\bar{z}}|$ (and that $f_z \neq 0$ throughout Ω for a locally-injective mapping). The quantities in equations (2) are the singular values of J_f and quantify the isometric distortion. They are equal to the lengths of the major and minor axes of the image ellipse $J_f(\mathbb{S}^1)$ and satisfy $0 < \sigma_b \leq \sigma_a < \infty$. In the setting of locally-injective maps, the second of these equations simplifies to $\sigma_b = |f_z| - |f_{\bar{z}}|$. The quantity in equation (3) denotes the conformal distortion and satisfies $0 \leq k < 1$ in the locally injective setting. In light of these formulae, we switch to complex notation from now on and consider $\Omega \subseteq \mathbb{C}$.

With these definitions, we may define the space of *bounded distortion harmonic mappings*, denoted $\mathcal{BD}$ (as done in [LW16]). For a triplet of distortion bounds (C_k, C_a, C_b) where $C_k \in [0, 1)$ and $0 < C_b \leq C_a < \infty$, consider the set of harmonic mappings satisfying the following:

$$\begin{aligned} k(z) &\leq C_k \quad \forall z \in \Omega \\ \sigma_a(z) &\leq C_a \quad \forall z \in \Omega \\ \sigma_b(z) &\geq C_b \quad \forall z \in \Omega. \end{aligned} \tag{4}$$

$\mathcal{BD}$ is this space and is non-empty for any suitable choice of (C_k, C_a, C_b), as any scaling map $z \mapsto sz$ for $C_b \leq s \leq C_a$ will be in the space. Finally, we note the following observation from [CW15]:

Observation 1 When Ω is simply-connected, an equivalent set of conditions to equations (4) is obtained if you enforce the distortion bounds only on $\partial\Omega$ and ask that $f_z \neq 0$ within Ω.

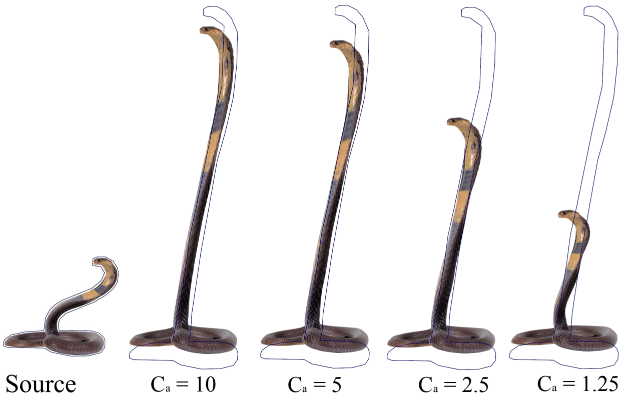


Figure 4: *Efficacy of distortion bound variation. Four results of our ATP deformation method are shown, with differing max isometric distortion bounds C_a, the values of which are denoted below each result. The other distortion bounds used with ATP were $C_k = 0.5$ and $C_b = 0.2$.*

3.1. The $\mathcal{L}_\nu$ Space

In [LW16], three different characterizations of $\mathcal{BD}$ were constructed. We use the $\mathcal{L}_\nu$ space as it was found to have performed the best qualitatively, of the three. To introduce it, we need a fact about harmonic mappings:

Observation 2 When Ω is simply-connected, we have a decomposition: $f = \Phi + \overline{\Psi}$, where Φ and Ψ are holomorphic functions. This decomposition is unique up to a complex constant added to one of Φ and $\overline{\Psi}$ and subtracted from the other.

Essentially, one obtains this decomposition by integrating the pointwise decomposition of the Jacobian into similarity and antisimilarity parts. A detailed proof is found in [Dur04]. Note that $f_z = \Phi_z$ and $f_{\bar{z}} = \overline{\Psi_z}$, so f may be recovered up to a constant by complex integration if f_z and $f_{\bar{z}}$ are known.

We may also further manipulate f_z and $f_{\bar{z}}$. In particular, we may take the logarithm of f_z, as it is never 0 for a locally injective map. For this, we pick a point $z_0 \in \Omega$ and ask that $\log(f_z(z_0))$ takes the value of the principal branch, and compute the rest of the logarithm by integration of f_z'/f_z (for details, see Section 3.5 of [CCW16]). We denote this quantity l for shorthand.

Similarly, we construct $\nu = \overline{f_{\bar{z}}}/f_z$, the *second complex dilatation*, whose modulus is k our conformal distortion measure. It is the only holomorphic function (up to global rotation) that shares the same modulus as the Beltrami coefficient. Note again that from l and ν, it is possible to reconstruct f up to a complex translation constant, as $f_z = e^l$ and $f_{\bar{z}} = \overline{\nu e^l}$.

Thus, for a harmonic mapping f, we can transform conditions (4) into conditions on l and ν. This leads us to the definition of the $\mathcal{L}_\nu$ space, a subspace of Hol × Hol, where Hol denotes the set of holomorphic functions on Ω. A pair of functions (l, ν) is in $\mathcal{L}_\nu$ if the following conditions hold for a valid choice of (C_k, C_a, C_b):

$$\begin{aligned} k(z) &= \quad |\nu(z)| && \leq C_k \quad \forall z \in \partial\Omega && (5)\\ \sigma_a(z) &= e^{\mathrm{Re}(l(z))}(1+|\nu(z)|) && \leq C_a \quad \forall z \in \partial\Omega && (6)\\ \sigma_b(z) &= e^{\mathrm{Re}(l(z))}(1-|\nu(z)|) && \geq C_b \quad \forall z \in \partial\Omega.. && (7) \end{aligned}$$

For a consistent choice of constants (C_k, C_a, C_b), and a fixing of integration constants, it is shown in [LW16] that $\mathcal{L}_\nu$ is in bijection with $\mathcal{BD}$. The argument makes precise the informal discussion above. In Appendix B, we strengthen this bijection and show that it is a homeomorphism, preserving the topological structure of these functional spaces.

4. Convexification & Discretization

In this section, we convexify and discretize the $\mathcal{L}_\nu$ space, allowing provably convergent alternating projection methods to be used.

4.1. Convexification

As noted in [LW16], conditions (5) and (7) are convex conditions, while (6) is not. This is most easily seen by viewing things in the $\mathrm{Re}(l)$ and $|\nu|$ plane, as done in Fig. 5. Note that conditions (6) and (7) may be rearranged as:

$$\mathrm{Re}(l) \leq \log(C_a) - \log(1+|\nu|) \qquad (8)$$

$$\mathrm{Re}(l) \geq \log(C_b) - \log(1-|\nu|). \qquad (9)$$

As done in [LW16], we convexify (8) by linearizing $\log(1+|\nu|)$ at $\nu = 0$. Additionally, we replace (9) by a constraint that is linear in $|\nu|$, and we are left with our *convexified* $\mathcal{L}_\nu$ *space*, denoted $\mathcal{L}_\nu^{conv}$:

$$|\nu| \leq C_k \qquad (10)$$

$$\mathrm{Re}(l) \leq \log(C_a) - |\nu| \qquad (11)$$

$$\mathrm{Re}(l) \geq \log(C_b) - \frac{\log(1-\xi)}{\xi}|\nu|. \qquad (12)$$

In the above equations, $\xi = \min\left(C_k, 1 - W\left(e\frac{C_b}{C_a}\right)\right)$ where W is the Lambert W function. The intuition behind this equation is that a linear bound is enforced which runs between $(0, \log(C_b))$ and the point (in the $\mathrm{Re}(l)$ and $|\nu|$ plane) where condition (9) meets either condition (10) or condition (11), whichever comes first. Again, we refer to Fig. 5 for illustration. The linear bound is enforced for ease of projection, as explained later in Section 6.2. Lastly, it is implicit above that all of these conditions are enforced along $\partial\Omega$.

Before moving on to discretization, note that unlike many other convexification approaches, this one has no parameters (e.g., frames) and the conditions keep the same form everywhere along $\partial\Omega$. Additionally, it is guaranteed to be nonempty, as for any $C_b \leq s \leq C_a$, the scaling map $z \mapsto sz$ satisfies the constraints (so feasibility is guaranteed).

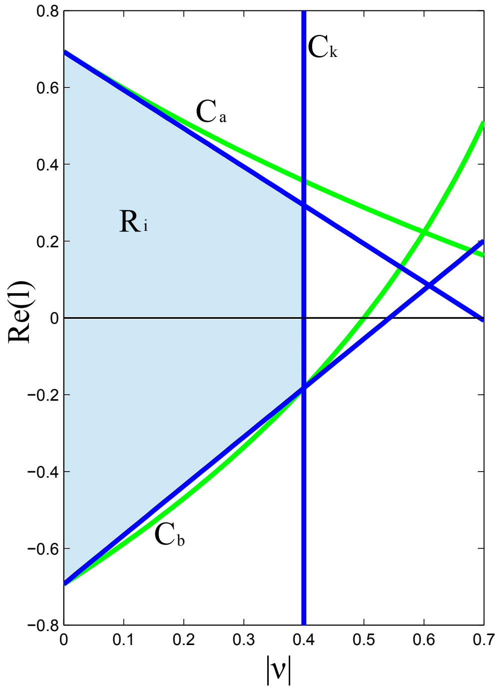


Figure 5: *A schematic illustrating both the* $\mathcal{L}_\nu$ *and* $\mathcal{L}_\nu^{conv}$ *spaces in the* $\mathrm{Re}(l)$ *and* $|\nu|$ *plane with distortion bounds* $C_k = 0.4$, $C_a = 2$, *and* $C_b = 0.5$. *The inequalities defining* $\mathcal{L}_\nu$: (5), (6), *and* (7), *are illustrated in green; while those defining* $\mathcal{L}_\nu^{conv}$: (10), (11), *and* (12), *are illustrated in blue. The distortion bounds label the corresponding inequalities (the conformal inequalities are the same). For discretization of* $\mathcal{L}_\nu^{conv}$, *one considers the bounded distortion space at a sample point* p_i, *a subset of* $\mathbb{C}^2$, *given by* l *and* ν *values satisfying inequalities* (10), (11), *and* (12). *To visualize this space, one rotates the shaded region* R_i *about the* $\mathrm{Re}(l)$ *axis (obtaining a 3-dimensional volume), and then products the result with* $\mathbb{R}$ *to account for the possible values of* $\mathrm{Im}(l)$.

4.2. Discretization

To discretize $\mathcal{L}_\nu^{conv}$, we enforce these distortion constraints at m densely sampled points $\{p_1, p_2, \ldots, p_m\} \in \partial\Omega$ and use n Cauchy basis functions from an offset cage to form an n-dimensional subspace of holomorphic functions from which l and ν will be taken:

$$l(z) = \sum_{j=1}^{n} s_j C_j(z) \quad \& \quad \nu(z) = \sum_{j=1}^{n} t_j C_j(z).$$

The $C_j(z)$ are the Cauchy basis functions [WBCG09] and the s_j and t_j are complex coefficients which specify l and ν. We assume that $m > n$, as would be expected for practical use.

With these let us define a linear transformation $T : \mathbb{C}^n \to \mathbb{C}^m$ that takes in complex Cauchy basis coefficients and returns the values of the resulting function at the sample points. The matrix entries are $(T)_{ij} = C_j(p_i)$.

Now, we may see that the resulting discretized $\mathcal{L}_\nu^{conv}$ space is a convex subspace of $\mathbb{C}^{2m}$ and is the intersection of two convex sets. The first is $\mathrm{im}(\tilde{T})$ where $\tilde{T} : \mathbb{C}^{2n} \to \mathbb{C}^{2m}$ is a linear map with matrix $\left(\begin{smallmatrix} T & 0 \\ 0 & T \end{smallmatrix}\right)$ and describes the possible sample point values for l and ν arising from choices of s_j and t_j.

The second is the product space arising from enforcement of the $\mathcal{L}_\nu^{conv}$ conditions (10), (11), and (12) at each of the sample points p_i. If we let $\mathfrak{B}_i$ denote the pointwise spaces, then we may denote this second space with $\mathfrak{B} = \Pi_{i=1}^m \mathfrak{B}_i$. From Fig. 5, we may visualize an individual $\mathfrak{B}_i$ by rotating the shaded region R_i about the $\mathrm{Re}(l)$ axis, and producting with $\mathbb{R}$ to represent the possible values for $\mathrm{Im}(l)$.

In this discretized setting, the problem of deformation (or projection) is that we are given input data: l and ν values from a cage or mapping specifying a point in $\mathbb{C}^{2m}$, and would like to project to the discretized $\mathcal{L}_\nu^{conv}$ space. One method for doing so, is to use standard convex optimization techniques. As we shall see, this method is effective, but quite slow compared to alternating projection methods for this problem.

5. Alternating Projections

In this section, we briefly describe the standard method of alternating projections (MAP), and our modified version: alternating tangential projections (ATP). This is done in a fairly general setting, as the basic idea may be applicable to other problems. A detailed proof of convergence is given for ATP.

5.1. Method Statements

Consider two closed convex sets $A, B \subseteq \mathbb{R}^d$ with $A \cap B \neq \emptyset$. The method of alternating projections (MAP) takes an arbitrary point $p \in \mathbb{R}^d$ and returns a nearby point in $A \cap B$, and does this by alternating projections onto A and B.

To be more precise, for a closed convex set $C \in \mathbb{R}^d$ let $\mathrm{proj}_C : \mathbb{R}^d \to C$ denote the projection map which takes a point in $\mathbb{R}^d$ and maps it to the unique closest point in C. Letting $\langle , \rangle$ denote the standard inner product on $\mathbb{R}^d$, we have that $\mathrm{proj}_C(x)$ is the only point in C satisfying *Kolmogorov's criterion* (also see Fig. 6):

For a closed convex set $C \subseteq \mathbb{R}^d$ and an arbitrary point $x \in \mathbb{R}^d$,

$$\langle x - \mathrm{proj}_C(x), \mathrm{proj}_C(x) - c \rangle \geq 0, \quad \forall c \in C. \tag{K}$$

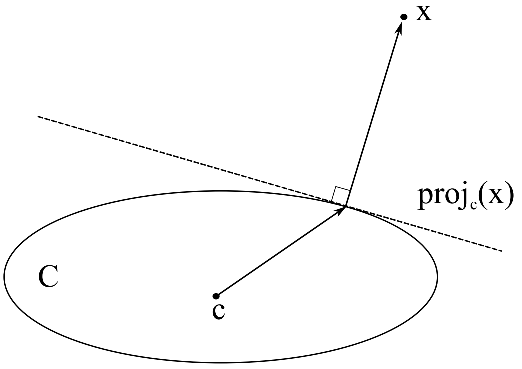


Figure 6: *A schematic illustrating Kolmogorov's criterion* (K).

MAP forms sequences of points $(a_1, a_2, a_3, \dots)$ in A and $(b_1, b_2, b_3, \dots)$ in B with the projection maps:

$$\begin{aligned} a_1 &= \mathrm{proj}_A(p) \\ b_i &= \mathrm{proj}_B(a_i), \quad i \geq 1 \\ a_{i+1} &= \mathrm{proj}_A(b_i), \quad i \geq 1 \end{aligned}$$

It is well-known that the sequences $(a_i)_{i=1}^\infty$ and $(b_i)_{i=1}^\infty$ will converge to a point $p^* \in A \cap B$ [BB93].

In our applied setting, we have that $A = \mathrm{im}\left(\tilde{T}\right)$ is a linear subspace and $B = \mathfrak{B}$ is a full-dimensional compact convex subset, so we assume that A and B are of this type. These conditions are not entirely necessary for convergence, but we maintain them here for pedagogical reasons. This setting is quite similar to that of [KABL15], and inspired by their approach, we propose a method which restricts the projection to A to the hyperplane containing b_i and with normal vector $a_i - b_i$. See Fig. 7 for a useful schematic. When B is smooth around b_i, this is the tangent plane to ∂B at b_i, thus we refer to it as the method of alternating tangential projections. It is inspired by Newton's method for root finding, and by Gauss-Newton approaches to non-linear least squares problems.

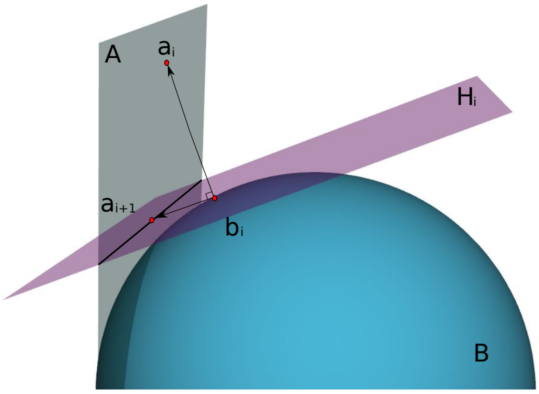


Figure 7: *A schematic illustrating the tangential projection step. The projection of b_i to A is restricted to H_i (with normal vector $a_i - b_i$ illustrated as well). Note that the resulting a_{i+1} is closer to $A \cap B$ than the result of unrestricted projection of b_i to A.*

To be precise, let us define $g_i(x) := \langle a_i - b_i, b_i - x \rangle$ and let H_i denote the hyperplane given by the zero set:

$$H_i = \{x \in \mathbb{R}^d : g_i(x) = \langle a_i - b_i, b_i - x \rangle = 0\}$$

ATP constructs sequences $(a_i)_{i=1}^\infty$ in A and $(b_i)_{i=1}^\infty$ in B according to the following equations:

$$\begin{aligned} a_1 &= \mathrm{proj}_A(p) \\ b_i &= \mathrm{proj}_B(a_i), \quad & i \geq 1 \\ a_{i+1} &= \mathrm{proj}_{A \cap H_i}(b_i), \quad & i \geq 1 \end{aligned}$$

Note that if we ever have $a_i = b_i$, then both are in $A \cap B$ and the process has converged to a limit point, so H_i will always be a hyperplane when the limit point has not been reached. We prove a small lemma to show feasibility of the method.

Lemma 3 If $A \cap B \neq \emptyset$, then for each tangential step of ATP:

$$A \cap H_i \neq \emptyset.$$

Proof Consider a point $q \in A \cap B$ and consider the line between q and a_i. Note that $g_i(a_i) = -\|a_i - b_i\| < 0$ and that $g_i(q) > 0$ by (K) (if $q = b_i$ a limit point has been reached). As g_i is continuous (in fact, affine), there must be a point r on the line between q and a_i for which $g_i(r) = 0$. As A is convex, $r \in A$ and as $g_i(r) = 0$, $r \in H_i$. □

Experimentally, we will see that the convergence of this method (ATP) is faster than that of the standard MAP method in Section 7. But first, we give a proof that ATP will converge to a limit point in $A \cap B$.

5.2. Convergence Proof

The proof follows the strategy of [BB93] in showing that the sequences are Fejér monotone with Kolmogorov's criterion (K). In order to be self-contained, we present an entire argument here without discussing Fejér monotonicity. The reader uninterested in this theoretical argument need only read Theorem 5 and may skip this section otherwise.

We first argue that for any point $q \in A \cap B$, $b_i = \mathrm{proj}_B(a_i)$ is closer to it than a_i.

$$\begin{aligned} \|a_i - q\|^2 &= \|a_i - b_i + b_i - q\|^2 \\ &= \|a_i - b_i\|^2 + 2\langle a_i - b_i, b_i - q\rangle + \|b_i - q\|^2 \\ &\geq \|a_i - b_i\|^2 + \|b_i - q\|^2 \end{aligned} \tag{13}$$

The final inequality follows from (K).

In the proof of convergence for MAP, an analogous argument would be used to show that a_{i+1} is closer to q than b_i, but the fact that a_{i+1} is restricted to H_i requires us to modify our argument.

Lemma 4 For each tangential step in ATP:

$$\|a_{i+1} - q\|^2 \leq \|b_i - q\|^2 - \|b_i - a_{i+1}\|^2.$$

Proof As with equation (13), we again have that:

$$\begin{aligned} \|b_i - q\|^2 &= \|b_i - a_{i+1} + a_{i+1} - q\|^2 \\ &= \|b_i - a_{i+1}\|^2 + 2\langle b_i - a_{i+1}, a_{i+1} - q\rangle + \|a_{i+1} - q\|^2 \end{aligned} \tag{14}$$

We again aim to show that $\langle b_i - a_{i+1}, a_{i+1} - q\rangle \geq 0$. For this, let us define $\tilde{g}_i(x) := \langle b_i - a_{i+1}, a_{i+1} - x\rangle$, and argue by contradiction: suppose that $\tilde{g}_i(q) < 0$. For intuition, the reader is referred to Figure 8.

As in the argument for Lemma 3, we may consider the line between q and a_i and the value of g_i along it, to find a point $r \in A \cap H_i$. Similarly, we may consider the value of $\tilde{g}_i$ along this line, and note the following:

$$\begin{aligned} \tilde{g}_i(a_i) &= \langle b_i - a_{i+1}, a_{i+1} - a_i\rangle + 0 \\ &= \langle b_i - a_{i+1}, a_{i+1} - a_i\rangle + \langle b_i - a_{i+1}, a_i - b_i\rangle \\ &= \langle b_i - a_{i+1}, a_{i+1} - b_i\rangle \\ &= -\|b_i - a_{i+1}\|^2 < 0 \end{aligned} \tag{15}$$

As $\tilde{g}_i(q) < 0$ and $\tilde{g}$ is affine, we have that $\tilde{g}_i(r) < 0$. This violates Kolmogorov's criterion (K) when applied to $H_i \cap A$ as a convex subset of H_i. □

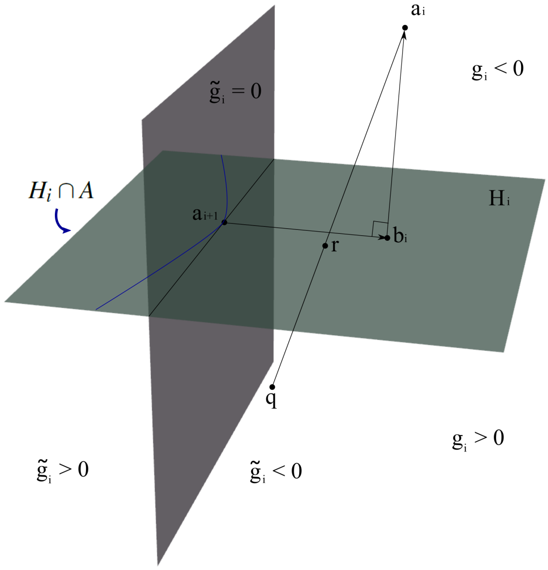


Figure 8: *A guide to the proof of Lemma 4. This schematic depicts an impossible scenario, with $\tilde{g}_i(q) < 0$. Note that A and B are not illustrated above, but $H_i \cap A$ is (its boundary in H_i is in blue). The planes defined by $g_i = 0$ (H_i) and $\tilde{g}_i = 0$ are orthogonal with normal vectors $a_i - b_i$ and $b_i - a_{i+1}$, respectively. As argued in the proof, we see that a line from q to a_i will intersect H_i at some point r for which Kolomogorov's criterion (K) restricted to H_i is violated.*

With Lemma 4 and equation (13), we may easily prove the following:

Theorem 5 If $A, B \subseteq \mathbb{R}^d$ are a linear subspace and a full-dimensional compact convex subset, respectively, with $A \cap B \neq 0$, then the sequences $(a_i)_{i=1}^{\infty}$ and $(b_i)_{i=1}^{\infty}$ constructed with ATP converge to a limit point $p^* \in A \cap B$.

Proof Note first that Lemma 4 and equation (13) imply that the sequence $(\|a_i - q\|, \|b_i - q\|, \|a_{i+1} - q\|, \ldots)$ converges (for any $q \in A \cap B$), as it is decreasing and bounded below by 0. Furthermore, this implies that:

$$a_i - b_i \to 0 \quad \& \quad b_i - a_{i+1} \to 0.$$

The remaining argument is a basic analytical one. As each step of ATP brings us closer to any $q \in A \cap B$, the sequence $(a_i)_{i=1}^{\infty}$ lies in a compact set, and has a limit point. Let $a_{i_k} \to a^*$ denote a subsequence converging to this limit point, which must lie in A by closedness. Closedness of B also allows us to easily see that $a^* \in B$ as $d(a_{i_k}, B) \leq \|a_{i_k} - b_{i_k}\| \to 0$ and:

$$d(a^*, B) \leq \|a^* - a_{i_k}\| + d(a_{i_k}, B) \quad \forall i_k.$$

Thus, if we let $p^* = a^*$, we have our result. Lemma 4 and equation (13) imply that $\|a_i - p^*\|$ is a decreasing sequence, and the existence of a subsequence converging to 0, implies that the overall sequence does as well. We also have that $b_i \to p^*$ as $\|b_i - p^*\| \leq \|b_i - a_{i+1}\| + \|a_{i+1} - p^*\| \to 0$. ☐

Before discussing the algorithms and implementation details, we note that the restrictions on A and B for Theorem 5 are unnecessary, and the arguments above merely use their convexity and closedness. The ideas may also be used to prove convergence if you ask for movement in an orthogonal direction for every step (instead of just every other one). A brief discussion of this is in Appendix A.

Lastly, we note that only in the case where A and B are affine subspaces, is it known that the standard MAP converges to $\mathrm{proj}_{A\cap B}(p)$ [vN50]. In practice, both MAP and ATP converge to points very close to $\mathrm{proj}_{A\cap B}(p)$ (see Fig. 9).

6. Our Algorithms

With the description of MAP and ATP, we may describe the algorithms we have designed. These take in a point $p = (p_l, p_\nu) \in \mathbb{C}^{2m}$ as input, representing l and ν data at each of the m sample points in $\partial\Omega$. Then MAP or ATP is applied to p, successively projecting it onto $\mathrm{im}\left(\tilde{T}\right)$ and $\mathfrak{B}$. The resulting point $p^* = \tilde{T}(s_1, \dots, s_n, t_1, \dots, t_n)$ for some Cauchy basis coefficients s_j and t_j. Finally, $l = \sum_j s_j C_j$ and $\nu = \sum_j t_j C_j$ are used to reconstruct a locally injective bounded distortion harmonic mapping via the appropriate algebraic manipulations and numerical integration over Ω. The steps above are described in greater detail in the subsections below.

As noted in Section 2.1, we may view our basic setting and use of alternating projection methods as a local-global method [SA07, LZX*08]. Projection to $\mathfrak{B}$ may be viewed as a local step, ensuring local distortion bounds are satisfied, while projection to $\mathrm{im}\left(\tilde{T}\right)$ may be viewed as a global step, ensuring that the l and ν values at each step come from globally defined holomorphic functions. For application of both MAP and ATP, we will see that both local and global steps may be easily parallelized, resulting in the speed which allows our method to be real-time. We denote the full algorithms for shape deformation with the labels MAP-gpu and ATP, with the change in the first label made to distinguish it from the method of alternating projections, and to emphasize the parallelized nature of the algorithm (shared by both methods).

6.1. Gathering Input Data

For our deformation methods, the user specifies a polygonal cage about the figure, which serves as our domain Ω. A slightly offset cage is utilized for user input, with handles between the polygonal segments. In order to enrich the space, the offset cage is upsampled uniformly to provide n Cauchy basis functions. Additionally, m sample points are uniformly chosen on $\partial\Omega$, where our data will come from. Each sample point is associated to one of the original polygonal segments. The number of sample points should exceed the number of basis functions to ensure proper enforcement of the distortion bounds. Experimentally, a ratio of $m = 10n$ was found to be sufficient under all practical circumstances.

To infer the input l and ν values from the target cage, we consider for each sample point the unique affine transformation that takes its source cage segment vector e_j to its target cage segment vector $\hat{e}_j$, and maps the outward unit normals $n_j \mapsto \hat{n}_j$ to each other. The linear part of this transformation may be considered as the composition of three linear transformations: $P_2 \circ S \circ P_1$, defined below utilizing complex notation (where a complex number $z = x + \mathrm{i}y$ denotes a vector $(x, y) \in \mathbb{R}^2$).

$$P_1(z) = \rho_1 z, \qquad \rho_1 = \frac{|e_j|}{e_j}\mathrm{i}$$

$$S(z) = s_z z + s_{\bar{z}}\bar{z}, \qquad s_z = \frac{|e_j| + |\hat{e}_j|}{2|e_j|} \quad \& \quad s_{\bar{z}} = \frac{|e_j| - |\hat{e}_j|}{2|e_j|}$$

$$P_2(z) = \rho_2 z, \qquad \rho_2 = -\frac{|\hat{e}_j|}{\hat{e}_j}\mathrm{i}$$

P_1 is an orthogonal transformation that maps n_j and e_j into the positive x- and y-axes, respectively. S performs a scaling in the y-direction by $|\hat{e}_j| / |e_j|$. Lastly, P_2 is another orthogonal transformation which rotates $S \circ P_1(n_j)$ and $S \circ P_2(e_j)$, into $\hat{n}_j$ and $\hat{e}_j$, respectively. The similarity and antisimilarity parts of the unique affine transformation may now be easily derived:

$$f_z = \rho_2 s_z \rho_1 = \frac{1}{2}\left(\frac{|\hat{e}_j| + |e_j|}{|\hat{e}_j|}\right)\frac{\hat{e}_j}{e_j} \tag{16}$$

$$f_{\bar{z}} = \rho_2 s_{\bar{z}} \overline{\rho_1} = \frac{1}{2}\left(\frac{|\hat{e}_j| - |e_j|}{|\hat{e}_j|}\right)\frac{\hat{e}_j}{\overline{e_j}}$$

With these values, l is taken to be a logarithm of f_z and ν is taken to be $\overline{f_{\bar{z}}}/f_z$. To choose the logarithm of f_z for sample points on segment e_{j+1}, denoted l_{j+1}, we base it upon the choice for sample points on segment e_j (likewise denoted l_j). This avoids inconsistent assignments due to poor branch choice. The formula $l_{j+1} = l_j + d_{j+1}$ is used, where $\mathrm{Im}(d_j)$ gives the corner angle difference for the source and target cages:

$$d_{j+1} = \mathrm{Log}\left(\frac{e_{j+1} f_z^{j+1}}{e_j f_z^j}\right) - \mathrm{Log}\left(\frac{e_{j+1}}{e_j}\right).$$

Above, Log denotes the principal branch, and f_z^j and f_z^{j+1} denote quantity (16) for the sample points on e_j and e_{j+1}, respectively. The method presented here generalizes some of those given in [Web17], which we refer to for additional discussion.

Lastly, we note that we may also use any existing cage-based barycentric method to provide such input data, by taking the resulting mapping and using it to calculate the l and ν values. However, it is more efficient to take data directly from the cage manipulation, which also more accurately reflect the user's intent. Evidence of this may be seen in Fig. 3.

6.2. Local projections

For projection of a point $p \in \mathbb{C}^{2m}$ to $\mathfrak{B} = \Pi_i \mathfrak{B}_i$, we may use the product structure to parallelize the projection. To be more specific, let l_i and ν_i denote the l and ν values at sample point i, so that $p = (l_1, \dots, l_m, \nu_1, \dots, \nu_m)$. In addition, let $(l_i^b, \nu_i^b) = \mathrm{proj}_{\mathfrak{B}_i}((l_i, \nu_i))$. Then we have:

$$\mathrm{proj}_{\mathfrak{B}}(p) = (l_1^b, \dots, l_m^b, \nu_1^b, \dots, \nu_m^b).$$

This follows from the fact that the standard inner product on $\mathbb{C}^{2m}$ is the sum of the componentwise standard inner products on the

(a) Source (b) MOSEK (c) MAP-gpu (d) ATP

Figure 9: *Near-optimality of projection by alternating projection methods. Deformations are computed after projection to the discretized $\mathcal{L}_{\nu}^{conv}$ space by three different methods. Column (a) illustrates the source domain. Column (b) illustrates the result of a convex optimization with Mosek [ApS15], column (c) illustrates the result of our parallelized MAP-gpu method, and column (d) illustrates the result of ATP. Column (b) represents the ground truth geometric projection, and the results of MAP-gpu and ATP are not numerically identical, but are visually indistinguishable.*

$\mathbb{C}^2$ factors. A dedicated GPU kernel performs these parallelized projections.

To calculate $(l_i^b, \nu_i^b) = \mathrm{proj}_{\mathfrak{B}_i}((l_i, \nu_i))$, we may also exploit some special structure of $\mathfrak{B}_i$. In particular, constraints (10), (11), and (12) cut out a subset of $\mathbb{C}^2$ which is a product of a rotationally symmetric convex subset of $\mathbb{R}^3$, spanned by the $\mathrm{Re}(l_i)$, $\mathrm{Re}(\nu_i)$, and $\mathrm{Im}(\nu_i)$ axes; and $\mathbb{R}$, spanned by the remaining $\mathrm{Im}(l_i)$ axis. Thus, we only need to do the projection in $\mathbb{R}^3$, and maintain the value of $\mathrm{Im}(\nu_i)$.

For the projection in $\mathbb{R}^3$, we are projecting to a set which is the rotation of R_i (see again Fig. 5) about the $\mathrm{Re}(l_i)$ axis. Thus, if we let $\left(\mathrm{proj}_{R_i}(\mathrm{Re}(l_i), |\nu_i|)\right)_x$ and $\left(\mathrm{proj}_{R_i}(\mathrm{Re}(l_i), |\nu_i|)\right)_y$ denote the components of this projection in the $(|\nu_i|, \mathrm{Re}(l_i))$-plane, we get the following formula for projection to $\mathfrak{B}_i$:

$$\mathrm{proj}_{\mathfrak{B}_i}\left(\mathrm{Re}(l_i), \mathrm{Im}(l_i), |\nu_i| e^{i\arg(\nu_i)}\right)$$
$$= \left(\left(\mathrm{proj}_{R_i}(\mathrm{Re}(l_i), |\nu_i|)\right)_y, \mathrm{Im}(l_i), \left(\mathrm{proj}_{R_i}(\mathrm{Re}(l_i), |\nu_i|)\right)_x e^{i\arg(\nu_i)}\right).$$

Lastly, we note that $\mathrm{proj}_{R_i}(l_i, \nu_i)$ is a simple projection onto a convex polytope in two dimensions and is easily accomplished. Constraint (11) is utilized for this purpose, as the original constraint (6) (still convex) does not allow for a simple closed-form solution for the projection.

6.3. Global projections

For the global projections, MAP-gpu merely solves a least-squares system. In particular, given a point $b_i \in \mathfrak{B}$, one is after the least-squares solution to $\tilde{T}x = b_i$ where $x = (s_1, \ldots, s_n, t_1, \ldots, t_n)$. The solution to this system is $\tilde{M}^{-1}\tilde{T}^t$, where we establish the notation $\tilde{M} := \tilde{T}^t\tilde{T}$ and is invertible as $\tilde{T}$ is invertible. Recalling the block nature of $\tilde{T}$ we get the following formula:

$$\tilde{M}^{-1}\tilde{T}^t = \begin{pmatrix} M^{-1}T^t & 0 \\ 0 & M^{-1}T^t \end{pmatrix} \tag{17}$$

where $M := T^tT$. This allows us to reduce the least-squares problem into two identical smaller least-squares problems, further speeding the method. If we let $b_i = (l_1^i, \ldots, l_m^i, \nu_1^i, \ldots, \nu_m^i)$, so that l^i and ν^i denote the components of b_i, then the solution to the least-squares problem, denoted x_{i+1}, takes the form:

$$x_{i+1} = \begin{pmatrix} M^{-1}T^tl^i \\ M^{-1}T^t\nu^i \end{pmatrix} \tag{18}$$

The final step is to set $a_{i+1} = \tilde{T}x_{i+1}$ (which may also be blocked easily, left to the reader).

For ATP, the global projections are more complicated due to the restriction to H_i. As in [KABL15], we use Lagrange multipliers and

(a)

MOSEK:

m \ n	50	100	200	400	800	1600
500	764	1570	4082	12275		
1000	1491	3165	8726	24320	93337	
2000	3246	6522	17517	49603	156659	681795
4000	6837	14230	39328	101737	345922	1293360
8000	15389	14120	79851	206840	723958	-
16000	37218	69373	174414	465316	-	-

(b)

MAP-gpu:

m \ n	50	100	200	400	800	1600
500	77 (x10)	59 (x27)	51 (x80)	61 (x201)		
1000	101 (x141)	87 (x36)	116 (x75)	101 (x241)	111 (x841)	
2000	188 (x17)	160 (x41)	188 (x93)	156 (x318)	185 (x847)	255 (x2674)
4000	317 (x22)	201 (x71)	329 (x119)	256 (x397)	409 (x846)	544 (x2377)
8000	439 (x35)	242 (x58)	601 (x133)	529 (x391)	666 (x1087)	1138
16000	756 (x49)	762 (x91)	1063 (x164)	1104 (x421)	1361	2074

(c)

ATP:

m \ n	50	100	200	400	800	1600
500	17 (x45)	21 (x75)	30 (x136)	37 (x332)		
1000	21 (x71)	23 (x138)	23 (x379)	32 (x760)	60 (x1556)	
2000	24 (x135)	26 (x251)	40 (x438)	43 (x1154)	64 (x2448)	151 (x4515)
4000	35 (x195)	39 (x365)	57 (x690)	72 (x1413)	115 (x3008)	201 (x6435)
8000	49 (x314)	47 (x300)	97 (x823)	134 (x1544)	225 (x3218)	382
16000	65 (x573)	95 (x730)	159 (x1097)	241 (x1931)	410	760

(d)

iterations:

m \ n	50		100		200		400		800		1600	
500	203	23	133	25	114	24	91	37				
1000	250	26	176	28	158	23	118	25	97	39		
2000	314	27	227	29	199	30	167	26	125	27	102	40
4000	375	27	289	31	246	32	173	30	156	28	122	29
8000	437	29	289	31	286	33	201	32	143	31	139	30
16000	533	29	430	33	328	33	233	33	157	31	130	31

Values on the left are for MAP-gpu, Values on the right are for ATP

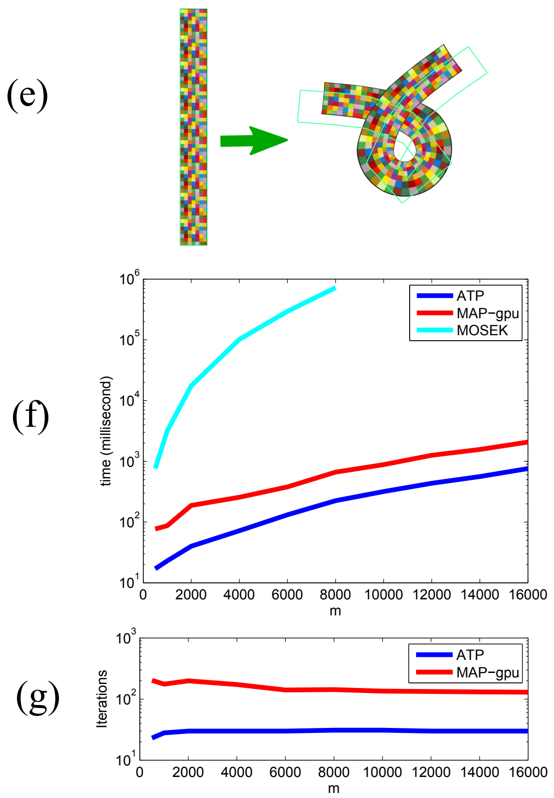


Figure 10: *Detailed timings and iteration data. In this figure, we present timings (in milliseconds) and iteration data for computation of the deformation (e), for varying problem sizes. In the charts, n and m denote the number of Cauchy basis functions and the number of sample points on ∂Ω, respectively. For timings, we report data from three different methods: direct convex optimization with MOSEK [ApS15] in (a), MAP-gpu in (b), and ATP in (c). For MAP-gpu and ATP, the number of iterations (pairs of local and global projections) needed for convergence is also given in (d). Graphs on the right, (f) and (g) plot the values on the diagonal of the charts. The upshot is that MAP-gpu tends to be orders of magnitudes faster than MOSEK, and ATP is 3-5 times faster than MAP-GPU. For a more detailed discussion, see Section 7.2.*

are in need of a solution to the following system:

$$\begin{pmatrix} \tilde{M} & \tilde{T}^t\vec{n}_i \\ \vec{n}_i^{\,t}\tilde{T} & 0 \end{pmatrix} \begin{pmatrix} x \\ \lambda \end{pmatrix} = \begin{pmatrix} \tilde{M}x_i \\ \vec{n}_i^{\,t} b_i \end{pmatrix}. \tag{19}$$

In the above system $\vec{n}_i = a_i - b_i$, x_i denotes the solution from the previous global step ($\tilde{T}x_i = a_i$), and λ denotes the Lagrange multiplier. Note that the [KABL15] approach geometrically projects a_i instead of b_i to $H_i \cap A$, but these approaches are equivalent as $\|a_i - y\|^2 = \|a_i - b_i\|^2 + \|b_i - y\|^2$ for any point $y \in H_i$ (by the Pythagorean theorem).

Performing the same analysis as was done in [KABL15], we arrive at a solution x_{i+1}:

$$x_{i+1} = x_i - \frac{\|\vec{n}_i\|^2}{\vec{n}_i^{\,t}\tilde{T}\tilde{M}^{-1}T^t\vec{n}_i}\tilde{M}^{-1}\tilde{T}^t\vec{n}_i \tag{20}$$

Note here that we only require one linear solve, giving a matrix-vector product $\tilde{M}^{-1}\tilde{T}^t\vec{n}_i$, as opposed to the two required in [KABL15]. This simplification comes from the fact that the global space in [KABL15] is not merely the image of a linear map, but also has additional linear constraints (arising from positional constraints in their perspective). Additionally, we precompute $\tilde{M}^{-1}\tilde{T}^t$, and may exploit the block structure in this calculation, equation (17), and the one above. The final step in the tangential hyperplane projection is to solve for $a_{i+1} = \tilde{T}x_{i+1}$.

Equations (18) and (20) for MAP-gpu and ATP, respectively, were evaluated on the GPU. With both local and global steps described, we also note that alternating projections for both methods will stop after $\|\vec{n}_i\| < \varepsilon$ for some small ε parameter. Once the projections stop, the Cauchy basis coefficients are taken from $x_i = (s_1, \ldots, s_n, t_1, \ldots, t_n)$, and the resulting $l = \sum_j s_j C_j$ and $\nu = \sum_j t_j C_j$ are manipulated to obtain f_z and $f_{\bar{z}}$. These are then numerically integrated over Ω to obtain the final map. This process is described in great detail in [LW16], so we omit it here, for brevity. Any additional details not found here, may also be found there.

7. Results

For the results displayed in the figures, we implemented our algorithm as a plug-in to Autodesk Maya 2016. The processor and graphics card used were an i7-4770k core with Nvidia Quadro K5000 GPU. NVIDIA CUDA Basic Linear Algebra Subroutines (cuBLAS) were used for linear algebra operations on the GPU. We used $\varepsilon = 10^{-4}$ for the stopping criterion.

For the convex optimization results (Figs. 10 and 9), we used CVX 2.1, with Mosek [ApS15] as the underlying solver. We now discuss the figures, referencing them, as needed.

7.1. Figure Discussion

In Fig. 1, some example deformations are shown. They demonstrate the efficacy of our ATP method. See caption for more.

In Fig. 2, we compare a deformation obtained with ATP to several projection methods. These projection methods take as input the top-center holomorphic mapping, computed with Cauchy basis functions [WBCG09], whereas ATP directly infers user intent from the cage, as described in Section 6.1. As can be seen, ATP produces qualitatively superior results, avoiding the poor initialization that the input mapping provides. The method of [KABL15] fails to recover from the lack of local injectivity, and the methods of [Lip12] and [CW15] are less reflective of the user's desires.

In Fig. 3, we compare the results of our ATP method with several cage-based deformation methods that utilize various kinds of barycentric coordinates. Our deformation is the only locally injective one, is qualitatively superior, and has low distortion. Conformal and isometric distortion are illustrated with heat maps.

In Fig. 4, the effect of distortion bound variation is shown. The source domain and cage are shown, and the max isometric distortion bound C_a varies from 10 to 1.25. As can be seen, the resulting deformations fit the bound and continuously scale down.

In Fig. 9, we demonstrate the near-optimality of our alternating projection methods in projecting to the discretized $\mathcal{L}_{\mathrm{V}}^{conv}$ space. Three representative examples are used, and we see that the results of MAP-gpu and ATP are visually indistinguishable from those of the ground truth, obtained via convex optimization with Mosek. The maps are not truly identical and differences may be detected if they are overlaid, but are hard to see. The image demonstrates that the limit points of the alternating projection methods are quite near the geometric projection to $\mathcal{L}_{\mathrm{V}}^{conv}$.

7.2. Timings and Iterations Data

In Fig. 10, we have detailed timing and iteration data for a particular deformation shown in window (e). The distortion constraint constants used for this image were $C_k = 0.4$, $C_a = 2$, and $C_b = 0.5$. The statement of what is illustrated in each window is in the caption, and we give additional discussion here. In windows (a)-(d), the blacked out cells correspond to $n > m$, having more Cauchy basis functions than sample points, which is undesirable for use, so we omit this data. In window (a), the last three cells are blank as the computer ran out of memory.

In comparing charts (a)-(c) and looking at graph (f), we see that both of our alternating projection methods, MAP-gpu and ATP are orders of magnitude faster than direct convex optimization with Mosek, which is the strategy used by [LW16]. The speedup factors over Mosek are given in parentheses in charts (b) and (c). As can be seen, the advantage grows as problem size grows. For practical purposes, standard problem sizes typically use between 100 and 200 basis functions and 1000 to 2000 sample points. In this domain, we see that ATP is 3-5 times faster than MAP-gpu, and this ratio does not vary too much for different problem sizes. This serves as strong validation for the modification made to obtain ATP.

Further validation for ATP is obtained in chart (d) and graph (g), which show the number of iterations (one iteration is a local projection step, and a global projection step together) needed for different problem sizes. As can be seen, MAP-gpu usually requires 6-7 times more iteration steps for standard problem sizes. While each global projection step of MAP-GPU is slightly more efficient, the fewer number of iterations required for ATP makes it faster overall. It's also interesting to note that the iteration numbers do not vary too much as problem size changes. This suggests that the $\mathcal{L}_{\mathrm{V}}$ space is approximated well for many of these varying levels of discretization. Lastly, the practical consequences of this timing data is that on standard problem sizes, ATP achieves framerates of 35-45 fps on our machine.

8. Summary & Discussion

As can be seen from the results presented above, both MAP-gpu and ATP are orders of magnitude faster than a direct convex optimization. They produce real-time framerates on standard problem sizes and are able to guarantee local injectivity and user-prescribed distortion bounds. The methods share the efficiency and speed of cage-based barycentric methods and guarantees of projection methods, combining the strengths of both. These gains are due to the convexified framework that allows for straightforward GPU parallelization, and our modified alternating projection algorithm.

On the theoretical side, we have provided a proof of convergence for ATP that holds outside of our particular setting, and may find use in other applied problems. Appendix B contains a strengthening of the bijection from [LW16] into a homeomorphism.

8.1. Limitations & Future Work

We acknowledge that our method does not support positional constraints, as is desired by many deformation methods. In general, it is an open problem to find a method which is capable of both bounded distortion deformation and positional constraints. In particular, given particular positional constraints and distortion bounds, such a deformation may not even be possible. In the future, we may attempt to incorporate hard or soft positional constraints into a related method.

We also acknowledge that our method is restricted to planar deformations. As a surface immersed in $\mathbb{R}^3$ inherits a Riemannian metric, and thus a notion of harmonicity for functions, we may look to apply some of these ideas to parametrization of surfaces.

Lastly, as noted above, ATP and ANP (Appendix A) may find use in other applications, and we intend to search for these. Additionally, theoretical questions on convergence rate and bounds on the difference of the limit point from geometric projection remain to be investigated.

Acknowledgements

This research was partially funded by the Israel Science Foundation (grants No. 1869/15 and 2102/15). We also thank Ben Hefetz for implementation contributions, Zohar Levi for assistance with Figure 2, and Alec Jacobson for the hand image in Figure 1.

References

[AL13] Aigerman N., Lipman Y.: Injective and bounded distortion mappings in 3d. *ACM Trans. Graph. 32*, 4 (July 2013), 106:1–106:14. 1, 3

[APL14] Aigerman N., Poranne R., Lipman Y.: Lifted bijections for low distortion surface mappings. *ACM Trans. Graph. 33*, 4 (July 2014), 69:1–69:12. 1, 3

[ApS15] ApS M.: *The MOSEK optimization toolbox for MATLAB manual. Version 7.1 (Revision 28).*, 2015. 9, 10, 11

[BB93] Bauschke H. H., Borwein J. M.: On the convergence of von neumann's alternating projection algorithm for two sets. *Set-Valued Analysis 1*, 2 (1993), 185–212. 3, 6, 7

[BB96] Bauschke H. H., Borwein J. M.: On projection algorithms for solving convex feasibility problems. *SIAM Rev. 38*, 3 (Sept. 1996), 367–426. 3

[BDS*12] Bouaziz S., Deuss M., Schwartzburg Y., Weise T., Pauly M.: Shape-up: Shaping discrete geometry with projections. *Comput. Graph. Forum 31*, 5 (Aug. 2012), 1657–1667. 3

[BS08] Botsch M., Sorkine O.: On linear variational surface deformation methods. *IEEE Transactions on Visualization and Computer Graphics 14*, 1 (Jan. 2008), 213–230. 2

[CCW16] Chien E., Chen R., Weber O.: Bounded distortion harmonic shape interpolation. *ACM Trans. Graph. 35*, 4 (July 2016), 105:1–105:15. 1, 4

[CLW16] Chien E., Levi Z., Weber O.: Bounded distortion parametrization in the space of metrics. *ACM Trans. Graph. 35*, 6 (Nov. 2016), 215:1–215:16. 1

[CPS11] Crane K., Pinkall U., Schröder P.: Spin transformations of discrete surfaces. *ACM Trans. Graph. 30* (2011). 2

[CW15] Chen R., Weber O.: Bounded distortion harmonic mappings in the plane. *ACM Trans. Graph. 34*, 4 (July 2015), 73:1–73:12. 1, 2, 3, 4, 11

[Dur04] Duren P.: *Harmonic mappings in the plane*. Cambridge University Press, 2004. 4

[FL16] Fu X.-M., Liu Y.: Computing inversion-free mappings by simplex assembly. *ACM Trans. Graph. 35*, 6 (Nov. 2016), 216:1–216:12. 1

[FLG15] Fu X.-M., Liu Y., Guo B.: Computing locally injective mappings by advanced mips. *ACM Transactions on Graphics (TOG) 34*, 4 (2015), 71. 3

[Flo97] Floater M. S.: Parametrization and smooth approximation of surface triangulations. *Comput. Aided Geom. Des. 14*, 3 (Apr. 1997), 231–250. 3

[HF06] Hormann K., Floater M. S.: Mean value coordinates for arbitrary planar polygons. *ACM Trans. Graph. 25*, 4 (Oct. 2006), 1424–1441. 2, 3

[Jac13] Jacobson A.: Bijective mappings with generalized barycentric coordinates: A counterexample. *Journal of Graphics Tools 17*, 1-2 (2013), 1–4. 2

[JMDG07] Joshi P., Meyer M., DeRose T., Green B.: Harmonic coordinates for character articulation. In *ACM Transactions on Graphics (TOG)* (2007), vol. 26, p. 71. 2, 3

[KABL14] Kovalsky S. Z., Aigerman N., Basri R., Lipman Y.: Controlling singular values with semidefinite programming. *ACM Trans. Graph. 33*, 4 (July 2014), 68:1–68:13. 1, 3

[KABL15] Kovalsky S. Z., Aigerman N., Basri R., Lipman Y.: Large-scale bounded distortion mappings. *ACM Trans. Graph. 34*, 6 (Oct. 2015), 191:1–191:10. 1, 2, 3, 6, 9, 10, 11

[Lip12] Lipman Y.: Bounded distortion mapping spaces for triangular meshes. *ACM Trans. Graph. 31*, 4 (July 2012), 108:1–108:13. 1, 2, 3, 11

[LKCOL07] Lipman Y., Kopf J., Cohen-Or D., Levin D.: GPU-assisted positive mean value coordinates for mesh deformations. In *Symposium on geometry processing* (2007), vol. 257, pp. 117–123. 2

[LLCO08] Lipman Y., Levin D., Cohen-Or D.: Green coordinates. In *ACM Transactions on Graphics (TOG)* (2008), vol. 27, p. 78. 2

[LW16] Levi Z., Weber O.: On the convexity and feasibility of the bounded distortion harmonic mapping problem. *ACM Trans. Graph. 35*, 4 (July 2016), 106:1–106:15. 1, 2, 3, 4, 5, 10, 11, 13, 14

[LZ14] Levi Z., Zorin D.: Strict minimizers for geometric optimization. *ACM Trans. Graph. 33*, 6 (Nov. 2014), 185:1–185:14. 1

[LZX*08] Liu L., Zhang L., Xu Y., Gotsman C., Gortler S. J.: A local/global approach to mesh parameterization. In *Proceedings of the Symposium on Geometry Processing* (Aire-la-Ville, Switzerland, Switzerland, 2008), SGP '08, Eurographics Association, pp. 1495–1504. 2, 3, 8

[PL14] Poranne R., Lipman Y.: Provably good planar mappings. *ACM Trans. Graph. 33*, 4 (July 2014), 76:1–76:11. 1, 3

[RPPSH17] Rabinovich M., Poranne R., Panozzo D., Sorkine-Hornung O.: Scalable locally injective mappings. *ACM Transactions on Graphics 36* (2017). to appear. 3

[SA07] Sorkine O., Alexa M.: As-rigid-as-possible surface modeling. In *Proceedings of the Fifth Eurographics Symposium on Geometry Processing* (Aire-la-Ville, Switzerland, Switzerland, 2007), SGP '07, Eurographics Association, pp. 109–116. 2, 3, 8

[SKPSH13] Schüller C., Kavan L., Panozzo D., Sorkine-Hornung O.: Locally injective mappings. In *Proceedings of the Eleventh Eurographics/ACMSIGGRAPH Symposium on Geometry Processing* (Aire-la-Ville, Switzerland, Switzerland, 2013), SGP '13, Eurographics Association, pp. 125–135. 3

[SS15] Smith J., Schaefer S.: Bijective parameterization with free boundaries. *ACM Trans. Graph. 34*, 4 (July 2015), 70:1–70:9. 3

[SS16] Schost É., Spaenlehauer P.-J.: A quadratically convergent algorithm for structured low-rank approximation. *Foundations of Computational Mathematics 16*, 2 (2016), 457–492. 3

[TSG*14] Tang C., Sun X., Gomes A., Wallner J., Pottmann H.: Form-finding with polyhedral meshes made simple. *ACM Trans. Graph. 33*, 4 (July 2014), 70:1–70:9. 3

[Tut63] Tutte W. T.: How to draw a graph. *Proceedings of the London Mathematical Society s3-13*, 1 (1963), 743–767. 3

[VMW15] Vaxman A., Müller C., Weber O.: Conformal mesh deformations with möbius transformations. *ACM Transactions on Graphics (TOG) 34*, 4 (2015), 55. 2

[vN50] von Neumann J.: *Functional Operators Volume II: The Geometry of Orthogonal Spaces*. Princeton University Press, 1950. 3, 8

[WBCG09] Weber O., Ben-Chen M., Gotsman C.: Complex barycentric coordinates with applications to planar shape deformation. *Computer Graphics Forum 28*, 2 (2009), 587–597. 2, 3, 5, 11

[WBCGH11] Weber O., Ben-Chen M., Gotsman C., Hormann K.: A complex view of barycentric mappings. In *Computer Graphics Forum* (2011), vol. 30, Wiley Online Library, pp. 1533–1542. 2, 3

[Web17] Weber O.: Planar shape deformation. In *Generalized Barycentric Coordinates in Computer Graphics and Computational Mechanics*, Hormann K., Sukumar N., (Eds.). CRC Press, Boca Raton, FL, 2017, ch. 7. 8

[WG10] Weber O., Gotsman C.: Controllable conformal maps for shape deformation and interpolation. *ACM Trans. Graph. 29*, 4 (July 2010), 78:1–78:11. 2

[WMZ12] WEBER O., MYLES A., ZORIN D.: Computing extremal quasiconformal maps. *Comput. Graph. Forum 31*, 5 (Aug. 2012), 1679–1689. 1

[WZ14] WEBER O., ZORIN D.: Locally injective parametrization with arbitrary fixed boundaries. *ACM Trans. Graph. 33*, 4 (July 2014), 75:1–75:12. 3

Appendix A: ATP Scope & Generalization

As mentioned at the end of Section 5, the proof of Theorem 5 only requires that A and B be closed convex subsets of $\mathbb{R}^d$. So we actually proved a more general theorem:

Theorem 6 If $A, B \subseteq \mathbb{R}^d$ are closed convex subsets with $A \cap B \neq 0$, then the sequences $(a_i)_{i=1}^{\infty}$ and $(b_i)_{i=1}^{\infty}$ constructed with ATP converge to a limit point $p^* \in A \cap B$.

Also, one might consider performing "tangential" projections for every step, to even further lessen the number of iterations needed for convergence. The reason for the quotes is that the projections in such a method are no longer tangential projections, but are better thought of as restrictions that force each step to be orthogonal to the previous one. As such, we refer to this method as ANP for alternating normal projections.

Adopting the definition of $\tilde{g}_i$ from the proof of Lemma 4, we may define $\tilde{H}_i$ to be its zero set:

$$\tilde{H}_i := \{x \in \mathbb{R}^d : \tilde{g}_i(x) = \langle b_i - a_{i+1}, a_{i+1} - x\rangle = 0\}$$

Then ANP constructs sequences $(a_i)_{i=1}^{\infty}$ in A and $(b_i)_{i=1}^{\infty}$ in B according to the following equations:

$$\begin{aligned} a_1 &= \mathrm{proj}_A(p) \\ b_1 &= \mathrm{proj}_B(a_1) \\ a_i &= \mathrm{proj}_{A\cap H_{i-1}}(b_{i-1}), \quad && i \geq 2 \\ b_i &= \mathrm{proj}_{B\cap \tilde{H}_{i-1}}(a_i), && i \geq 2 \end{aligned}$$

An analogue of Lemma 3 on feasibility may be proven in the same way as was done in Section 5.1. A proof of convergence for ANP goes through in a similar fashion, asking for an argument like that of Lemma 4 for both restricted projection steps. The arguments dovetail in an interesting way with the inequalities $g_{i-1}(q) > 0$ and $\tilde{g}_{i-1}(q) > 0$ obtained in one such argument being necessary for the success of the next one. We show this below, referencing Lemma 4 often:

Lemma 7 In ANP, for $i \geq 2$, the following inequalities hold:

$$\|a_i - q\|^2 \leq \|b_{i-1} - q\|^2 - \|b_{i-1} - a_i\|^2, \tag{21}$$

$$\|b_i - q\|^2 \leq \|a_i - q\|^2 - \|a_i - b_i\|^2 \tag{22}$$

for any $q \in A \cap B$.

Proof We argue only for equation (21) as the argument for equation (22) is entirely analogous. We again have that:

$$\begin{aligned} \|b_{i-1} - q\|^2 &= \|b_{i-1} - a_i + a_i - q\|^2 \\ &= \|b_{i-1} - a_i\|^2 + 2\langle b_{i-1} - a_i, a_i - q\rangle + \|a_i - q\|^2 \end{aligned}$$

which is just equation (14) with an index shift. As before, we aim to show that $\tilde{g}_{i-1}(q) = \langle b_{i-1} - a_i, a_i - q\rangle \geq 0$, and argue by contradiction.

If $\tilde{g}_{i-1}(q) < 0$, consider the line between q and a_{i-1}. For $i = 2$, we have that $g_1(q) > 0$ by Kolmogorov's criterion (K). For $i > 2$, we note that in arguing for equation (22) for the previous step (so with index shifted back one), we proved $g_{i-1}(q) > 0$. By definition, $g_{i-1}(a_{i-1}) < 0$, so there is a point r along the line for which $g_{i-1}(r) = 0$ and $r \in H_{i-1}$.

The rest of the argument proceeds as it does in the proof for Lemma 4. We have the analogue of equation (15), which is $\tilde{g}_{i-1}(a_{i-1}) < 0$. With our assumption $\tilde{g}_{i-1}(q) < 0$ we get that $\tilde{g}_i(r) < 0$, which violates Kolmogorov's criterion (K) when applied to $H_{i-1} \cap A$ as a convex subset of H_{i-1}. ☐

With Lemma 7 above, it is easy to establish the following:

Theorem 8 If $A, B \subseteq \mathbb{R}^d$ are closed convex subsets with $A \cap B \neq 0$, then the sequences $(a_i)_{i=1}^{\infty}$ and $(b_i)_{i=1}^{\infty}$ constructed with ANP converge to a limit point $p^* \in A \cap B$.

Lastly, we note that ANP was not used for the problem at hand in this paper, as the local normal projection step would be of the same difficulty as the overall problem, requiring one to project to the intersection of an affine space and a convex set.

Appendix B: $\mathcal{BD}$ Homeomorphic to $\mathcal{L}_\nu$

In [LW16] the authors construct an operator $\mathcal{F} : \mathcal{BD} \to \mathcal{L}_\nu$, which they show to be bijective under appropriate choices of integration constaints. In this section, we quotient out by these integration constants, and argue for continuity of the identifications induced by $\mathcal{F}$ and its inverse, showing they are not merely bijections, but also homeomorphisms, under all practical settings.

Technical Conditions

With our cage-based approach, the domain Ω is a compact region bounded by a polygonal Jordan curve with a finite number of segments. As Ω is not open, harmonic and holomorphic functions on Ω are those that are extendible to harmonic or holomorphic functions on some open neighborhood of Ω. In the following arguments, due to our setting, we consider these function spaces with domain $\Omega_\lambda = \{z \in \mathbb{C} : d(z, \Omega) \leq \lambda\}$, where $\lambda > 0$ is a small constant such that Ω_λ is simply-connected.

Additionally, due to the nature of Ω, Ω_λ is of finite diameter as a length space, meaning that the lengths of the shortest paths (within Ω_λ) between pairs of points in Ω_λ is bounded above by some positive constant.

Spaces and Maps

Let us define our functional spaces and the maps between them. We must demonstrate two homeomorphisms:

$$\mathrm{Har}_{LI}\big/\tau \;\cong\; \mathrm{D}_{LI}^{\mathrm{Har}} \;\cong\; \mathrm{Hol}_{cyl} \times \mathrm{Hol}_{\mathbb{D}} \tag{23}$$

Har_{LI} is the set of locally-injective harmonic mappings on Ω:

$$\mathrm{Har}_{LI} := \left\{ f : \Omega_\lambda \to \mathbb{R}^2 \;\middle|\; \Delta f = 0 \text{ in } \Omega_\lambda,\ \det(J_f) > 0 \text{ in } \Omega \right\}.$$

The quotient by τ, denotes that we are considering such maps up to post-composition with translations $z \mapsto z+c$. $\mathrm{D}_{LI}^{\mathrm{Har}}$ is the space of Wirtinger derivatives of locally-injective harmonic mappings:

$$\mathrm{D}_{LI}^{\mathrm{Har}} := \left\{ (\phi, \overline{\psi}) \in \mathrm{Hol}_{\mathbb{C}-\{0\}} \times \overline{\mathrm{Hol}} \,\middle|\, |\phi(z)| > \left|\overline{\phi}(z)\right| \forall z \in \Omega \right\}.$$

Above, Hol and $\overline{\mathrm{Hol}}$ denote the set of holomorphic and anti-holomorphic mappings on Ω_λ, respectively.

The maps $\mathrm{Diff} : \mathrm{Har}_{LI} \to \mathrm{D}_{LI}^{\mathrm{Har}}$ and $\mathrm{Int} : \mathrm{D}_{LI}^{\mathrm{Har}} \to \mathrm{Har}_{LI} \big/ \tau$, which are the differentiation and integration operators, define the first homeomorphism in equation (23). Diff takes a harmonic mapping to its Wirtinger derivatives, which is invariant under post-composition by a translation, so induces a well-defined map $\underline{\mathrm{Diff}} : \mathrm{Har}_{LI} \big/ \tau \to \mathrm{D}_{LI}^{\mathrm{Har}}$. Int takes a pair of derivatives to a mapping (up to translation), obtained after integration and summation.

For the second homeomorphism, we first define the notation:

$$cyl := \mathbb{C} \Big/ \langle z \mapsto z + 2\pi i \rangle$$

for the Riemann surface obtained by quotient of $\mathbb{C}$ by the group of translations by integer multiples of $2\pi i$. We may define $\Xi : \mathrm{D}_{LI}^{\mathrm{Har}} \to \mathrm{Hol}_{cyl} \times \mathrm{Hol}_{\mathbb{D}}$:

$$\Xi\left(\phi, \overline{\psi}\right) = \left(\log(\phi), \frac{\psi}{\phi} \right)$$

where no branch is needed for log due to the quotient in *cyl*. The second component of Ξ is ν for a pair of Wirtinger derivatives.

First Homeomorphism

For the sake of brevity, we only argue for the first homeomorphism of equation (23), and focus on the continuity arguments, leaving the proofs of bijection to the reader. These bijection arguments are similar (and simpler) than those made in [LW16] for the $\mathcal{H}$ space. To fully mimic the results there, one will have to apply the appropriate distortion constraints to each functional space given constants (C_k, C_a, C_b). For the second homeomorphism, the continuity arguments are easier and follow from compactness of Ω_λ and its images under continuous maps.

The standard compact-open topology is used on all of our functional spaces. As all of our mappings are to a metric space $\mathbb{C}$ or (some subset or quotient) and have compact domain Ω or Ω_λ, the standard compact-open topology is equivalent to the topology of uniform convergence. This is just the topology induced by the L^∞ norm, e.g., $d(f,g) = \max_{z \in \Omega_\lambda} |f(z) - g(z)|$ on Har_{LI}. On $\mathrm{D}_{LI}^{\mathrm{Har}}$, we impose the L^∞ norm where the maximum is taken over just Ω for the sake of our argument. This is still a norm as holomorphic and antiholomorphic functions agree entirely if they agree on an open set, such as the interior of Ω.

Lastly, before continuing, recall that f_z and $f_{\bar{z}}$ encode the similarity and antisimilarity parts of the Jacobian, and that the real and imaginary parts of f_z and $f_{\bar{z}}$ are linear functions of u_x, u_y, v_x, v_y (where $f = (u,v)$, as always). In fact, the linear transformation is a similarity transformation (in $\mathbb{R}^4$), and $|f_z|^2 + |f_{\bar{z}}|^2 = \frac{1}{2}\|J_f\|^2$. Thus, the topology on $\mathrm{D}_{LI}^{\mathrm{Har}}$ induced by the Frobenius norm is equivalent to that induced by the standard complex norm on $\mathbb{C}^2$, and we may also argue with the partial derivatives u_x, u_y, v_x, v_y.

Continuity of Diff

We show that the set $\mathrm{Diff}^{-1}(B_\varepsilon(f_z, f_{\bar{z}}))$ is open for any $(f_z, f_{\bar{z}}) \in \mathrm{D}_{LI}^{\mathrm{Har}}$ and $\varepsilon > 0$ (where $B_\varepsilon(\cdot)$ is an open ε-ball). For this, we consider a mapping $g \in \mathrm{Diff}^{-1}(B_\varepsilon(f_z, f_{\bar{z}}))$ and construct a $\delta > 0$ such that $B_\delta(g) \subseteq \mathrm{Diff}^{-1}(B_\varepsilon(f_z, f_{\bar{z}}))$. We need to argue that bounding the L^∞ norm of a mapping bounds the L^∞ norm of its derivatives, or Jacobian, a fact that follows from the harmonicity of our mappings.

Note that u_x, u_y, v_x, v_y are all harmonic, and thus we may bound their L^∞ norm by considering their values on just the boundary. The mean value property and Green's Theorem gives us the following inequality on the magnitude of $\partial_w u$, the directional derivative of u in the direction of a unit vector w:

$$|\partial_w u(z)| \leq \frac{2}{r} \max_{z \in \partial B_r} |u|$$

for any $r > 0$. An analogous inequality clearly holds for directional derivates of v in any direction w.

Consider $h \in B_\delta(g)$ so that $|h - g| < \delta$ on Ω_λ. By the previous inequality, with $r = \lambda$, we find that $|\partial_x \mathrm{Re}\,(h-g)\,(z)| \leq \frac{2}{\lambda}\delta$ for all $z \in \partial\Omega$. Analogous inequalities hold for $\partial_y \mathrm{Re}\,(h-g)\,, \partial_x \mathrm{Im}\,(h-g)\,, \partial_y \mathrm{Im}\,(h-g)$. These inequalities together imply that the $\|D(h-g)\| \leq \frac{8}{\lambda}\delta$ for all $z \in \partial\Omega$, where $D(h-g)$ denotes the Jacobian. From this, we see that a choice of $\delta < \frac{\lambda\varepsilon}{8}$ implies that $B_\delta(g) \subseteq \mathrm{Diff}^{-1}(B_\varepsilon(f_z, f_{\bar{z}}))$. It is clear that this argument need not change when considering the quotient topology on $\mathrm{Har}_{LI} \big/ \tau$.

Continuity of Int

As in the previous section, we show that a set $\mathrm{Int}^{-1}(B_\varepsilon(f))$ is open. Consider $B_\delta((g_z, g_{\bar{z}}))$ for a pair of functions $(g_z, g_{\bar{z}}) \in \mathrm{Int}^{-1}(B_\varepsilon(f))$. Pairs of functions in this set represent Jacobians that differ from that given by $(g_z, g_{\bar{z}})$ by $\sqrt{2}\delta$ in Frobenius norm. Utilizing the finite diameter $\mathrm{diam}(\Omega_\lambda)$, and the fact that $|\int h| \leq \int |h|$, we see that any integrated Jacobian will not differ from g by more than $\sqrt{2}\delta\,\mathrm{diam}(\Omega_\lambda)$. In essence, controlling the Jacobian difference allows for control over the difference of the integrated values due to the finite diameter. Thus, choosing $\delta < \varepsilon\left(\sqrt{2}\,\mathrm{diam}(\Omega_\lambda)\right)^{-1}$ will ensure that $B_\delta((g_z, g_{\bar{z}})) \subseteq \mathrm{Int}^{-1}(B_\varepsilon(f))$. It's not hard to to see that this argument is compatible with the quotient topology.

The resulting theorem is:

Theorem 9 $\mathrm{Har}_{LI} \big/ \tau \cong \mathrm{Hol}_{cyl} \times \mathrm{Hol}_{\mathbb{D}}$ in the standard compact-open topology,

and this implies the homeomorphisms between $\mathcal{BD}$ and $\mathcal{L}_\nu$ by translating the distortion constraints in one domain to the other, as done in [LW16].