PROCEEDINGS A

royalsocietypublishing.org/journal/rspa

Research

Cite this article: Mosleh S, Choi GPT, Mahadevan L. 2025 Data-driven quasi-conformal morphodynamic flows. *Proc. R. Soc. A* **481**: 20240527. https://doi.org/10.1098/rspa.2024.0527

Received: 18 July 2024
Accepted: 14 March 2025

Subject Areas:
computational biology, applied mathematics, computer graphics

Keywords:
quasi-conformal flow, mathematical modelling, morphodynamics, growth

Author for correspondence:
L. Mahadevan
e-mail: lmahadev@g.harvard.edu

Data-driven quasi-conformal morphodynamic flows

Salem Mosleh[1,4,†], Gary P. T. Choi[5,†] and L. Mahadevan[1,2,3]

[1]School of Engineering and Applied Sciences, [2]Department of Physics, and [3]Department Organismic and Evolutionary Biology, Harvard University, Cambridge, MA 02138, USA
[4]Department of Natural Sciences, University of Maryland Eastern Shore, Princess Anne, MD 21853, USA
[5]Department of Mathematics, The Chinese University of Hong Kong, Hong Kong, China

GPTC, 0000-0001-5407-9111; LM, 0000-0002-5114-0519

Temporal imaging of biological epithelial structures yields shape data at discrete-time points, leading to a natural question: how can we reconstruct the most likely path of growth patterns consistent with these discrete observations? We present a physically plausible framework to solve this inverse problem by creating a framework that generalizes quasi-conformal maps to quasi-conformal flows. By allowing the spatio-temporal variation of the shear and dilation fields during the growth process, subject to regulatory mechanisms, we are led to a type of generalized Ricci flow. When guided by observational data associated with surface shape as a function of time, this leads to a constrained optimization problem. Deploying our data-driven algorithmic approach to the shape of insect wings, leaves and even sculpted faces, we show how optimal quasi-conformal flows allow us to characterize the morphogenesis of a range of surfaces.

1. Introduction

Morphogenesis, the process by which organisms generate and regulate their shape, involves a complex interplay between biochemical and physical factors. A key goal is to identify the biophysical factors—and how they interact with biochemical signalling—that drive the local deformations and flows during morphogenesis [1]. However, often we do not obtain the local growth and

†These authors contributed equally to this study.

Electronic supplementary material is available online at https://doi.org/10.6084/m9.figshare.c.7741393.

THE ROYAL SOCIETY PUBLISHING

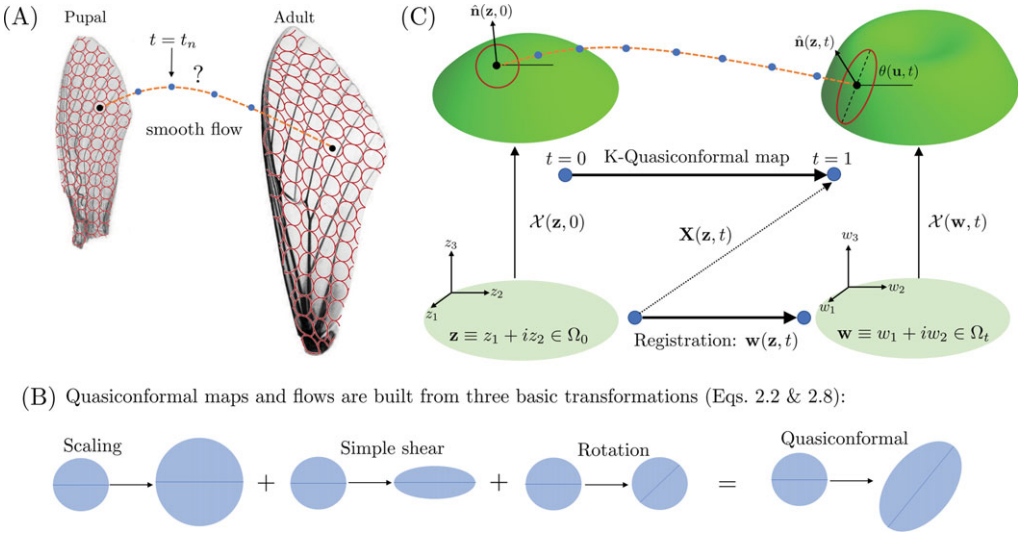

Figure 1. Quasi-conformal flows for morphodynamics. (A) Images of growing structures do not provide the correspondence between points at different times t (normalized by dividing with the total duration T). We need to infer the (Lagrangian) trajectory of points computationally. (B) Quasi-conformal (QC) maps and flows are useful for accomplishing this, since they describe systems that deform smoothly with anisotropic growth, where infinitesimal circles deform into infinitesimal ellipses over time due to the action of scaling, shear and rotation fields. (C) An illustration of the geometry of deforming surfaces. Each surface, for $t > 0$, is given as the function $\mathcal{X}(\mathbf{w}, t) : \Omega_t \in \mathbb{R}^2$. Our task is to find a suitable registration $\mathbf{w}(\mathbf{z}, t)$ that allows us to infer the true growth patterns shown in panel (A) by constructing the function $\mathbf{X}(\mathbf{z}, t) \equiv \mathcal{X}(\mathbf{w}(\mathbf{z}, t), t)$. Here $\hat{\mathbf{n}}(\mathbf{z}, t)$ is the unit normal to the surface at time t.

flow patterns of cells from experiments, which require live imaging. Thus, a key mathematical challenge is to infer the local growth patterns from snapshots of growing structures taken at different developmental stages, possibly from different individuals, and learn the dynamical laws that generate these structures. This requires us to determine the mapping (or flow) rules that connect points on the different snapshots of the growing structure.

One way to address this correspondence problem is through the use of conformal maps between planar domains or two-dimensional surfaces embedded in three dimensions. These maps do not distort angles [2–5]. For example, given two simply connected regions in the plane, by the Riemann mapping theorem, a unique (up to Möbius transformations) conformal map exists between them and can be used to determine the correspondence between all points in the interior domains given their boundary curves. This mathematical fact has been put to use to study growing leaves [6], insect wings [7] and other biological systems [8,9]. However conformal maps—which generate isotropic growth, mapping infinitesimal circles to rescaled infinitesimal circles—are inadequate to describe biological morphogenesis where anisotropic growth—which distorts angles and maps infinitesimal circles to infinitesimal ellipses (figure 1A)—is the rule rather than the exception [10,11].

For nearly a century, a generalization of conformal maps known as quasi-conformal maps that do account for dilation, shear and rotation (figure 1B) has been known [12], although it is only recently that they have been deployed to study morphogenesis, e.g. to allow for planar deformations that minimize elastic distortions while also accounting for the presence of landmarks (points whose mapping is given *a priori*) [13,14]. When combined with different types of regularizations, e.g. minimizing squared gradients of the Beltrami fields [13] or restrictions to smooth Teichmüller maps [14], it has been possible to study both the phylogeny and ontogeny of wing shapes using two-dimensional quasi-conformal maps. In parallel, the field of computational

quasi-conformal mappings has grown rapidly with applications to complex three-dimensional shapes, such as teeth, faces, etc. [15,16].

As we gain the ability to monitor growing systems and extract their shape over time continuously, it is natural to ask whether one can take advantage of the temporal structure of the growth process by considering *flows* (instead of maps between discrete times points), which give smooth transformations of a shape into another shape over time [17,18]. To this end, we consider k-quasi-conformal flows, which have been studied previously [19–22], in place of K-quasi-conformal maps that deform (infinitesimal) circles to ellipses with eccentricity bounded by the value K, but not in the context here, i.e. to compress, and quantify biological shape changes. As shown below, k-quasi-conformal flows can be considered as the $dt \to 0$ limit of a sequence of $(1 + kdt)$-quasi-conformal maps.

Due to the limited resolution of spatio-temporal live-tracking of points, the prescribed data often are not sufficient to uniquely determine a quasi-conformal map and, therefore, additional criteria are required. Common criteria are minimal distortion [23] and minimal spatial variation [13] mappings. However, growing structures (including leaves and electrochemical interfaces) are governed by dynamical processes that couple the curvature, mechanical strain and other fields. Here, in addition to considering minimal distortion and spatial variation flow, we find optimal quasi-conformal flows that fit the prescribed data to a dynamical equation that takes the form of a geometric partial differential equation, similar to a generalized Ricci flow. Our approach is a step towards blending data-driven and physics-based approaches, since it allows us to automate the process of the discovery of growth laws and predict growth patterns from incomplete data.

This paper is organized as follows. In §2, we discuss the geometry and kinematics relevant to evolving surfaces, and define the cost function that will be minimized in the following sections. Section 3 will discuss the discretization and numerical implementation, and §4 will present experiments performed on various natural and synthetic examples of flowing surfaces. We conclude our work and discuss future directions in §5.

2. Formulation of quasi-conformal flow models

This section introduces the formalism for optimal quasi-conformal flows and defines the cost function that we use in the following sections.

(a) Geometry of surface and quasi-conformal maps

We start with a smooth one parameter family (flow) of surfaces $\tilde{\mathcal{X}}(\mathbf{w}, \tilde{t})$, which are embedded in $\mathbb{R}^3$ with parametrization $\mathbf{w} \in \Omega_t \subset \mathbb{R}^2$ and time $\tilde{t} \in [0, T]$ (figure 1C). We normalize lengths so that $\mathcal{X}(\mathbf{w}, t) \equiv \tilde{\mathcal{X}}(\mathbf{w}, t)/\sqrt{\mathcal{A}}$, where $\mathcal{A}$ is the area of $\tilde{\mathcal{X}}(\mathbf{w}, 0)$, and normalize time so that $t \equiv \tilde{t}/T \in [0, 1]$. Our goal is to find a registration—a correspondence between points on the surfaces across time—as a coordinate system $\mathbf{z} = z_1 + iz_2 \in \Omega_0 \subset \mathbb{R}^2$ and a family of maps $\mathbf{w}(\mathbf{z}, t)$ that gives a common parametrization of the surface across time, $\mathbf{X}(\mathbf{z}, t) \equiv \mathcal{X}(\mathbf{w}(\mathbf{z}, t), t) : \Omega_0 \to \mathbb{R}^3$. Thus, for a fixed $\mathbf{z}$, the function $\mathbf{X}(\mathbf{z}, t)$ gives the trajectory of the 'same' point evolving over time (orange dashed curve in figure 1C) and our task is to obtain these trajectories.

For notational convenience, we will express the coordinate vector as $\mathbf{z} \equiv z_i$ $(i = 1, 2)$, and define derivative operators as $\partial_i \equiv \partial/\partial z_i$. Furthermore, we will use the Einstein summation convention, where pairs of repeated indices in an expression are summed over. Using this notation and the chain rule, a tangent vector to the surface that connects two infinitesimally separated points z_i and $z_i + \mathrm{d}z_i$ at time t can be written as $d\mathbf{X}(\mathbf{z}, t) = \partial_i \mathbf{X}(\mathbf{z}, t)\, \mathrm{d}z_i$.

To describe the kinematics of growing surfaces, it is useful to consider how angles and lengths change as the surface evolves. The length squared of the line segment connecting the points z_i and $z_i + \mathrm{d}z_i$ is given by

$$\mathrm{d}\ell^2(t) = |d\mathbf{X}(\mathbf{z}, t)|^2 = g_{ij}(\mathbf{z}, t)\, \mathrm{d}z_i\, \mathrm{d}z_j \quad \text{and} \quad g_{ij}(\mathbf{z}, t) \equiv \partial_i \mathbf{X}(\mathbf{z}, t) \cdot \partial_j \mathbf{X}(\mathbf{z}, t), \tag{2.1}$$

where $g_{ij}(\mathbf{z},t)$ gives the components of the metric tensor in the z_i coordinate system. Using the metric tensor, we can define lengths of (and therefore angles between) line segments on the surface at time t. The inverse of the metric is denoted (with upper indices) as $g^{ij}(\mathbf{z},t)$, and satisfies $g^{ij}(\mathbf{z},t)g_{jk}(\mathbf{z},t)=\delta_{ik}$, where δ_{ik} is the Kronecker delta.

Next, we describe the metric changes in terms of rotations, dilation and shear of the tangent plane. To facilitate this, we consider the Cartesian coordinates on the plane tangent to the surface $\mathbf{X}(\mathbf{z},0)$ at $\mathbf{z}$, which we denote as $\tilde{\mathbf{z}}$. The Jacobian of the transformation between $\tilde{\mathbf{z}}$ and $\mathbf{z}$ is given by $\mathbf{J}_i(\mathbf{z})\equiv\partial_i\tilde{\mathbf{z}}(\mathbf{z})$ and the initial metric is given by $g_{ij}(\mathbf{z},0)=\mathbf{J}_i^{\top}(\mathbf{z})\cdot\mathbf{J}_j(\mathbf{z})$, where $\top$ (as a subscript or superscript) indicates a vector or matrix transpose. We will also denote the inverse Jacobian $\mathbf{J}^i(\mathbf{z})$ so that $\mathbf{J}^i_{\top}(\mathbf{z})\mathbf{J}_j(\mathbf{z})=\delta_{ij}$ and $\mathbf{J}^i(\mathbf{z})\mathbf{J}_i^{\top}(\mathbf{z})=\mathbf{I}_2$, where $\mathbf{I}_2$ is the two-dimensional identity matrix. As the surface evolves over time, the tangent plane at $\mathbf{X}(\mathbf{z},0)$ will be convected by the flow and will undergo rotation, dilation and shear that we explicitly account for using a polar decomposition of the deformation gradient of the tangent plane followed by a coordinate transform—using the Jacobian $\mathbf{J}_i(\mathbf{z})$—which leads to the following expression for the metric tensor at time t:

$$g_{ij}(\mathbf{z},t)=\omega(\mathbf{z},t)\mathbf{J}_i^{\top}(\mathbf{z})\mathbf{R}^{\top}(\mathbf{z},t)\mathbf{E}(\mathbf{z},t)\mathbf{R}(\mathbf{z},t)\mathbf{J}_j(\mathbf{z}) \tag{2.2}$$

and

$$\mathbf{R}(\mathbf{z},t)\equiv\begin{pmatrix}\cos[\theta(\mathbf{z},t)] & -\sin[\theta(\mathbf{z},t)]\\ \sin[\theta(\mathbf{z},t)] & \cos[\theta(\mathbf{z},t)]\end{pmatrix},\quad \mathbf{E}(\mathbf{z},t)\equiv\begin{pmatrix}K(\mathbf{z},t) & 0\\ 0 & \dfrac{1}{K(\mathbf{z},t)}\end{pmatrix}. \tag{2.3}$$

Here the scalar $\omega(\mathbf{z},t)>0$ is the conformal factor giving the dilation of the transformation, the dilatation $K(\mathbf{z},t)\geq 1$ is the ratio of major to minor axes of an ellipse at time t that started as an infinitesimal circle on $\mathbf{X}(\mathbf{z},0)$—which is the square root of the ratio between the components of $\mathbf{E}(\mathbf{z},t)$, since the metric gives the squared length of infinitesimal line segments. The two-dimensional rotation matrix $\mathbf{R}(\mathbf{z},t)$ gives the direction of the major axes of the ellipse in the tangent plane at $\mathbf{z}$, with $\theta(\mathbf{z},t)\in[0,\pi)$ being uniquely determined when $K(\mathbf{z},t)>1$ (figure 1). The Beltrami coefficient [12] of this transformation is then given by

$$\mu(\mathbf{z},t)=\frac{K(\mathbf{z},t)-1}{K(\mathbf{z},t)+1}\exp[-2\mathrm{i}\theta(\mathbf{z},t)]. \tag{2.4}$$

The inverse metric corresponding to equation (2.2) can be written as

$$g^{ij}(\mathbf{z},t)=\frac{1}{\omega(\mathbf{z},t)}\mathbf{J}^i_{\top}(\mathbf{z})\mathbf{R}^{\top}(\mathbf{z},t)\mathbf{E}^{-1}(\mathbf{z},t)\mathbf{R}(\mathbf{z},t)\mathbf{J}^j(\mathbf{z}), \tag{2.5}$$

an expression which can be checked by direct multiplication with the metric in equation (2.2) and using the properties of the inverse Jacobian.

In addition to length, the curvature of line segments on the surface will be a useful measure of its geometry. For a curve on the surface tangent to the line segment connecting the points z_i and $z_i+\mathrm{d}z_i$, whose length is given in equation (2.1), the normal curvature is defined as the projection of the curvature of the curve in space ($\boldsymbol{\kappa}$) in the direction of the unit normal to the surface $\hat{\mathbf{n}}(\mathbf{z},t)$. Thus, we write

$$\kappa_N\equiv\boldsymbol{\kappa}\cdot\hat{\mathbf{n}}(\mathbf{z},t)=b_{ij}(\mathbf{z},t)\frac{\mathrm{d}z_i}{\mathrm{d}\ell}\frac{\mathrm{d}z_j}{\mathrm{d}\ell}\quad\text{and}\quad b_{ij}(\mathbf{z},t)\equiv\hat{\mathbf{n}}(\mathbf{z},t)\cdot\partial_k\partial_j\mathbf{X}(\mathbf{z},t), \tag{2.6}$$

where we defined the curvature tensor $b_{ij}(\mathbf{z},t)$. At a given point on the surface, the principal directions (directions of maximum and minimum normal curvatures) will be denoted as $\hat{\boldsymbol{\kappa}}_1(\mathbf{z},t)$ and $\hat{\boldsymbol{\kappa}}_2(\mathbf{z},t)$, with corresponding principal curvatures $\kappa_1(\mathbf{z},t)$ and $\kappa_2(\mathbf{z},t)$. From the principal curvatures, we can define the mean curvature $\mathcal{H}(\mathbf{z},t)\equiv[\kappa_1(\mathbf{z},t)+\kappa_2(\mathbf{z},t)]/2$ and Gaussian curvature $\mathcal{K}(\mathbf{z},t)\equiv\kappa_1(\mathbf{z},t)\kappa_2(\mathbf{z},t)$.

(b) Quasi-conformal flows

In this section, we consider the temporal rate of change of the quantities defined in the previous section. Since we assume that the trajectories of points $\mathbf{X}(\mathbf{z},t)$ are smooth in time, we define the

royalsocietypublishing.org/journal/rspa Proc. R. Soc. A 481: 20240527

velocity field as

$$\mathbf{V}(\mathbf{z},t) \equiv \dot{\mathbf{X}}(\mathbf{z},t) = \mathbf{V}_{\parallel}(\mathbf{z},t) + V_{\perp}(\mathbf{z},t)\,\hat{\mathbf{n}}(\mathbf{z},t), \quad \mathbf{z} \in \Omega_0, \tag{2.7}$$

where the over dot denotes a partial derivative with respect to time, and we have split the velocity into a component that is parallel to the surface, and a component along the surface normal $\hat{\mathbf{n}}(\mathbf{z},t)$. For points along the boundary of an open surface, we define the unit tangent vector to the boundary as $\hat{\boldsymbol{\tau}}(\mathbf{z},t)$ and write

$$\mathbf{V}_{\parallel}(\mathbf{z},t) = V_{\tau}(\mathbf{z},t)\hat{\boldsymbol{\tau}}(\mathbf{z},t) + V_{\partial}(\mathbf{z},t)\hat{\boldsymbol{n}}_{\partial}(\mathbf{z},t), \quad \hat{\boldsymbol{n}}_{\partial}(\mathbf{z},t) \equiv \hat{\boldsymbol{\tau}}(\mathbf{z},t) \times \hat{\mathbf{n}}(\mathbf{z},t),\ \mathbf{z} \in \partial\Omega_0. \tag{2.8}$$

The components of the velocity normal to the surface and normal to the boundary curve, $V_{\perp}(\mathbf{z},t)$ and $V_{\partial}(\mathbf{z},t)$, will be independent of registration since changing the registration $\mathbf{w}(\mathbf{z},t)$ at time t can only move the points parallel to the surface $\mathbf{X}(\mathbf{z},t)$ [24]. Therefore, the task of finding the registration $\mathbf{w}(\mathbf{z},t)$ given a flow of surfaces $\mathcal{X}(\mathbf{w},t)$ is equivalent to finding $\mathbf{V}_{\parallel}(\mathbf{z},t)$ given the perpendicular velocities $V_{\perp}(\mathbf{z},t)$ and $V_{\partial}(\mathbf{z},t)$.

Next, we consider the rate of change of the metric, or strain rate tensor, which can be found by taking the time derivative of equation (2.1),

$$\dot{g}_{ij}(\mathbf{z},t) = \partial_i \mathbf{X}(\mathbf{z},t) \cdot \partial_j \mathbf{V}(\mathbf{z},t) + \partial_i \mathbf{V}(\mathbf{z},t) \cdot \partial_j \mathbf{X}(\mathbf{z},t). \tag{2.9}$$

Note that, for a fixed registration, the coordinate $\mathbf{z}$ represents a Lagrangian (material) coordinate system and therefore the time derivative of the metric tensor is still a tensor with the same rank. The rate of change of the metric may also be found by taking the time derivative of the decomposition given in equation (2.2) and using the chain rule, yielding

$$\dot{g}_{ij} = \frac{\dot{\omega}}{\omega} g_{ij} + \frac{\dot{K}\,\omega}{K} \mathbf{J}_i^{\top}\,\mathbf{R}^{\top}\,\mathbf{E}\,\sigma_Z\,\mathbf{R}\,\mathbf{J}_j + \omega\,\mathbf{J}_i^{\top}[\dot{\mathbf{R}}^{\top}\mathbf{R}, \mathbf{R}^{\top}\,\mathbf{E}\,\mathbf{R}]\mathbf{J}_j,\ \sigma_Z \equiv \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \tag{2.10}$$

where $[\mathcal{O}_1, \mathcal{O}_2] \equiv \mathcal{O}_1\mathcal{O}_2 - \mathcal{O}_2\mathcal{O}_1$, for any matrices $\mathcal{O}_1$ and $\mathcal{O}_2$ and the dependence on $(\mathbf{z},t)$ has been suppressed for simplicity. The first term in this equation is due to the time derivative of the conformal factor $\omega(\mathbf{z},t)$. The second term comes from the derivative of the matrix $\mathbf{E}(z,t)$, which is given by

$$\dot{\mathbf{E}}(\mathbf{z},t) = \begin{pmatrix} \dot{K}(\mathbf{z},t) & 0 \\ 0 & \dfrac{-\dot{K}(\mathbf{z},t)}{K^2(\mathbf{z},t)} \end{pmatrix} = \frac{\dot{K}(\mathbf{z},t)}{K(\mathbf{z},t)} \begin{pmatrix} K(\mathbf{z},t) & 0 \\ 0 & \dfrac{-1}{K(\mathbf{z},t)} \end{pmatrix} = \frac{\dot{K}(\mathbf{z},t)\mathbf{E}(\mathbf{z},t)\sigma_Z}{K(\mathbf{z},t)}. \tag{2.11}$$

The third term in equation (2.10) comes from taking times derivatives of the rotation matrix $\mathbf{R}(\mathbf{z},t)$ and its transpose and exploiting the relation $\mathbf{R}(\mathbf{z},t)\mathbf{R}^{\top}(\mathbf{z},t) = \mathbf{I}_2$, to replace

$$\dot{\mathbf{R}}(\mathbf{z},t)\mathbf{R}^{\top}(\mathbf{z},t) = -\mathbf{R}(\mathbf{z},t)\dot{\mathbf{R}}^{\top}(\mathbf{z},t) \implies \dot{\mathbf{R}}(\mathbf{z},t) = -\mathbf{R}(\mathbf{z},t)\dot{\mathbf{R}}^{\top}(\mathbf{z},t)\mathbf{R}(\mathbf{z},t). \tag{2.12}$$

Lastly, using equation (2.3), we have that $\dot{\mathbf{R}}^{\top}(\mathbf{z},t)\mathbf{R}(\mathbf{z},t) = \dot{\theta}(\mathbf{z},t)\Sigma_2$, where $\Sigma_2 = \mathrm{i}\sigma_Y$ is the two-dimensional Levi–Civita symbol and σ_Y is a Pauli matrix. The three terms in equation (2.10) represent changes in the metric due to scalings, shears and rotations, respectively. Using equation (2.10), we calculate the dilation and shear rates as:

$$\mathcal{D}(\mathbf{z},t) \equiv \frac{1}{2}\mathrm{Tr}[\dot{g}_{ij}(\mathbf{z},t)] \equiv \frac{1}{2}g^{ij}(\mathbf{z},t)\dot{g}_{ij}(\mathbf{z},t) = \frac{\dot{\omega}(\mathbf{z},t)}{\omega(\mathbf{z}t)} \tag{2.13}$$

and

$$\mathcal{S}^2(\mathbf{z},t) \equiv \frac{1}{2}\mathrm{Tr}[\underline{\dot{g}}_{ij}(\mathbf{z},t)^2] = \left(\frac{\dot{K}(\mathbf{z},t)}{K(\mathbf{z},t)}\right)^2 + \frac{[K^2(\mathbf{z},t)-1]^2\,\dot{\theta}^2(\mathbf{z},t)}{K^2(\mathbf{z},t)}, \tag{2.14}$$

where $\underline{\dot{g}}_{ij}(\mathbf{z},t)$ is the traceless part of the strain rate tensor which, after multiplying by the inverse metric in equation (2.5), is given by (see electronic supplementary material, text, Section S1)

$$\underline{\dot{g}}^{i}_{j} \equiv g^{kj}\underline{\dot{g}}_{ik}(\mathbf{z},t) = \mathbf{J}^{i}_{\top}\mathbf{R}^{\top}\left[\frac{\dot{K}}{K}\sigma_Z + \dot{\theta}(\mathbf{E}^{-1}\Sigma_2\mathbf{E} - \Sigma_2)\right]\mathbf{R}\mathbf{J}_j, \tag{2.15}$$

where we suppressed the dependence on $(\mathbf{z}, t)$ in this equation for simplicity. Here, the two terms follow from the last two terms in equation (2.10), which are traceless. Squaring this expression and taking the trace leads to equation (2.14) as shown in the electronic supplementary material.

Note that $\mathcal{D}(\mathbf{z}, t)$—which may be positive or negative depending on whether the local area is expanding or contracting—and $\mathcal{S}(\mathbf{z}, t)$—which is always positive—are instantaneous quantities, depending only on the change in the metric at time t, while $\omega(\mathbf{z}, t)$, $K(\mathbf{z}, t)$ and $\theta(\mathbf{z}, t)$ depend on the initial surface, with $\omega(\mathbf{z}, 0) = 1$ and $K(\mathbf{z}, 0) = 1$. Note that $\mathcal{S}(\mathbf{z}, 0) = \dot{K}(\mathbf{z}, 0)$, which implies that $\mathcal{S}(\mathbf{z}, 0)$ is the rate of growth of the dilatation $K(\mathbf{z}, t)$ at $t = 0$. We define k-quasi-conformal flows as ones for which $\mathcal{S}(\mathbf{z}, t) \leq k$, which therefore can be thought of as a sequence of $(1 + k\,\mathrm{d}t)$-quasi-conformal maps. Using equation (2.14), it can be shown (see electronic supplementary material, Section S1) that the map generated by a k-quasi-conformal flow at $t = 1$ will satisfy $K < e^k$, consistent with the results of Reimann [19].

The quantities described so far in this section are intrinsic (depending on the metric) and do not quantify changes in embedding. For example, when unrolling a cylinder into a flat sheet, the strain rate tensor would vanish, $\dot{g}_{ij}(\mathbf{z}, t) = 0$. To account for changes in embedding that do not stretch the surface (isometric deformations), we follow Jermyn *et al.* [25] and define the bending strain as the rate of change of the unit normal, $\dot{\mathbf{n}}(\mathbf{z}, t) \equiv \partial \hat{\mathbf{n}}(\mathbf{z}, t)/\partial t$, to write a bending-like term in the cost function as

$$\mathcal{C}_{\text{bend}}[\mathbf{w}(\mathbf{z}, t)] = A_3 \int \mathrm{d}t\, \mathrm{d}\mathcal{A}\, \dot{\mathbf{n}}^2(\mathbf{z}, t), \tag{2.16}$$

where the area element is defined using the determinant of the metric, $g(\mathbf{z}, t)$, as $\mathrm{d}\mathcal{A} \equiv \mathrm{d}z_1\, \mathrm{d}z_2 \sqrt{g(\mathbf{z}, t)}$ and A_3 is a dimensionless constant.

(c) Flow rules

Next, we consider the criteria by which a quasi-conformal flow is uniquely selected from the many possible ones that fit the given data $\mathcal{X}(\mathbf{w}, t)$. In addition to common choices that focus on the parsimony of the mapping—such as minimizing an energy that measures the extent of the distortion and its spatial variation—we incorporate dynamical processes that generate and regulate morphogenetic flows into our framework. For concreteness, we consider shapes that may be generated from a combination of Ricci flows [26] and mean curvature flows [27], which may be relevant for growing leaves, beaks and bacterial cell shapes [28–31]. To accomplish this, we define a growth strain tensor which is the difference between the strain rate tensor $\dot{g}_{ij}$ and the predicted value based on the dynamical law, which can be written as an expansion involving terms proportional to the metric and curvature tensors [32],

$$G_{ij}(\mathbf{z}, t) = \frac{1}{2}\dot{g}_{ij}(\mathbf{z}, t) - \lambda_1\, g_{ij}(\mathbf{z}, t) - \lambda_2\, \mathcal{H}(\mathbf{z}, t) b_{ij}(\mathbf{z}, t) - \lambda_3\, \mathcal{K}(\mathbf{z}, t) g_{ij}(\mathbf{z}, t), \tag{2.17}$$

where $b_{ij}(\mathbf{z}, t)$ is the curvature tensor given in equation (2.6), $\mathcal{H}(\mathbf{z}, t)$ is the mean curvature, and $\mathcal{K}(\mathbf{z}, t)$ is the Gaussian curvature. Here, the dimensionless parameters $\lambda_\star$ are independent of $\mathbf{z}$ but may depend on time. If the flow $\mathcal{X}(\mathbf{w}, z)$ is in fact generated by the dynamical law, there exists a registration $\mathbf{w}(\mathbf{z}, t)$ such that $G_{ij}(\mathbf{z}, t) = 0$. For such flows, the λ_1 term gives exponential rate of expansion or contraction, the λ_2 term is related to the mean curvature flow, while the λ_3 term is related to the Ricci flow [32].

To find a quasi-conformal flow that is as close to satisfying $G_{ij}(\mathbf{z}, t) = 0$ as possible, we add a term quadratic in $G_{ij}(\mathbf{z}, t)$ to the cost function:

$$\mathcal{C}_{\text{viscous}}[\mathbf{w}(\mathbf{z}, t)] = \int \mathrm{d}t\, \mathrm{d}\mathcal{A}\, \mathcal{R}_1^{ijkl}(\mathbf{z}, t)\, G_{ij}(\mathbf{z}, t)\, G_{kl}(\mathbf{z}, t). \tag{2.18}$$

Assuming distortions at different locations and along different axes are penalized symmetrically—homogeneity and isotropy—the tensor $\mathcal{R}_1^{ijkl}(\mathbf{z}, t)$ must satisfy

$$\mathcal{R}_1^{ijkl}(\mathbf{z}, t) \equiv \left(A_1 - \frac{B_1}{2}\right) g^{ij}(\mathbf{z}, t) g^{kl}(\mathbf{z}, t) + B_1\, g^{ik}(\mathbf{z}, t) g^{jl}(\mathbf{z}, t), \tag{2.19}$$

where A_1 and B_1 are dimensionless constants—which are analogous to the (scaled) bulk and shear moduli, respectively, in elasticity theory, with G_{ij} playing the role of the stress tensor in this analogy [33]. Note that when setting $\lambda_1 = \lambda_2 = \lambda_3 = 0$, equation (2.18) measures the extent of the distortion, since $G_{ij}(\mathbf{z}, t)$ reduces to the strain rate $\dot{g}_{ij}(\mathbf{z}, t)/2$ in that case. Subtracting $B_1/2$ in equation (2.19) then ensures that only A_1 couples to the trace of $\dot{g}_{ij}(\mathbf{z}, t)$ since the integrand of equation (2.18) will be $(A_1\mathcal{D}^2 + 0.5B_1\mathcal{S}^2)$ when $\lambda_1 = \lambda_2 = \lambda_3 = 0$.

Furthermore, to obtain best-fitting quasi-conformal flows that are as uniform as possible, we will add the following term—analogous to the Dirichlet energy [34]—to the cost

$$\mathcal{C}_{\text{grad}}[\mathbf{w}(\mathbf{z}, t)] = \int \mathrm{d}t\, \mathrm{d}\mathcal{A}\, \mathcal{R}_2^{ijkl}(\mathbf{z}, t)\, \nabla_m G_{ij}(\mathbf{z}, t) \nabla^m G_{kl}(\mathbf{z}, t), \tag{2.20}$$

where all repeated indices summed over, ∇_i is the covariant derivative compatible with the metric $g^{ij}(\mathbf{z}, t)$, and $\nabla^i \equiv g^{ij}(\mathbf{z}, t)\nabla_j$ [35]. The tensor $\mathcal{R}_2^{ijkl}(\mathbf{z}, t)$ is defined analogously to equation (2.19),

$$\mathcal{R}_2^{ijkl}(\mathbf{z}, t) \equiv \left(A_2 - \frac{B_2}{2}\right) g^{ij}(\mathbf{z}, t) g^{kl}(\mathbf{z}, t) + B_2\, g^{ik}(\mathbf{z}, t) g^{jl}(\mathbf{z}, t), \tag{2.21}$$

where A_2 and B_2 are constants.

(d) Imposing (soft) constraints

Since our goal is to find an optimal quasi-conformal flow that coincides with the given data $\mathcal{X}(\mathbf{w}, t)$, we need to impose constraints that fix the computed flow to the data. As mentioned after equation (2.8), this amounts to fixing the components of the velocity field that are normal to the surface and its boundary curve, while allowing the tangent components of the velocity field to vary. We constrain the normal components by adding the following terms to the cost function being minimized:

$$\mathcal{C}_{\text{normal}}\,[\mathbf{w}(\mathbf{z}, t)] = C_n \int_{\mathcal{X}} \mathrm{d}t\, \mathrm{d}\mathcal{A}[V_\perp(\mathbf{z}, t) - \bar{V}_\perp(\mathbf{z}, t)]^2 \tag{2.22}$$

and

$$\mathcal{C}_{\text{boundary}}\,[\mathbf{w}(\mathbf{z}, t)] = C_b \int_{\partial\mathcal{X}} \mathrm{d}t\, \mathrm{d}s[V_\partial(\mathbf{z}, t) - \bar{V}_\partial(\mathbf{z}, t)]^2, \tag{2.23}$$

where $\bar{V}_\perp(\mathbf{z}, t)$ and $\bar{V}_\partial(\mathbf{z}, t)$ are the registration independent normal displacements that are prescribed by the given data, and C_n, C_b are dimensionless constants. Note that the first integral is over the area of the surface, while the second is over its boundary curve.

Landmarks are points for which we know the trajectory *a priori*, either from experiments or other considerations [36]. If we take $\mathbf{z}_\star$ as the coordinate of a given landmark point, whose velocity is given by $\bar{\mathbf{V}}_\star(t)$, we account for landmark constraints by adding the following to the cost function:

$$\mathcal{C}_{\text{landmark}}\,[\mathbf{w}(\mathbf{z}, t)] = C_L \int \mathrm{d}t \sum_\star [\mathbf{V}(\mathbf{z}_\star, t) - \bar{\mathbf{V}}_\star(t)]^2, \tag{2.24}$$

where C_L is a dimensionless constant and we sum over all landmarks.

We are now in a position to define the cost function that will be used for the rest of the paper to be

$$\mathcal{C}_{\text{total}}\,[\mathbf{w}(\mathbf{z}, t)] = \mathcal{C}_{\text{viscous}} + \mathcal{C}_{\text{grad}} + \mathcal{C}_{\text{bend}} + \mathcal{C}_{\text{normal}} + \mathcal{C}_{\text{boundary}} + \mathcal{C}_{\text{landmark}}, \tag{2.25}$$

where the individual terms are defined in equations (2.16)–(2.20) and the dimensionless parameters we choose are A_1, B_1, A_2, A_3, and the constraints will be enforced by choosing $C_n = C_b = C_L = 10^5$ for all models (see table 1 for a summary of the terms and equations).

In general, the choice of parameters will depend on the context of a particular application. To demonstrate our approach, we will consider four different choices of parameters in this paper (table 2).

— Almost-conformal ($B_1 = 1$, all other parameters zero), which generates an as-conformal-as-possible fit to the flow;

Table 1. The terms and equations in the models for quasi-conformal flows. The growth strain, defined in equation (2.17), is the difference between the strain rate tensor $\dot{g}_{ij}(\mathbf{z}, t)$ and the value predicted from the geometric flow, where $b_{ij}(\mathbf{z}, t)$ is the curvature tensor given in equation (2.6), $\mathcal{H}(\mathbf{z}, t)$ is the mean curvature and $\mathcal{K}(\mathbf{z}, t)$ is the Gaussian curvature. $\mathcal{C}_{\text{viscous}}$ penalizes non-zero growth strain tensor $G_{ij}(\mathbf{z}, t)$ given in equation (2.17), while $\mathcal{C}_{\text{grad}}$ penalizes its spatial variations. The bending energy $\mathcal{C}_{\text{bend}}$ penalizes changes in the shape that do not stretch the surface and the last three costs enforce the constraints from data. In practice, the constraints are enforced by choosing $C_n = C_b = C_L = 10^5$.

terms	equations
growth strain (equation (2.17))	$G_{ij}(\mathbf{z}, t) = \frac{1}{2}\dot{g}_{ij}(\mathbf{z}, t) - \lambda_1 g_{ij}(\mathbf{z}, t) - \lambda_2 \mathcal{H}(\mathbf{z}, t) b_{ij}(\mathbf{z}, t) - \lambda_3 \mathcal{K}(\mathbf{z}, t) g_{ij}(\mathbf{z}, t)$
viscous cost (equation (2.18))	$\mathcal{C}_{\text{viscous}}[\mathbf{w}(\mathbf{z}, t)] = \int dt\, d\mathcal{A}\left(\left(A_1 - \frac{B_1}{2}\right) \text{Tr}[G_{ij}]^2 + B_1 \text{Tr}[G_{ij}^2] \right)$
spatial gradient cost (equation (2.20))	$\mathcal{C}_{\text{grad}}[\mathbf{w}(\mathbf{z}, t)] = \int dt\, d\mathcal{A}\left(\left(A_2 - \frac{B_2}{2}\right) \text{Tr}[\nabla_k G_{ij}]^2 + B_2 \text{Tr}[\nabla_k G_{ij}^2] \right)$
bending cost (equation (2.16))	$\mathcal{C}_{\text{bend}}[\mathbf{w}(\mathbf{z}, t)] = A_3 \int dt\, d\mathcal{A}\, \vec{\mathbf{n}}^2(\mathbf{z}, t)$
enforcing normal displacement (equation (2.22))	$\mathcal{C}_{\text{normal}}[\mathbf{w}(\mathbf{z}, t)] = C_n \int_{\mathcal{X}_t} dt\, d\mathcal{A} [V_\perp(\mathbf{z}, t) - \bar{V}_\perp(\mathbf{z}, t)]^2$
enforcing boundary displacements (equation (2.23))	$\mathcal{C}_{\text{boundary}}[\mathbf{w}(\mathbf{z}, t)] = C_b \int_{\partial \mathcal{X}_t} dt\, ds [V_\partial(\mathbf{z}, t) - \bar{V}_\partial(\mathbf{z}, t)]^2$
enforcing landmark velocities (equation (2.24))	$\mathcal{C}_{\text{landmark}}[\mathbf{w}(\mathbf{z}, t)] = C_L \int dt \sum_\star [\mathbf{V}(\mathbf{z}_\star, t) - \bar{\mathbf{V}}_\star(t)]^2$
overall cost function (equation (2.25))	$\mathcal{C}[\mathbf{w}(\mathbf{z}, t)] = \mathcal{C}_{\text{viscous}} + \mathcal{C}_{\text{grad}} + \mathcal{C}_{\text{bend}} + \mathcal{C}_{\text{normal}} + \mathcal{C}_{\text{boundary}} + \mathcal{C}_{\text{landmark}}$

Table 2. The parameter choices for different quasi-conformal flow models. The almost-conformal model ($B_1 = 1$, and all other parameters zero) generates a fit to the flow that is as conformal as possible. The viscous model ($A_1 = B_1 = A_3 = 1$, and all other parameters zero) generates a minimal distortion fit to the flow. The almost-uniform model ($A_2 = B_2 = 1$, all other parameters zero) minimizes the spatial variation of the flow in space. The geometric model ($A_1 = B_1 = 1$, solve for $\lambda_1, \lambda_2, \lambda_3 \neq 0$, and all other parameters zero) fits the flow to a dynamical equation, which is a geometric flow in the example considered in this paper.

parameters / models	A_1	B_1	A_2	B_2	A_3	$\lambda_1, \lambda_2, \lambda_3$
almost-conformal	0	1	0	0	0	0
viscous	1	1	0	0	1	0
almost-uniform	0	0	1	1	0	0
geometric	1	1	0	0	0	fitted

— viscous ($A_1 = B_1 = A_3 = 1$, and all other parameters zero), which generates a minimal distortion fit to the flow;
— almost-uniform ($A_2 = B_2 = 1$, and all other parameters zero), which minimizes the variation of the flow in space; and
— geometric ($A_1 = B_1 = 1$, solve for $\lambda_1, \lambda_2, \lambda_3 \neq 0$, and all other parameters zero), which fits the flow to a dynamical equation, which in this paper is a geometric flow parametrized by λ_1, λ_2 and λ_3 (which are solved for during our fitting procedure).

Note that (C_n, C_b, C_L) are much larger than other model parameters since they enforce constraints from the registration-independent component of the velocity field that can be extracted from the given data. The choice of parameters corresponding to the viscous model minimizes the quadratic viscous term equation (2.18) along with the bending term (2.16) subject to the constraints. Under this choice of parameters (including $\lambda_i = 0$), we look for a velocity field that makes the rate of change of the metric (strain rate) and the normal (bending rate) as small as possible, which is why we refer to this as a viscous fit to the data. On the other hand, if we only

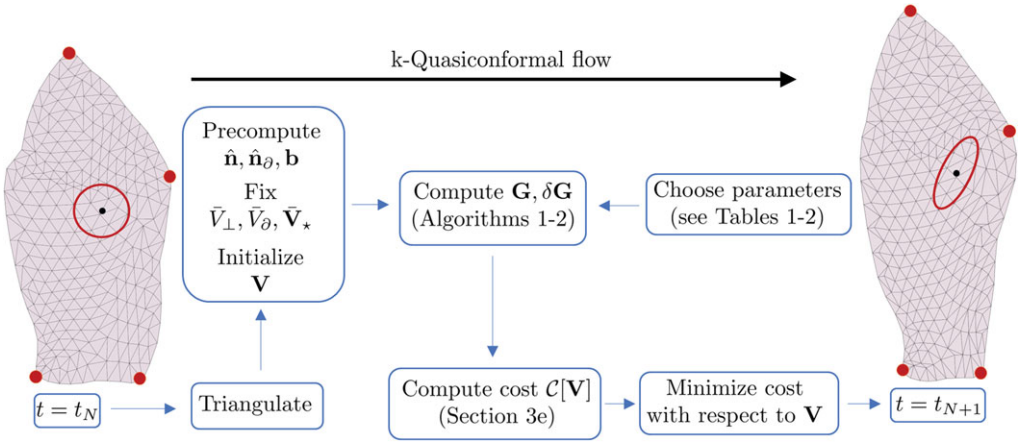

Figure 2. Surface registration with optimal quasi-conformal flows. This figure illustrates the workflow in our approach: We first choose a model type from those shown in table 2, which gives us a fixed choice of parameters in the cost function. The flow is then broken up into discrete-time steps t_N, triangulating each surface and pre-computing geometric quantities that do not depend on the variable being optimized (the velocity field $\mathbf{V}$), such as the unit normal to the surface $\hat{\mathbf{n}}$, unit normal to the boundary curve $\hat{\mathbf{n}}_\partial$, and the curvature tensor $\mathbf{b}$. The component of the velocity normal to the surface $\bar{V}_\perp$ and boundary curve $\bar{V}_\partial$, in addition to the velocity of the landmark points $\bar{\mathbf{V}}_\star$ are fixed from the input data (see also figure 3 for an illustration of these quantities). The velocity $\mathbf{V}$ is calculated for each vertex at time $t = t_N$ before moving to the next time step.

want an as-conformal-as-possible fit to the flow, then we set $A_1 = A_3 = 0$ since in that case we only need to minimize the values of $\mathcal{S}(\mathbf{z}, t)$, while $\mathcal{D}(\mathbf{z}, t)$ and $\dot{\mathbf{n}}(\mathbf{z}, t)$ can have arbitrary values. Lastly, the almost-uniform case minimizes the spatial variations of $\mathbf{G}(\mathbf{z}, t)$ given by equation (2.20).

3. Computational procedure

In this section, we describe how we compute and minimize the cost function defined in equation (2.25). We start by pre-processing input data and determining mesh connectivity and curvature, which are quantities that do not vary throughout the minimization process. We then compute the registration-dependent quantities that characterize the distortion of the shape data over time and across space. Lastly, we will describe how we minimize the cost function to obtain an optimal velocity field and post-process the results (figure 2).

(a) Data preparation and mesh pre-processing

In this section, we describe how we pre-processed the various datasets to obtain a sequence of triangulations $\mathcal{T}(t_N) \equiv \{\mathcal{F}, \mathcal{E}, \mathcal{V}(t_N)\}$ at (equally spaced) time points t_N, with $1 \le N \le N_{\max} = 30$. The connectivity of each triangulation, which is fixed over time, is defined by the faces $f_M \in \mathcal{F}$ (a list of triplets of indices, where $1 \le M \le M_{\max}$) and edges $e_L \in \mathcal{E}$ (a list of pairs of indices, where $1 \le L \le L_{\max}$). The list of vertex positions, which vary across time, is denoted as $\mathcal{V}(t_N)$.

To study the growth of two-dimensional insect wings, we considered the *Manduca sexta* (tobacco hawk moth) and *Junonia coenia* (buckeye butterfly) wing data from Nijhout *et al*. [37]. For each species, two-dimensional triangular meshes were constructed based on the wing images at the larval, prepupal, pupal and adult stages using the `distmesh` package [38]. An initial registration between each pair of consecutive stages was then computed using a variation of the mapping method in [14], where the boundary correspondence was obtained using the curvature-guided matching method in [14] and the interior correspondence was obtained using the smooth quasi-conformal mapping formulation in [9]. After getting the initial registration between the four stages above, we then applied the Piecewise Cubic Hermite Interpolating Polynomial (`pchip`)

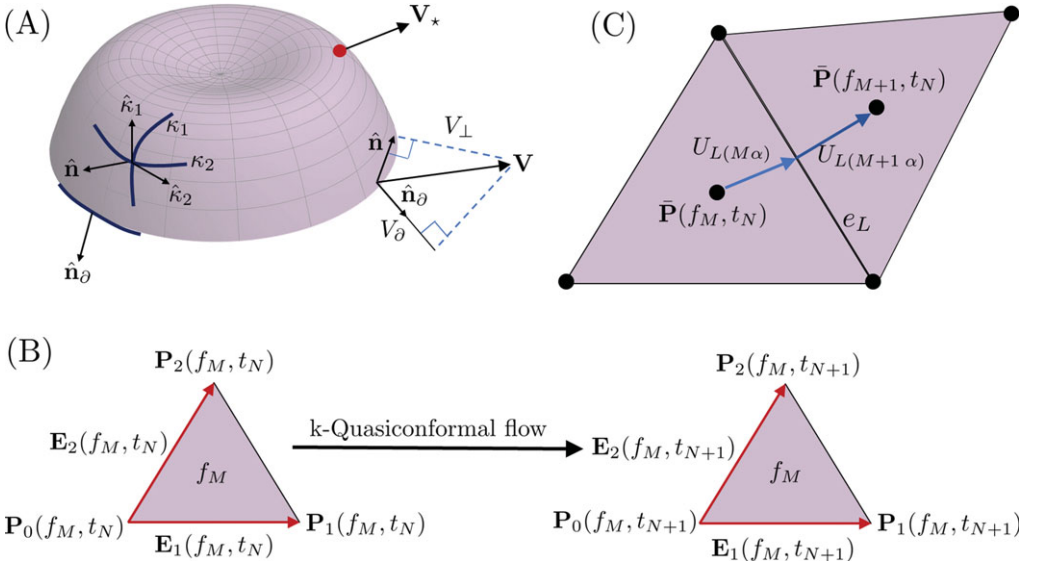

Figure 3. Geometric quantities used in our computational procedure. (A) Illustrates the surface normal $\hat{\mathbf{n}}$, in addition to the principal directions $(\hat{\kappa}_1, \hat{\kappa}_2)$ and curvatures (κ_1, κ_2) at the same point. The vector $\hat{\mathbf{n}}_\partial$ is normal to both the boundary curve and $\hat{\mathbf{n}}$. The corresponding normal velocity components $V_\perp$ and V_∂, defined by equations (2.7) and (2.8), are illustrated on the lower right. The figure also illustrates the velocity $\mathbf{V}_\star$ of the landmark point shown in red. (B) Shows the vertex positions $\mathbf{P}_i(f_M, t_N)$, $i = 0, 1, 2$ for the triangle f_M at consecutive time points t_N and t_{N+1}. The face index satisfies $1 \leq M \leq M_{\max}$ and the time index satisfies $1 \leq N \leq N_{\max}$. The edge vectors $\mathbf{E}_i(f_M, t_N)$ shown in red are defined in equation (3.2). (C) The centroid positions of each triangle, $\bar{\mathbf{P}}(f_M, t_N)$ which are averages of the vertex positions of the triangle are indicated for two consecutive triangles f_M and f_{M+1}. The displacement vectors starting at the centroid of triangle M and ending at the midpoint of edge e_L are indicated by $U_{L(M\alpha)}$, where $\alpha \in \{1, 2, 3\}$ indicates the component of the vector in $\mathbb{R}^3$. The parenthesis $(M\alpha)$ represents a multi-index, thus expressing $U_{L(M\alpha)}$ as an $L_{\max} \times 3 * M_{\max}$ matrix.

function in MATLAB to get a flow of wing shapes with smooth growth over 30 time steps, which then served as the starting point for our subsequent analysis.

To study the growth of three-dimensional plant leaves using the proposed method, we considered the *persea americana* (avocado) leaf dataset from Derr & Bastien [39]. Each leaf scan consists of about 100 tracked points in $\mathbb{R}^3$. To create a triangular mesh for each scan, we first used the principal component analysis (`pca`) function in MATLAB to project the points onto the two-dimensional plane and constructed a two-dimensional Delaunay triangulation using the MATLAB `delaunay` function, with sharp triangles at the boundary removed. The planar triangulation then induced a triangulation on the set of three-dimensional points, resulting in a triangulated leaf mesh. Next, we further applied the Taubin smoothing [40] and the Loop subdivision [41] to obtain a smooth, refined mesh with about 400 vertices.

(b) Computation of geometric quantities

After pre-processing the meshes, we consider geometric quantities that are independent of the registration—gauge invariant quantities as described in [24]—which allows us to pre-compute these quantities, thus reducing the time per function call during the optimization procedure. These quantities include (figures 2 and 3) the normal vector to the surface $\hat{\mathbf{n}}$, the normal vector to the boundary curve $\hat{\mathbf{n}}_\partial$, and the principal curvatures κ_1 and κ_2 along with the corresponding principal directions $\hat{\kappa}_1$ and $\hat{\kappa}_2$.

The boundary normal vector $\hat{\mathbf{n}}_\partial$ was calculated using the MATLAB `LineNormals2D` package [42]. For surfaces in $\mathbb{R}^3$, we computed the surface normal $\hat{\mathbf{n}}$, principal curvatures

κ_1, κ_2, mean curvature $\mathcal{H} = (\kappa_1 + \kappa_2)/2$ and Gaussian curvature $\mathcal{K} = \kappa_1\kappa_2$ using the MATLAB `patchcurvature` package [43]. The tangent and normal vectors of the boundaries of open surfaces were calculated using the MATLAB `frenet_robust` package [44].

(c) Discretizing the growth and bending strains

In this section, we describe how we calculate the bending strain, defined in equation (2.16), and the growth strain tensor, defined in equation (2.17), as summarized in algorithm 1 (see also electronic supplementary material, Section S2 for a concrete example of the calculation).

Algorithm 1. Discretize the growth and bending strains. (*See equations (2.16)–(2.17)*).

Input: Triangulations $\mathcal{T}(t_N)$. (*See* §3a.)
Output: The discretized growth $\mathbf{G}(f_M, t_N)$ and bending $\dot{\mathbf{n}}(f_M, t_N)$ strains.
for $1 \le N < N_{\max}$ **do**
  for $1 \le M < M_{\max}$ **do**
    ⋆ Compute basis edges $\mathbf{E}_i(f_M, t_N)$, and $\mathbf{E}_i(f_M, t_{N+1})$. (*See equation (3.2)*).
    ⋆ Compute the discrete metric $\mathbf{Q}_{\|}(f_M, t_N)$. (*See equations (3.6) and (3.7)*).
    ⋆ Compute the velocity gradient, $\delta_i \mathbf{V}(f_M, t_N)$. (*See equation (3.5)*).
    ⋆ Compute the discretized strain rate $\mathcal{J}(f_M, t_N)$. (*See equation (3.8)*).
    ⋆ Compute the discretized growth strain tensor $\mathbf{G}(f_M, t_N)$. (*See equation (3.10)*).
    ⋆ Compute the discretized bending strain $\dot{\mathbf{n}}(f_M, t_N)$. (*See equation (3.11)*).
  end for
end for

For a given triangle at time t_N, labelled with face index $f_M \in \mathcal{F}$ and spanned by the three vertices with positions $\mathbf{P}_0(f_M, t_N)$, $\mathbf{P}_1(f_M, t_N)$, and $\mathbf{P}_2(f_M, t_N)$, the triangle centroid will approximate the smooth surface at the point $\mathbf{z} = \mathbf{z}_M$ so that (figure 3B)

$$\mathbf{X}(\mathbf{z}_M, t_N) \approx \bar{\mathbf{P}}(f_M, t_N) \equiv \frac{\mathbf{P}_0(f_M, t_N) + \mathbf{P}_1(f_M, t_N) + \mathbf{P}_2(f_M, t_N)}{3}. \tag{3.1}$$

At the same point ($\mathbf{z} = \mathbf{z}_M$), the tangent plane will be spanned by the two edge vectors (shown in red in figure 3C):

$$\mathbf{E}_1(f_M, t_N) \equiv \mathbf{P}_1(f_M, t_N) - \mathbf{P}_0(f_M, t_N) \quad \text{and} \quad \mathbf{E}_2(f_M, t_N) \equiv \mathbf{P}_2(f_M, t_N) - \mathbf{P}_0(f_M, t_N), \tag{3.2}$$

which point along the edges of the triangle and correspond to the tangent vectors, $\partial_i \mathbf{X}(\mathbf{z}_M, t_N) \sim \mathbf{E}_i(f_M, t_N)$.[1] To compute the strain rate given in equation (2.9), we approximate the velocities of the vertices of the triangle (labelled by index $i = 0, 1, 2$), along with its centroid velocity, by

$$\mathbf{V}_i(f_M, t_N) \equiv \frac{\mathbf{P}_i(f_M, t_{N+1}) - \mathbf{P}_i(f_M, t_N)}{t_{N+1} - t_N} \tag{3.3}$$

and

$$\bar{\mathbf{V}}(f_M, t_N) \equiv \frac{\mathbf{V}_0(f_M, t_N) + \mathbf{V}_1(f_M, t_N) + \mathbf{V}_2(f_M, t_N)}{3}, \tag{3.4}$$

where the second equation defined the centroid velocity for face f_M. To calculate the strain rate tensor, we need the spatial gradient of the velocity field, $\partial_i \mathbf{V}(\mathbf{z}_M, t_N)$, which we approximate by

[1]This will be an (approximate) equality in local coordinate system $\mathbf{z}$ that parametrized a point on the triangle using $\mathbf{X}(\mathbf{z}_M, t_N) = \mathbf{P}_0(f_M, t_N) + z_1\mathbf{E}_1(f_M, t_N) + z_2\mathbf{E}_2(f_M, t_N)$, where $0 \le z_1, z_2 \le 1$ and $z_1 + z_2 \le 1$ (see electronic supplementary material, Section S2 for details).

its finite difference analogue,

$$\partial_i \mathbf{V}(\mathbf{z}_M, t_N) \sim \delta_i \mathbf{V}(f_M, t_N) \equiv \frac{\mathbf{E}_i(f_M, t_{N+1}) - \mathbf{E}_i(f_M, t_N)}{t_{N+1} - t_N}, \quad i = 1, 2, \tag{3.5}$$

where δ_i indicates a spatial finite difference. While the two vectors $\mathbf{E}_1(f_M, t_N)$ and $\mathbf{E}_2(f_M, t_N)$ span the plane tangent to the triangle labelled by index f_M, it will be convenient as in the continuum case to introduce the dual basis, denoted as $\mathbf{E}^i(f_M, t_N) \sim \partial^i \mathbf{X}(\mathbf{z}_M, t_N)$, which are defined as the (row) vectors in $\mathbb{R}^3$ satisfying

$$\mathbf{E}^i(f_M, t_N) \cdot \mathbf{E}_j(f_M, t_N) = \delta^i_j \quad \text{and} \quad \mathbf{E}_i(f_M, t_N) \otimes \mathbf{E}^i(f_M, t_N) = \mathbf{Q}_\parallel(f_M, t_N), \tag{3.6}$$

where $\otimes$ gives a direct product between the two vectors in $\mathbb{R}^3$ and $\mathbf{Q}_\parallel(f_M, t_N)$ is a (3×3) projection matrix onto the plane of the triangle indexed by f_M and represents an approximation of the metric tensor in the orthonormal frame spanning the plane of the triangle:

$$\mathcal{T}^{ij}(\mathbf{z}_M, t_N)\, g_{ij}(\mathbf{z}_M, t_N) \approx \mathbf{Q}_\parallel(f_M, t_N) \quad \text{and} \quad \mathcal{T}^{ij} \equiv [\partial^i \mathbf{X}(\mathbf{z}_M, t_N) \otimes \partial^j \mathbf{X}(\mathbf{z}_M, t_N)]. \tag{3.7}$$

Since it is composed of products of the Jacobian $\partial^i \mathbf{X}(\mathbf{z}_M, t_N)$, the factor $\mathcal{T}^{ij}(\mathbf{z}_M, t_N)$ transforms rank-two tensors from the local coordinate basis corresponding to the (arbitrary) coordinate system $\mathbf{z}$ to a tensor in the embedding Cartesian space, $\mathbb{R}^3$—this will turn out to be very useful when computing the gradient cost in the next section. The fact that the two expressions for $\mathbf{Q}_\parallel(f_M, t_N)$ given in equations (3.6) and (3.7) are the same can be checked by multiplying $\mathbf{Q}_\parallel(f_M, t_N)$ with an arbitrary vector (field) in $\mathbb{R}^3$ and showing that in both cases it functions as a projection operator onto the tangent plane to the surface (or triangle in the discrete setting).

Using equations (3.5)–(3.7) and (2.9), we define the discrete strain rate using the time derivative (or finite difference) of $\mathbf{Q}_\parallel$,

$$\begin{aligned} &\mathcal{T}^{ij}(\mathbf{z}_M, t_N)\, \dot{g}_{ij}(\mathbf{z}_M, t_N) \approx \mathcal{J}(f_M, t_N) \\ &\quad \equiv \mathbf{Q}_\parallel(f_M, t_N)[\delta_i \mathbf{V}(f_M, t_N) \otimes \mathbf{E}^i(f_M, t_N) + \mathbf{E}^i(f_M, t_N) \otimes \delta_i \mathbf{V}(f_M, t_N)]\mathbf{Q}_\parallel(f_M, t_N), \end{aligned} \tag{3.8}$$

where the approximate equality follows by evaluating both expressions on the left and right of $\mathcal{J}(f_M, t_N)$—which is a scalar with respect to the coordinate transformations on the surface and therefore can be evaluated in any coordinate system without loss of generality—in the particular coordinate system that parametrizes the triangle as $\mathbf{X}(\mathbf{z}_M, t_N) = \mathbf{P}_0(f_M, t_N) + z_1\ \mathbf{E}_1(f_M, t_N) + z_2\ \mathbf{E}_2(f_M, t_N)$, as shown in electronic supplementary material, Section S2.

The last ingredient we need to compute the growth strain, given in equation (2.17), will be the curvature tensor. Using the previously computed principal curvatures and directions, the curvature tensor is written as

$$\mathcal{T}^{ij}(\mathbf{z}_M, t_N)\, b_{ij}(\mathbf{z}_M, t_N) \approx \mathbf{B}(f_M, t_N) \equiv \kappa_1\, \hat{\boldsymbol{\kappa}}_1 \otimes \hat{\boldsymbol{\kappa}}_1 + \kappa_2\, \hat{\boldsymbol{\kappa}}_2 \otimes \hat{\boldsymbol{\kappa}}_2, \tag{3.9}$$

where the dependence of the principal curvatures and directions on (f_M, t_N) has been suppressed to simplify the expression. The equivalence of the left- and right-hand sides of equation (3.9) follows since both expressions describe a symmetric matrix whose eigenvalues are $(0, \kappa_1, \kappa_2)$ in the normal and two principal directions, respectively.

Next, we can calculate the discrete growth strain, which following equation (2.17) is given by

$$\begin{aligned} &\mathcal{T}^{ij}(\mathbf{z}_M, t_N)\, G_{ij}(\mathbf{z}_M, t_N) \approx \mathbf{G}(f_M, t_N) \\ &\quad \equiv \mathcal{J}(f_M, t_N) - \lambda_1\, \mathbf{Q}_\parallel(f_M, t_N) - \lambda_2\, \mathcal{H}(f_M, t_N)\, \mathbf{B}(f_M, t_N) - \lambda_3\, \mathcal{K}(f, t_N)\mathbf{Q}_\parallel(f_M, t_N). \end{aligned} \tag{3.10}$$

In addition to the growth strain, we will need the bending strain, which is the time derivative of the normal vector used in equation (2.16). We calculate this using the (finite) difference between

the normal vector to each triangle at two consecutive time points:

$$\hat{\mathbf{n}}(f_M, t_N) \equiv \frac{\mathbf{E}_1(f_M, t_N) \times \mathbf{E}_2(f_M, t_N)}{||\mathbf{E}_1(f_M, t_N) \times \mathbf{E}_2(f_M, t_N)||} \quad \text{and} \quad \dot{\mathbf{n}}(f_M, t_N) \equiv \frac{\hat{\mathbf{n}}(f_M, t_{N+1}) - \hat{\mathbf{n}}(f_M, t_N)}{t_{N+1} - t_N}. \tag{3.11}$$

The growth and bending strains calculated in this section will be used in computing the total cost below, which also includes the gradient (Dirichlet energy) contributions that we turn to next.

(d) Discretizing the growth strain gradient

Having calculated the growth strain tensors $\boldsymbol{G}(f_M, t_N)$ in equation (3.10), we proceed to estimate the discrete gradient, denoted as $\delta_\alpha \boldsymbol{G}(f_M, t_N)$, where $\alpha \in \{1, 2, 3\}$ indicates the direction of the derivative in $\mathbb{R}^3$. Our strategy will be to compute the gradient of $\boldsymbol{G}(f_M, t_N)$ as tensors in the embedding space ($\mathbb{R}^3$) and exploiting the relation (see electronic supplementary material, Section S1 for a derivation) between the covariant derivative of a tensor and the gradient in the Euclidean space $\mathbb{R}^3$ [45–47]. The steps are summarized in algorithm 2.

Algorithm 2. Discretizing the growth strain gradient. (*See equation (2.20).*)

Input: Triangulations $\mathcal{T}(t_N)$ and the growth strains $\mathbf{G}(f_M, t_N)$. (*See algorithm 1.*)
Output: The discrete growth strain $\delta\mathbf{G}(f_M, t_N)$. (*See equation (3.15).*)
for $1 \leq N < N_{\max}$ **do**
  for all faces f and all edges e in the triangulation $\mathcal{T}(t_N)$ **do**
    ⋆ Calculate the finite difference matrix Δ_{LM}. (*See equation (3.12).*)
    ⋆ Calculate the face-displacement vectors $U_{L(M\alpha)}$. (*See equation (3.12).*)
    ⋆ Calculate the pseudoinverse $U^{+}_{(M\alpha)L}$. (*See equation (3.13).*)
    ⋆ Find the gradient stiffness matrix, $\mathcal{M}_{f_1 f_2}$. (*See equation (3.14).*)
    ⋆ Find the discrete growth strain $\delta\mathbf{G}(f_M, t_N)$. (*See equation (3.15).*)
  end for
end for

To estimate the gradient $\nabla_\alpha \mathbf{G}(\mathbf{z}_M, t)$, we consider a pair of triangles (f_M and f_{M+1} in figure 3C) connected by edge e_L and denote the vector pointing from the centroid of triangle f_M towards the midpoint of the edge as $U_{L(M\alpha)}$. On the other hand, for the second triangle in the pair, we have $U_{L(M+1\,\alpha)}$ will point from the midpoint of the edge e_L towards its centroid, as shown in figure 3C. Lastly, when f_M is a triangle not contained in the pair connected by edge e_L, $U_{L(M+1\,\alpha)} = 0$. We also define the finite difference matrix, Δ_{LM}, which equals $\Delta_{LM} = -1$ for the first triangle in the pair connected by edge e_L, $\Delta_{LM+1} = 1$ for the other, and zero otherwise. Therefore, using a finite difference approximation, we have the relation

$$\sum_{(M\alpha)} U_{L(M\alpha)}\, \nabla_\alpha \boldsymbol{G}(f_M, t_N) \approx \sum_{M} \Delta_{LM}\, \boldsymbol{G}(f_M, t_N), \tag{3.12}$$

which expresses the fact that, for each edge e_L, dotting the gradient with the two displacement vectors connecting the triangle centroids (figure 3C) should give the difference $\boldsymbol{G}(f_{M+1}, t_N) - \boldsymbol{G}(f_M, t_N)$. Taking $U_{L(M\alpha)}$ as a matrix whose rows are labelled by the edges L and columns are labelled by the multi-index $(M\alpha)$, the above linear equation can be solved to give the gradient of the growth strain tensor as

$$\nabla_\alpha \boldsymbol{G}(\mathbf{z}_M, t_N) \approx \delta_\alpha \boldsymbol{G}(f_M, t_N) \equiv \sum_{M_1, L} U^{+}_{(M\alpha)L}\, \Delta_{LM_1}\, \boldsymbol{G}(f_{M_1}, t_N), \tag{3.13}$$

where $U^{+}_{(M\alpha)L}$ is the pseudoinverse of the matrix $U_{L(M\alpha)}$. This allows us to compute the gradient squared term in the cost function using the following matrix:

$$\mathcal{M}^2_{M_1 M_2} \equiv \sum_{L_1, L_2, (M\alpha)} \delta\mathcal{A}_M \Delta_{L_1 M_1}\, U^{+}_{(M\alpha)L_1}\, U^{+}_{(M\alpha)L_2}\, \Delta_{L_2 M_2}, \tag{3.14}$$

where $\delta\mathcal{A}_M$ is the area of triangle f_M. The matrix defined by equation (3.14)) is positive definite since it has the form of a matrix times its transpose, and therefore we may formally write its square root as $\mathcal{M}_{M_1M_2}$. The advantage of introducing the matrix $\mathcal{M}_{M_1M_2}$ is that it depends on the triangulation $\mathcal{T}(t_N)$, and can be precomputed so that during optimization, calculating the discrete growth strain gradient is done by the following matrix multiplication (see electronic supplementary material, Section S1),

$$\delta\boldsymbol{G}(f_M, t_N) \equiv \frac{1}{\delta\mathcal{A}_M^{1/2}} \sum_{M_1} \mathcal{M}_{MM_1}\, \boldsymbol{G}(f_{M_1}, t_N). \tag{3.15}$$

Thus, during optimization, only the growth strain $\boldsymbol{G}(f_M, t_N)$ needs to be computed, and the gradient term follows my matrix multiplication, which enables significantly faster implementation of the algorithm—through vectorization [48]—using the packages JAX and numpy in python [49,50]. Once obtained, the quantity $\delta\boldsymbol{G}(f_M, t_N)$ is used to compute the gradient contribution to the cost function in a manner similar to that used in computing the viscous term, as we show next (see also electronic supplementary material, Section S1).

(e) Discretizing and minimizing the total cost

Having calculated the growth strain along with its gradients, we now proceed to calculate the total cost function defined in equation (2.25) by adding up all the different contributions. Using equation (3.11), the bending contribution to the cost is

$$\mathcal{C}_{\text{bend}} = A_3 \sum_{M=1}^{M_{\max}} \sum_{N=1}^{N_{\max}} \delta t_N\, \delta\mathcal{A}_M [\hat{\mathbf{n}}(f_M, t_N) \cdot \dot{\mathbf{n}}(f_M, t_N)], \tag{3.16}$$

where $\delta\mathcal{A}_M$ is the area of the triangle f_M at time $t = t_N$, $\delta t_N \equiv t_{N+1} - t_N$, and we sum over all triangles and time steps. We compute the contribution of triangle f_M, at time t_N, to the viscous cost defined in equation (2.18) through

$$\mathcal{C}_{\text{viscous}} = \sum_{M=1}^{M_{\max}} \sum_{N=1}^{N_{\max}} \delta t_N\, \delta\mathcal{A}_M \left(\left(A_1 - \frac{B_1}{2} \right) \text{Tr}[\mathbf{G}(f_M, t_N)]^2 + B_1 \text{Tr}[\mathbf{G}(f_M, t_N)^2] \right). \tag{3.17}$$

The gradient cost defined in equation (2.20) is calculated (see electronic supplementary material, Section S1) as in equation (3.17) with $\mathbf{G}(f_M, t_N)$ replaced with $\delta\mathbf{G}(f_M, t_N)$:

$$\mathcal{C}_{\text{grad}} = \sum_{M=1}^{M_{\max}} \sum_{N=1}^{N_{\max}} \delta t_N\, \delta\mathcal{A}_M \left(\left(A_2 - \frac{B_2}{2} \right) \text{Tr}[\delta\mathbf{G}(f_M, t_N)]^2 + B_2 \text{Tr}[\delta\mathbf{G}(f_M, t_N)^2] \right). \tag{3.18}$$

The soft constraints forcing the normal velocities, defined in equations (2.22) and (2.23), can be calculated as

$$\mathcal{C}_{\text{normal}} = C_n \sum_{M=1}^{M_{\max}} \sum_{N=1}^{N_{\max}} \delta t_N\, \delta\mathcal{A}_M [\hat{\mathbf{n}}(f_M, t_N) \cdot \mathbf{V}(f_M, t_N) - \bar{V}_\perp(f_M, t_N)]^2 \tag{3.19}$$

and

$$\mathcal{C}_{\text{boundary}} = C_b \sum_{L=1}^{L_{\max}} \sum_{N=1}^{N_{\max}} \delta t_N\, \delta\mathcal{L}_L [\hat{\mathbf{n}}_\partial(e_L, t_N) \cdot \mathbf{V}(e_L, t_N) - \bar{V}_\partial(e_L, t_N)]^2, \tag{3.20}$$

where equation (3.20) is summed over the boundary edges $e \in \mathcal{E}_\partial$ with lengths given by $\mathcal{L}_L$.

The last contribution we must add is the landmark constraint given in equation (2.24)

$$\mathcal{C}_{\text{landmark}} = C_L \sum_{\star} \sum_{N=1}^{N_{\max}} \delta t_N [\mathbf{V}_*(t_N) - \bar{\mathbf{V}}_\star(t_N)]^2, \tag{3.21}$$

where, as before, $\star$ indexes the landmark vertices, $\mathbf{V}_*(t_N)$ is the velocity of the landmark vertex, while $\bar{\mathbf{V}}_*(t_N)$ is the prescribed landmark velocity.

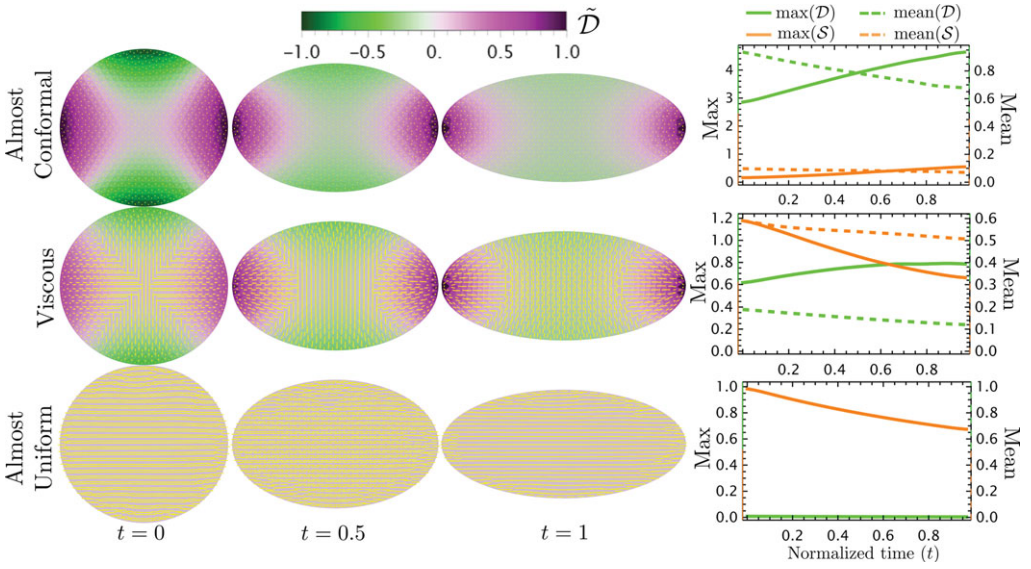

Figure 4. Optimal two-dimensional quasi-conformal flows for a disc undergoing simple shear. The starting point is a disc undergoing expansion by a factor of 3/2 in the horizontal direction, while shrinking by the same factor in the vertical, with area conserved at all times. To infer the possible growth patterns, we run the optimization with three different parameter choices given in table 2: almost-conformal ($A_1 = 1$, all other parameters zero), viscous ($A_1 = B_1 = A_3 = 1$, and all other parameters zero) and almost-uniform ($A_2 = B_2 = 1$, and all other parameters zero). Here $\mathcal{D}$ is the dilation rate given in equation (2.13), and $\mathcal{S}$ is the shear rate given in equation (2.14). The triangles are colour-coded according to the normalized dilation rate, $\tilde{\mathcal{D}} \equiv \mathcal{D}/\max(\mathcal{D}, \mathcal{S})$, while the length of the yellow line segments represents the magnitude of normalized shear rate, $\tilde{\mathcal{S}} \equiv \mathcal{S}/\max(\mathcal{D}, \mathcal{S})$, and their direction is along the largest eigen direction of the strain rate tensor. The last column shows the maximum and mean dilation and shear rates plotted against normalized time for the different registrations in each row.

Finally, we can sum up the contributions to get the total cost as in equation (2.25),

$$\mathcal{C}_{\text{total}}\left[\mathcal{T}(t_N)\right] = \mathcal{C}_{\text{viscous}} + \mathcal{C}_{\text{grad}} + \mathcal{C}_{\text{bend}} + \mathcal{C}_{\text{normal}} + \mathcal{C}_{\text{boundary}} + \mathcal{C}_{\text{landmark}}. \tag{3.22}$$

(f) Minimization and post-processing

To speed up the computation and minimization of equation (3.22), we use the package JAX for just-in-time (JIT) compilation and automatic differentiation [50]. We use the limited memory BFGS [51] to find the optimal velocity field for each vertex at each time step T_N. Once an optimal velocity field is obtained, it can be used to analyse the resulting deformation fields: For each triangle f_M and time t_N, we calculate the growth strain tensor using the velocity field as explained in the previous sections. To visualize the growth strain tensor $\mathbf{G}(f_M, t_N)$—that resulted from the minimization procedure—we calculate its eigenvalues and eigenvectors. One eigenvalue will be zero, with corresponding eigenvector normal to the triangle, while the other two $(\mathcal{G}_1(f_M, t_N), \mathcal{G}_2(f_M, t_N))$ will be in the plane of the triangle. The sum of the eigenvalues will be the dilation rate, $\mathcal{D}(f_M, t_N) \equiv \mathcal{G}_1(f_M, t_N) + \mathcal{G}_2(f_M, t_N)$, while the difference gives the shear rate, $\mathcal{S}(f_M, t_N) \equiv |\mathcal{G}_1(f_M, t_N) - \mathcal{G}_2(f_M, t_N)|$. The dilation rate $\mathcal{D}(f_M, t_N)$ is represented using a colour code (figure 4), while the shear rate $\mathcal{S}(f_M, t_N)$ is illustrated using the length of line segments that point along the eigenvector corresponding to the largest eigenvalue of $\mathbf{G}(f_M, t_N)$.

When post-processing our results, we perform a smoothing step by applying a Gaussian filter to the dilation and shear rates. This is done in Mathematica [52] by using the function GraphDistanceMatrix that calculates the geodesic distances $d_{M_1,M_2}(t_N)$ between the centroids of triangles f_{M_1} and f_{M_2} in the mesh $\mathcal{T}(t_N)$ (represented as a weighted graph with edge weights given

by their lengths). Smoothing of a quantity $\mathcal{Q}(f_M, t_N)$ is then given by

$$\mathcal{Q}_{\text{smooth}}\,(f_{M_1}, t_N) = \frac{1}{\mathcal{N}(M_1)} \sum_{M_2=1}^{M_{\max}} \exp\left(-\frac{d^2_{M_1,M_2}(t_N)}{2\bar{d}^2}\right) \mathcal{Q}(f_{M_2}, t_N), \tag{3.23}$$

where $\bar{d}$ is the smoothing length scale of the filter, which we take as 10% the maximum distance between any two triangles and $\mathcal{N}(M_1) \equiv \sum_{M_2=1}^{M_{\max}} \exp\left(-d^2_{M_1,M_2}(t_N)/2\bar{d}^2\right)$ is the filter normalization factor.

4. Experimental results

To demonstrate the effectiveness of our proposed framework, we consider different classes of surfaces starting from simple two-dimensional shapes with and without landmarks to more complex closed or open surfaces in three dimensions.

(a) Two-dimensional shapes without landmarks

Planar shapes are a special example of surfaces with boundary that are confined to the plane. In this case, all normal displacements vanish, $\bar{V}_\perp(\mathbf{z}, t) = 0$, and the first term of equation (2.22) is minimized by setting $V_\perp(\mathbf{z}, t) = 0$ everywhere. We start by studying the deformation of a disc undergoing simple shear, shrinking by a factor of 1.5 in the vertical direction and expanding by the same factor in the horizontal direction over the course of 30 discrete time steps (figure 4).

We perform three different minimizations, corresponding to the almost-conformal, viscous, and almost-uniform models described in table 2. The almost-conformal model minimizes the shear rate (absolute difference in eigenvalues of the strain rate tensor) and will therefore find a nearly conformal fit to the flow (figure 4). The viscous model minimizes a combination of shear and dilation rates (sum of eigenvalues of the strain rate tensor), while the almost-uniform model favours a strain rate tensor that is uniform in space. Line segments in the third row of figure 4 point in the same direction and have the same length, indicating a constant shear rate, while the dilation rate is zero everywhere. Note that the almost-uniform fit to the flow recovers the constant simple shear transformation that was used to generate this test example, while the almost-conformal and viscous fits predict a completely different growth pattern. Interestingly, as shown in the second row of figure 4, the viscous fits produce domains with distinct orientations and where the dilation rate changes from position (expansion) to negative (contraction).

(b) Two-dimensional shapes with landmarks

In this section, we assume that a subset of the vertices of the mesh are given as landmarks, points where a trajectory $\mathbf{x_p}(t)$ is given *a priori* (e.g. by florescent labeling). These trajectories are enforced in the minimization procedure by adding the corresponding cost as described in equation (2.24) and implemented as described in equation (3.21). We perform the same three minimizations as in the previous section (table 2).

Figure 5 shows the results of our analysis performed on *M. sexta* (tobacco hawk moth) data, which were taken from Nijhout *et al*. [37]. For the almost-conformal model, we observe that the more landmarks we have, the less conformal—as determined by higher values of $\mathcal{S}/\mathcal{D}$ shown in figure 5 compared with electronic supplementary material, figure S1. In addition, the inferred growth patterns depend on the assumed model, just as in the disc example (figure 4). For example, in the absence of landmark constraints (electronic supplementary material, figure S1), the almost-conformal model contains area shrinking (green regions) and expansion (purple regions), while the viscous and almost-uniform models contain only area expansion. Interestingly, we observe that the proximal and distal parts of the wing behave differently in all models with a wave of expansion moving from the proximal to the distal direction over time (see electronic supplementary material, Movie S1).

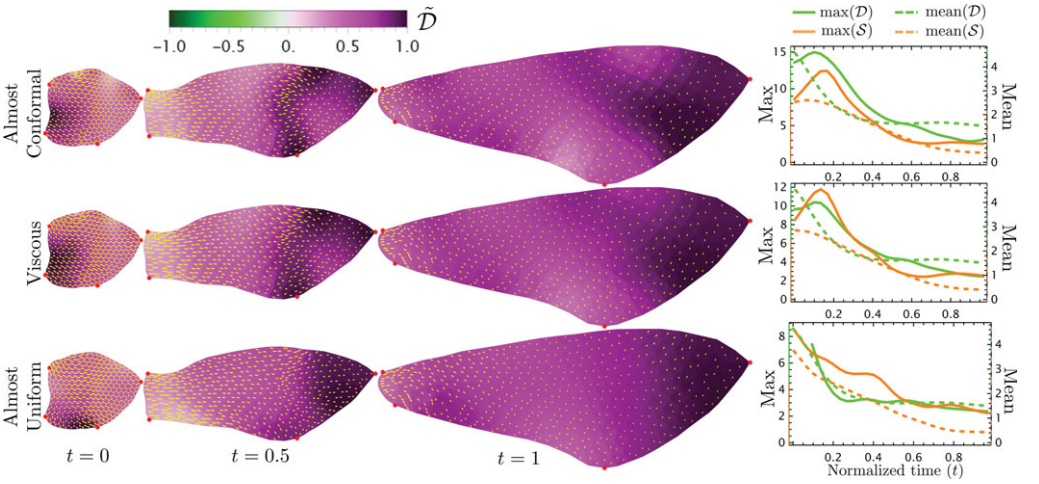

Figure 5. Optimal two-dimensional quasi-conformal flows for *M. sexta* (tobacco hawk moth) wings registration with landmarks. Images of growing *M. sexta* wings are taken from Nijhout *et al*. [37]. To infer the possible growth patterns, we run the optimization with three different parameter choices given in table 2: almost-conformal, viscous and almost-uniform (see the caption of figure 4 for details). See also electronic supplementary material, Movie S1 for the full quasi-conformal flow process.

The corresponding results for *J. coenia* (buckeye butterfly) wing data, also taken from Nijhout *et al*. [37], are shown in figure 6 (see also electronic supplementary material, figure S2 and Movie S2). By comparing the growth patterns obtained with and without landmarks along with the different choice of parameters, we gain insights into the likely growth patterns during insect wing development. For example, for butterfly wings (*J. coenia*), the growth pattern is dominated by isotropic dilation, with $\mathcal{S} < \mathcal{D}$ across all models (see figure 6 and electronic supplementary material, figure S2). Even in the almost-uniform model, where $\mathcal{S}$ is not minimized (only its spatial gradients), the calculated growth patterns exhibit low shear ($\mathcal{S} < \mathcal{D}$), which further decreases when landmarks are included. Furthermore, by noting that the curves for max($\mathcal{D}$) and mean($\mathcal{D}$) are nearly equal, we see that the spatial variation of $\mathcal{D}$ are small. In other words, the butterfly wing grows predominantly but not entirely by uniform scaling (isometric growth). On the other hand, moth wings (*M. sexta*) exhibit much higher shear rates, which increase further when landmarks are included (figure 5 and electronic supplementary material, figure S1). their wings change shape significantly. This can be expected since, as figure 5 shows, the moth wings grow more along the proximal–distal axis compared with the anterior–posterior axis (allometric growth).

(c) Closed surfaces in three dimensions

For surfaces embedded in three dimensions, vertices may move in all three directions, but their movement is constrained by the contributions to the cost function given in equations (2.22) and (2.23), which enforce the displacement of each vertex to be parallel to the surface and the boundary curve. For closed surfaces (our analysis should work independent of the topology of the surface as long as the flow is smooth), we do not need to enforce boundary displacements and we may run the same three registration models as before (almost-conformal, viscous, almost-uniform). The first example (electronic supplementary material, figure S3) is a sphere expanding in one direction by a factor of 1.5 and contracting in the other two to preserve volume in the course of 30 times steps. We see that the conformal model (first row in electronic supplementary material, figure S3) generates nearly conformal flows (small shear rates), where expansion is initially concentrated at the poles of the sphere while contraction dominates in the equator.

The second example we consider (see figure 7 and electronic supplementary material, Movie S3) is a head deforming into a sphere with a linear interpolation in the course of 30 time steps.

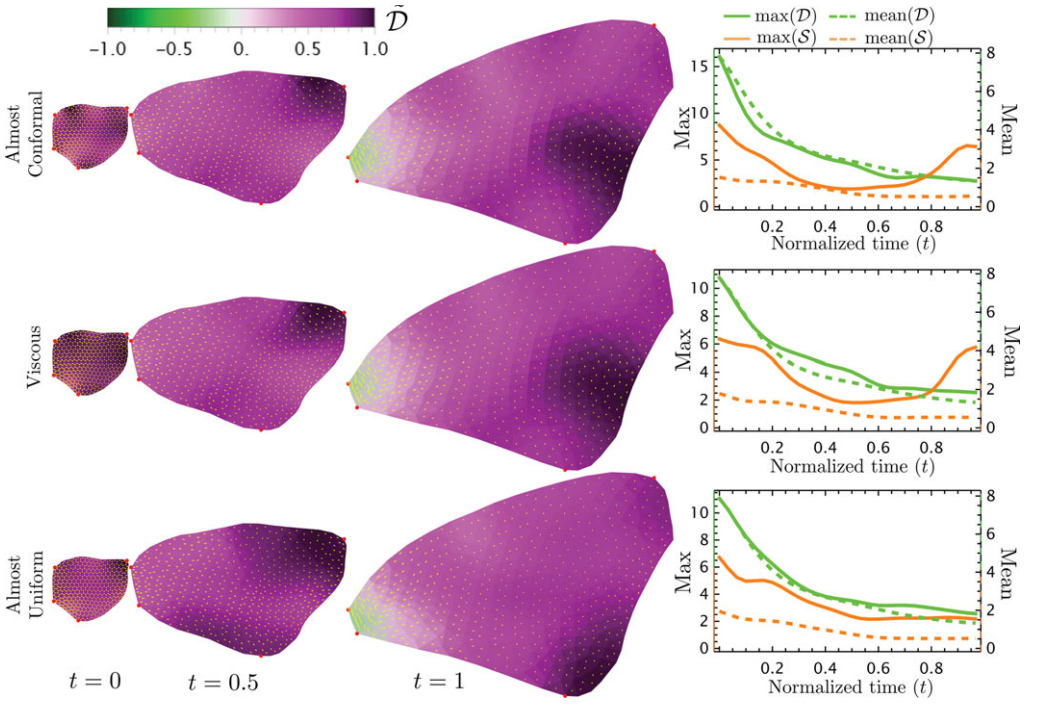

Figure 6. Optimal two-dimensional quasi-conformal flows for *J. coenia* (buckeye butterfly) wing registration with landmarks. Images of growing *J. coenia* wings are taken from Nijhout *et al*. [37]. To infer the possible growth patterns, we run the optimization with three different parameter choices given in table 2: almost-conformal, viscous and almost-uniform (see the caption of figure 4 for details). See also electronic supplementary material, Movie S2 for the full quasi-conformal flow process.

Unlike the first example (electronic supplementary material, figure S3), the almost-conformal model when deforming the head does not reduce the shear rate to zero (orange curve in the first row of figure 7), which happens at sharp features on the mesh, where it becomes harder to estimate the normal vector and tangent plane to the surface. On the other hand, the uniform model leads to a smoother pattern of dilation and shear rates (both in direction and magnitude), including at sharp features.

(d) Surfaces with boundary in three dimensions

We test our algorithm on a *persea americana* (avocado) leaf dataset [39], using almost-conformal, viscous and almost-uniform models (table 2). We see from the results shown in figure 8 (see also electronic supplementary material, Movie S4), that the conformal model leads to low values of the mean shear rate, and therefore a nearly conformal flow. However, it still leads to non-zero shear rate near the tips of the leaf, where the margin deviates from a smooth curve. As before, the inferred growth pattern depends strongly on the model assumed (almost-conformal, viscous or almost-uniform). Therefore, we now explore how modelling the dynamics generating a given flow can help us better recover the growth patterns.

To validate our algorithm, we generate a synthetic dataset involving a cylinder deforming according to the Ricci flow—a solution to $G_{ij} = 0$, where G_{ij} is given in equation (2.17) and $\lambda_1 = \lambda_2 = 0$, and $\lambda_3 = 0.007$—as given by the following equation (illustrated in electronic supplementary material, figure S5):

$$r(z_3) = (r_0 + r_1 \sin(kz_3)e^{\lambda t})\hat{r}(\phi) + \left(z_3 - \frac{r_1}{r_0 k}\cos(kz_3)[e^{\lambda t} - 1]\right) L_0 \hat{z}_3. \tag{4.1}$$

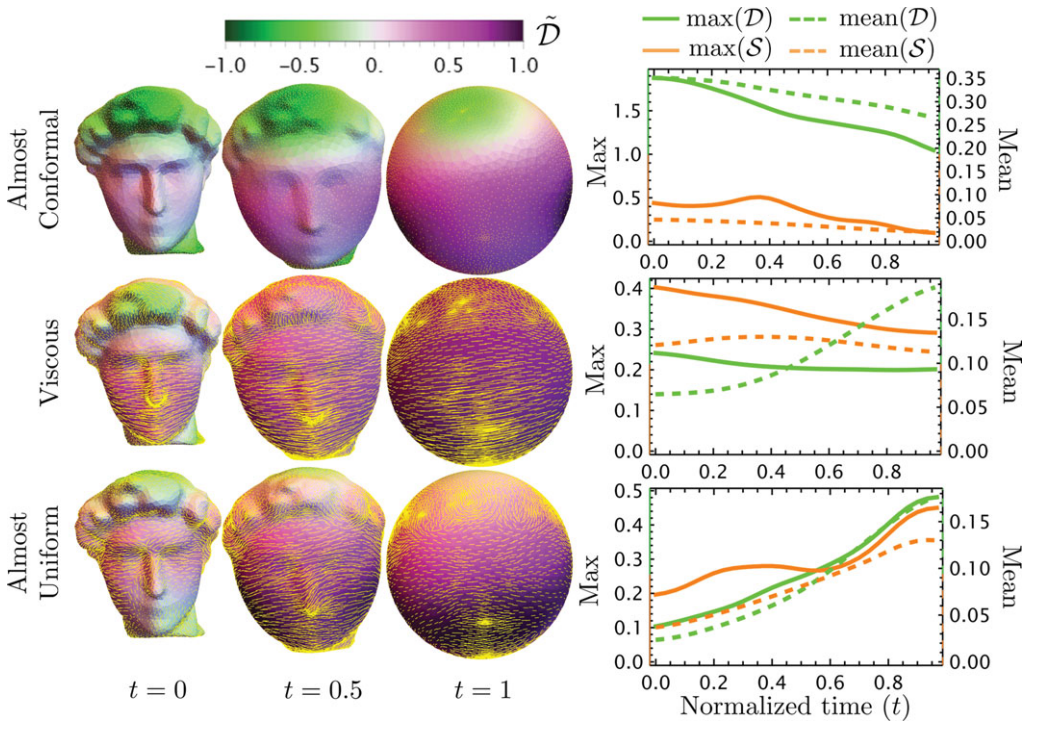

Figure 7. Optimal three-dimensional quasi-conformal flows for closed surfaces. Here we consider a head mesh evolving towards a sphere. We run the optimization with three different parameter choices given in table 2: almost-conformal, viscous and almost-uniform (see the caption of figure 4 for details). See also electronic supplementary material, Movie S3 for the full quasi-conformal flow process.

Here $z_3 \in [0, 1]$ is the vertical coordinate increasing along the length of the cylinder, $\phi \in [0, 2\pi]$ is the azimuthal coordinate, $\hat{r}$ and $\hat{z}_3$ are cylindrical basis vectors, $\lambda = 1$ is a growth rate (multiplied by the duration of the flow T), $r_0 = 0.16$ is the unperturbed cylinder radius, $r_1 = 0.1 r_0$ is the initial amplitude of the deformation, and $L_0 = 1$ is the initial height of the cylinder.

To determine whether the dynamical equation $G_{ij} = 0$ explains the given flow of surfaces $\mathbf{X}(\mathbf{z}, t)$, we minimize the cost function in equation (2.25) using the geometric model parameters given in the last row of table 2, where $\lambda_1, \lambda_2, \lambda_3 \neq 0$, and comparing it to a viscous model (having the same A_1, B_1 but with $\lambda_1 = \lambda_2 = \lambda_3 = 0$). In addition, we perform separate fits for the Ricci ($\lambda_1 = \lambda_2 = 0, \lambda_3 \neq 0$) and Mean ($\lambda_1 = \lambda_3 = 0, \lambda_2 \neq 0$) curvature flows. We find that, as expected, the geometric and Ricci fits, when applied to the growing cylinder dataset, significantly reduce the growth cost (electronic supplementary material, figures S4 and S5), especially for early times, as expected since the dataset was generated using equation (4.1), which is an approximate solution to the Ricci flow, assuming small deviations from a cylinder. Furthermore, the Ricci fit recovers the value $\lambda_3 \approx 0.007$ (with less than 3% error).

The last row of figure 8 shows the results for a growing leaf. While we obtain a correlation between curvature and dilation rate, the minimum growth cost $\mathcal{C}_{\text{viscous}}$ is not significantly reduced in the geometric model. This implies that while growth rate may be correlated with curvature in leaves, the equation $G_{ij} = 0$ is not sufficient to explain the entire growth pattern, some of which may be prescribed in-plane growth that is not curvature-dependent or is triggered by movement of the mid-vain as described in [39]. Furthermore, when we compared the flow inferred by each model to known landmark velocities on the leaf, we found that the Ricci or mean curvature flows are closest to the actual landmark flow compared with other models, followed by the nearly conformal model (see electronic supplementary material, figure S6 for more details).

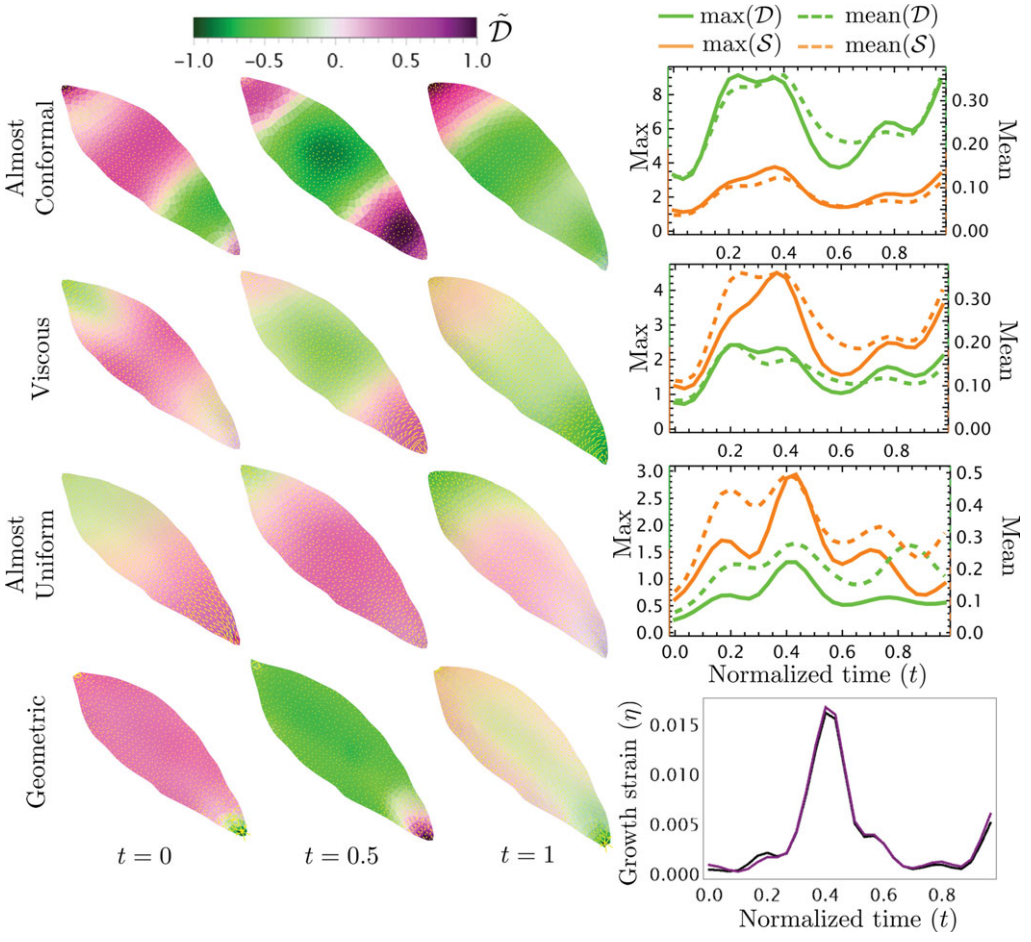

Figure 8. Optimal three-dimensional quasi-conformal flows using geometric flows for growing leaves. Leaf data were taken from Derr & Bastien [39] and processed as described in the text and figure 2. We run the optimization with four different parameter choices given in table 2: almost-conformal, viscous, almost-uniform and geometric (see the caption of figure 4 for details). See also electronic supplementary material, Movie S4 for the full quasi-conformal flow process. The plot on the lower right gives the magnitude of the growth cost, which is defined as $\eta(t) \equiv \langle \mathcal{D}(\mathbf{z}, t) + \mathcal{S}(\mathbf{z}, t) \rangle / 2$, where the angle brackets represent averaging over the area of the surface. The purple curve represents the viscous model, while black represents the geometric model.

(e) Computational cost

As we described above, in our proposed framework we can consider different models, including the almost-conformal, viscous and almost-uniform models. It is natural to ask how the computational cost of the algorithm scales with the input mesh size for each of these models. In electronic supplementary material, figure S7, we consider running the models with various meshes with different number of vertices and recording the run time per simulation step. We see that the almost-conformal and viscous models exhibit comparable computational cost, while the almost-uniform model is more computationally expensive due to the gradient cost computations.

(f) Comparison with other methods

In electronic supplementary material, figure S8, we compare our quasi-conformal flow model with the usual quasi-conformal mapping method [14] and the thin-plate splines mapping method

[53]. More specifically, for both methods, we only use the initial and final time points of the dataset and compute a landmark-based mapping between them. We can then use the mapping result to infer the intermediate state of the growth process. It can be observed that both the landmark-based quasi-conformal mapping method and the thin-plate splines method are unable to capture the intermediate growth patterns of the insect wing shape, and thus unable to interpolate between the initial and final shapes in a biologically consistent manner. This shows the importance of using the proposed quasi-conformal flow method in the study of the temporal developmental processes underlying morphogenesis.

5. Conclusion

In this paper, we have extended the notion of K-quasi-conformal maps to k-quasi-conformal flows for modelling the continuous growth process of biological shapes.

From a biological perspective, understanding the growth and form of shapes is a central problem in biology. The flexibility of our proposed framework allows us to infer growth patterns using a wide range of models with different properties, thereby paving a new way for unveiling the growth dynamics of different biological structures. In addition, besides the almost-conformal, viscous, almost-uniform, and geometric models considered in our current framework, we plan to apply our method to larger datasets to study the dynamics underlying growth processes associated with different biological structures, for example, those that involve nonlocal feedback between mechanical stress and growth [30,54].

From a mathematical perspective, it is worth noting that an optimal quasi-conformal flow[2] between two shapes generally differs from a linear temporal interpolation of an optimal quasi-conformal map between two shapes. Therefore, the use of quasi-conformal flows is important to capture the spatio-temporal variation of the shear and dilation fields during the growth process and opens important mathematical questions with applications to morphodynamics. Considering k-quasi-conformal flows leads to interesting mathematical problems, for example, a Teichmüller map (a K-quasi-conformal map with minimal K) between two surfaces may be distinct from the map generated by a k-quasi-conformal flow—$\mathcal{S}(\mathbf{z}, t) \leq k = K/T$ with minimal k—connecting them. Our framework, especially equation (2.14), can help elucidate such questions by connecting the quantities relevant to quasi-conformal maps, such as $K(\mathbf{z}, t)$, to those related to flows, such as $\mathcal{S}(\mathbf{z}, t)$ as given in equation (2.14).

From an algorithmic perspective, we have proposed a unifying framework for inferring growth patterns using a diverse set of criteria, including local geometric distortions, spatial variations, and fits to dynamical equations, such as geometric flows. Specifically, we can effectively fit different models to a sequence of observed data by considering different combinations of the cost functions described in our work. As demonstrated by our experimental results on different two- and three-dimensional shapes, including insect wings, plant leaves and other open or closed surfaces, our framework is useful whenever a growing structure—in nature or industry—is observed but the identities of individual points are not tracked over time (except possibly for a small set of landmarks).

Our approach depends on minimal finite difference approximations—for example, see equations (3.5)–(3.8)—valid up to first order in δt_N, and is therefore best suited to situations with dense sampling of the growth process. The cost function equation (2.25) can be thought of as a Riemannian metric [17] that gives lengths of paths (or flows) in the space of parametrized surfaces. Therefore, extending our methods beyond the first-order approximations in time amounts to higher-order interpolations of these geodesics. While such geodesics flows have been calculated in the past for special cost functions [56], the advantage of the current approach is greater flexibility in choosing the cost function.

[2]Since our flows minimize integral measures, given in equation (2.25), generalizations of quasi-conformal flows to flows of finite distortion [55]—for example, by requiring that $\mathcal{S}(\mathbf{z}, t)$ is square integrable, but not uniformly bounded by k—maybe more suitable to describe our flows.

Analogously, in the spatial domain, our computations rely on decomposing the displacements into components that are tangent and normal to the surface and its boundary. Such decomposition may not be well behaved for irregular meshes with sharp features. These potential limitations may be overcome by following the philosophy of discrete differential geometry, aiming to develop methods robust for discrete meshes [57,58]. Therefore, it will be fruitful to define discrete quasi-conformal flows in future iterations.

Another direction for future work is the comparison of hard and soft constraints to enforce landmark and normal displacements that are taken from the input data. The advantages of soft constraints are ease of implementation and tunability of the relative importance of input data compared with model prediction. This is analogous to optimal filters used in control theory to tune the relative importance of measurement data versus model predictions based on the strength of the noise from each source (measurement uncertainty and model error), or equivalently, the role of priors and data in the context of inference problems. More generally, there is much to be gained by combining the power of geometrical approaches that harness the structured information in morphodynamics with data-driven computational models that are efficient and fast, and this paper takes a step in this direction.

Data accessibility. The codes and data are available on the Dryad Digital Repository [59].

The data are provided in the electronic supplementary material [60].

Declaration of AI use. We have not used AI-assisted technologies in creating this article.

Authors' contributions. S.M.: conceptualization, data curation, formal analysis, investigation, methodology, software, validation, visualization, writing—original draft, writing—review and editing; G.P.T.C.: conceptualization, data curation, formal analysis, investigation, methodology, software, validation, visualization, writing—original draft, writing—review and editing; L.M.: conceptualization, formal analysis, project administration, resources, supervision, validation, writing—original draft, writing—review and editing.

All authors gave final approval for publication and agreed to be held accountable for the work performed therein.

Conflict of interest declaration. We declare we have no competing interests.

Funding. This work was supported in part by the CUHK Faculty of Science Direct Grant for Research no. 4053650 (to G.P.T.C.), the NSF-Simons Center for Mathematical and Statistical Analysis of Biology at Harvard NSF 1764269 (to L.M.), the Simons Foundation (L.M.) and the Henri Seydoux Fund (L.M.), and the Defense Advanced Research Project Agency (DARPA) under contract no. HR00112220032 (S.M. and L.M.).

References

1. Thompson DW. 1917 *On growth and form*, pp. 1–15. Cambridge, UK: Cambridge University Press.
2. Ahlfors LV. 1953 *Complex analysis: an introduction to the theory of analytic functions of one complex variable*, vol. 2, pp. 68–75. New York, NY: McGraw-Hill.
3. Sharon E, Mumford D. 2006 2D-shape analysis using conformal mapping. *Int. J. Comput. Vis.* **70**, 55–75. (doi:10.1007/s11263-006-6121-z)
4. Lévy B, Petitjean S, Ray N, Maillot J. 2002 Least squares conformal maps for automatic texture atlas generation. *ACM Trans. Graph.* **21**, 362–371. (doi:10.1145/566654.566590)
5. Desbrun M, Meyer M, Alliez P. 2002 Intrinsic parameterizations of surface meshes. *Comput. Graph. Forum* **21**, 209–218. (doi:10.1111/1467-8659.00580)
6. Mitchison G. 2016 Conformal growth of Arabidopsis leaves. *J. Theor. Biol.* **408**, 155–166. (doi:10.1016/j.jtbi.2016.08.023)
7. Alba V, Carthew JE, Carthew RW, Mani M. 2021 Global constraints within the developmental program of the Drosophila wing. *Elife* **10**, e66750. (doi:10.7554/eLife.66750)
8. Gu X, Wang Y, Chan TF, Thompson PM, Yau ST. 2004 Genus zero surface conformal mapping and its application to brain surface mapping. *IEEE Trans. Med. Imaging* **23**, 949–958. (doi:10.1109/TMI.2004.831226)
9. Choi PT, Lam KC, Lui LM. 2015 FLASH: fast landmark aligned spherical harmonic parameterization for genus-0 closed brain surfaces. *SIAM J. Imaging Sci.* **8**, 67–94. (doi:10.1137/130950008)

10. Kierzkowski D *et al.* 2019 A growth-based framework for leaf shape development and diversity. *Cell* **177**, 1405–1418. (doi:10.1016/j.cell.2019.05.011)
11. Zhao F *et al.* 2020 Microtubule-mediated wall anisotropy contributes to leaf blade flattening. *Curr. Biol.* **30**, 3972–3985. (doi:10.1016/j.cub.2020.07.076)
12. Ahlfors LV. 1966 *Lectures on quasiconformal mappings*, vol. 1, pp. 1–17. Princeton, NJ: D. Van Nostrand Company, Inc.
13. Jones GW, Mahadevan L. 2013 Planar morphometry, shear and optimal quasi-conformal mappings. *Proc. R. Soc. A* **469**, 20120653. (doi:10.1098/rspa.2012.0653)
14. Choi GPT, Mahadevan L. 2018 Planar morphometrics using Teichmüller maps. *Proc. R. Soc. A* **474**, 20170905. (doi:10.1098/rspa.2017.0905)
15. Choi GPT, Chan HL, Yong R, Ranjitkar S, Brook A, Townsend G, Chen K, Lui LM. 2020 Tooth morphometry using quasi-conformal theory. *Pattern Recognit.* **99**, 107064. (doi:10.1016/j.patcog.2019.107064)
16. Choi GPT, Qiu D, Lui LM. 2020 Shape analysis via inconsistent surface registration. *Proc. R. Soc. A* **476**, 20200147. (doi:10.1098/rspa.2020.0147)
17. Klassen E, Srivastava A, Mio M, Joshi SH. 2004 Analysis of planar shapes using geodesic paths on shape spaces. *IEEE Trans. Pattern Anal. Mach. Intell.* **26**, 372–383. (doi:10.1109/TPAMI.2004.1262333)
18. Beg MF, Miller MI, Trouvé A, Younes L. 2005 Computing large deformation metric mappings via geodesic flows of diffeomorphisms. *Int. J. Comput. Vis.* **61**, 139–157. (doi:10.1023/B:VISI.0000043755.93987.aa)
19. Reimann HM. 1976 Ordinary differential equations and quasiconformal mappings. *Invent. Math.* **33**, 247–270. (doi:10.1007/BF01404205)
20. Bonk M, Heinonen J, Saksman E. 2008 Logarithmic potentials, quasiconformal flows, and Q-curvature. *Duke Math. J.* **141**, 197–239. (doi:10.1215/00127094-2008-005)
21. Austin AD. 2020 Logarithmic potentials and quasiconformal flows on the Heisenberg group. *Adv. Math.* **364**, 107013. (doi:10.1016/j.aim.2020.107013)
22. Chang SYA, Prywes E, Yang P. 2022 Quasiconformal flows on non-conformally flat spheres. *Adv. Math.* **404**, 108373. (doi:10.1016/j.aim.2022.108373)
23. Srivastava A, Klassen EP. 2016 *Functional and shape data analysis*, vol. 1. New York, NY: Springer.
24. Tumpach AB, Drira H, Daoudi M, Srivastava A. 2015 Gauge invariant framework for shape analysis of surfaces. *IEEE Trans. Pattern Anal. Mach. Intell.* **38**, 46–59. (doi:10.1109/TPAMI.2015.2430319)
25. Jermyn IH, Kurtek S, Klassen E, Srivastava A. 2012 Elastic shape matching of parameterized surfaces using square root normal fields. In *Computer Vision–ECCV 2012: 12th European Conf. on Computer Vision*, Florence, Italy, 7–13 October 2012, pp. 804–817, Proceedings, Part V 12. Berlin, Germany: Springer.
26. Hamilton RS. 1988 The Ricci flow on surfaces, Mathematics and general relativity. *Contemp. Math.* **71**, 237–261. (doi:10.1090/conm/071/954419)
27. Mullins WW. 1956 Two-dimensional motion of idealized grain boundaries. *J. Appl. Phys.* **27**, 900–904. (doi:10.1063/1.1722511)
28. Chang F, Huang KC. 2014 How and why cells grow as rods. *BMC Biol.* **12**, 54. (doi:10.1186/s12915-014-0054-8)
29. Al-Mosleh S, Gopinathan A, Santangelo CD, Huang KC, Rojas ER. 2022 Feedback linking cell envelope stiffness, curvature, and synthesis enables robust rod-shaped bacterial growth. *Proc. Natl Acad. Sci. USA* **119**, e2200728119. (doi:10.1073/pnas.2200728119)
30. al Mosleh S, Mahadevan L. 2023 How to grow a flat leaf. *Phys. Rev. Lett.* **131**, 098401. (doi:10.1103/PhysRevLett.131.098401)
31. Al-Mosleh S, Choi GPT, Abzhanov A, Mahadevan L. 2021 Geometry and dynamics link form, function, and evolution of finch beaks. *Proc. Natl Acad. Sci. USA* **118**, e2105957118. (doi:10.1073/pnas.2105957118)
32. Al Mosleh S, Gopinathan A, Santangelo C. 2018 Growth of form in thin elastic structures. *Soft Matter* **14**, 8361–8371. (doi:10.1039/C8SM01136B)
33. Landau LD, Pitaevskii L, Kosevich AM, Lifshitz EM. 2012 *Theory of elasticity*, vol. 7. Amsterdam, The Netherland: Elsevier. pp. 1–37.
34. Stein O, Wardetzky M, Jacobson A, Grinspun E. 2020 A simple discretization of the vector Dirichlet energy. *Comput. Graph. Forum* **39**, 81–92. (doi:10.1111/cgf.14070)

35. Do Carmo MP. 2016 *Differential geometry of curves and surfaces: revised and updated*, 2nd edn. Mineola, NY: Courier Dover Publications. pp. 217–314.
36. Webster M, Sheets HD. 2010 A practical introduction to landmark-based geometric morphometrics. *Paleontol. Soc. Pap.* **16**, 163–188. (doi:10.1017/S1089332600001868)
37. Nijhout HF, Cinderella M, Grunert LW. 2014 The development of wing shape in Lepidoptera: mitotic density, not orientation, is the primary determinant of shape. *Evol. Dev.* **16**, 68–77. (doi:10.1111/ede.12065)
38. Persson PO, Strang G. 2004 A simple mesh generator in MATLAB. *SIAM Rev.* **46**, 329–345. (doi:10.1137/S0036144503429121)
39. Derr J, Bastien R. 2018 Fluttering of growing leaves as a way to reach flatness: experimental evidence on Persea Americana. *J. R. Soc. Interface* **15**, 20170595. (doi:10.1098/rsif.2017.0595)
40. Taubin G. 1995 Curve and surface smoothing without shrinkage. In *Proc. of IEEE Int. Conf. on Computer Vision*, Cambridge, MA, 20–23 June 1995. New York, NY: IEEE, pp. 852–857.
41. Loop C. 1987 Smooth subdivision surfaces based on triangles. Master's thesis, University of Utah, Department of Mathematics.
42. Kroon DJ. 2011 2D Line curvature and normals. See https://www.mathworks.com/matlabcentral/fileexchange/32696-2d-line-curvature-and-normals.
43. Kroon DJ. 2018 Curvature estimation on triangle mesh. See https://www.mathworks.com/matlabcentral/fileexchange/32573-patch-curvature.
44. Friedrich B. 2014 frenet_robust. See https://www.mathworks.com/matlabcentral/fileexchange/47885-frenet_robust-zip.
45. Swarztrauber PN, Williamson DL, Drake JB. 1998 The Cartesian method for solving partial differential equations in spherical geometry. *Dyn. Atmos. Oceans* **27**, 679–706. (doi:10.1016/S0377-0265(97)00038-9)
46. Ruuth SJ, Merriman B. 2008 A simple embedding method for solving partial differential equations on surfaces. *J. Comput. Phys.* **227**, 1943–1961. (doi:10.1016/j.jcp.2007.10.009)
47. Azencot O, Ovsjanikov M, Chazal F, Ben-Chen M. 2015 Discrete derivatives of vector fields on surfaces–an operator approach. *ACM Trans. Graph.* **34**, 1–13. (doi:10.1145/2723158)
48. Plangger R, Krall A. 2016 Vectorization in PyPy's tracing just-in-time compiler. In *Proc. of the 19th Int. Workshop on Software and Compilers for Embedded Systems*, Sankt Goar, Germany, 23–25 May 2016, pp. 67–76. New York, NY: ACM.
49. Harris CR *et al.* 2020 Array programming with NumPy. *Nature* **585**, 357–362. (doi:10.1038/s41586-020-2649-2)
50. Bradbury J *et al.* 2018 JAX: composable transformations of Python+NumPy programs.
51. Liu DC, Nocedal J. 1989 On the limited memory BFGS method for large scale optimization. *Math. Program.* **45**, 503–528. (doi:10.1007/BF01589116)
52. Wolfram Research, Inc. Mathematica, Version 14.0. 2024. www.wolfram.com/mathematica. Champaign, IL.
53. Bookstein FL. 1989 Principal warps: thin-plate splines and the decomposition of deformations. *IEEE Trans. Pattern Anal. Mach. Intell.* **11**, 567–585. (doi:10.1109/34.24792)
54. Harline K *et al.* 2022 Dynamic growth re-orientation orchestrates flatness in the Arabidopsis leaf. *BioRxiv* pp. 2022–11. (doi:10.1101/2022.11.01.514736)
55. Hencl S, Koskela P. 2014 *Lectures on mappings of finite distortion*, pp. 1–3. Berlin, Germany: Springer.
56. Kurtek S, Klassen E, Gore JC, Ding Z, Srivastava A. 2011 Elastic geodesic paths in shape space of parameterized surfaces. *IEEE Trans. Pattern Anal. Mach. Intell.* **34**, 1717–1730. (doi:10.1109/TPAMI.2011.233)
57. Grinspun E, Desbrun M, Polthier K, Schröder P, Stern A. 2006 Discrete differential geometry: an applied introduction. *ACM Siggraph Course* **7**, 1–83.
58. Crane K, Wardetzky M. 2017 A glimpse into discrete differential geometry. *Notices Am. Math. Soc.* **64**, 1153–1159. (doi:10.1090/noti1578)
59. Mosleh S, Choi GPT, Mahadevan L. 2025 Data-driven quasiconformal morphodynamic flows. Dryad Digital Repository. (doi:10.5061/dryad.v41ns1s6t).
60. Mosleh S, Choi GPT, Mahadevan L. 2025 Data-driven quasi-conformal morphodynamic flows. Figshare. (doi:10.6084/m9.figshare.c.7741393)