Quad mesh mechanisms

CAIGUI JIANG, Xi'an Jiaotong University, China
DMITRY LYAKHOV, KAUST, Saudi Arabia
FLORIAN RIST, TU Wien and KAUST, Austria / Saudi Arabia
HELMUT POTTMANN, TU Wien and KAUST, Austria / Saudia Arabia
JOHANNES WALLNER, TU Graz, Austria

This paper provides computational tools for the modeling and design of quad mesh mechanisms, which are meshes allowing continuous flexions under the assumption of rigid faces and hinges in the edges. We combine methods and results from different areas, namely differential geometry of surfaces, rigidity and flexibility of bar and joint frameworks, algebraic geometry, and optimization. The basic idea to achieve a time-continuous flexion is time-discretization justified by an algebraic degree argument. We are able to prove computationally feasible bounds on the number of required time instances we need to incorporate in our optimization. For optimization to succeed, an informed initialization is crucial. We present two computational pipelines to achieve that: one based on remeshing isometric surface pairs, another one based on iterative refinement. A third manner of initialization proved very effective: We interactively design meshes which are close to a narrow known class of flexible meshes, but not contained in it. Having enjoyed sufficiently many degrees of freedom during design, we afterwards optimize towards flexibility.

CCS Concepts: • **Computing methodologies → Shape modeling**; *Optimization algorithms.*

Additional Key Words and Phrases: transformable design, flexible meshes, kinematics, discrete differential geometry, isometry

ACM Reference Format:
Caigui Jiang, Dmitry Lyakhov, Florian Rist, Helmut Pottmann, and Johannes Wallner. 2024. Quad mesh mechanisms. *ACM Trans. Graph.* 43, 6, Article 243 (December 2024), 17 pages. https://doi.org/10.1145/3687939

1 INTRODUCTION

Our work on flexible meshes is motivated by several factors. Firstly, rigidity and flexibility is a highly interesting topic with a long list of beautiful and surprising results. Secondly, the field is known for resisting systematic modeling and computational design. Finally, flexible structures have many applications, from mechanical engineering to origami, and to the recent trend of *transformable designs* in art and architecture. This paper focuses on *meshes*, particularly quad meshes, whose faces move as rigid bodies while being connected with hinges. An example of this is given by Fig. 2, where flexible "Voss type" quad meshes have been realized as an artistic object [Baldwin 2018; Mitchell et al. 2018]. Mechanisms of this kind are not easy to design – in fact, transformable designs often have to employ bendable elements to achieve flexibility. It is precisely this design problem which is addressed in the present paper.

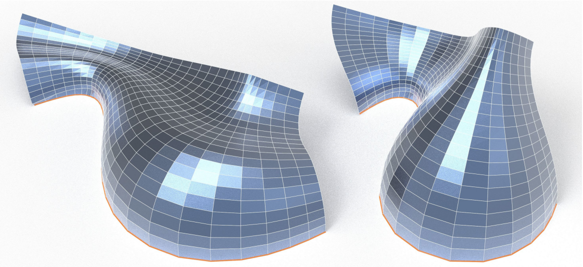


Fig. 1. A quad mesh mechanism as an architectural transformable design. It has been initialized by interactive design, using the 'optimization-driven exploration' method we present below. Besides flexibility, it fulfills other side-conditions important to this particular application: The highlighted horizontal boundary remains in the base plane during deployment. This has been achieved by keeping z coordinates of vertices fixed during optimization.

Specifically, we treat the following old and so far largely unsolved problem: Compute *quad mesh mechanisms*, i.e., quad meshes which are flexible under the assumption that faces are rigid and edges act as hinges between them. Our method uses optimization, an important part of which is the proper initialization. For that, we use results on discrete isometries and on infinitesimal isometries, in particular the work by Sauer [1970]. In this way, we are able to address the problem in greater generality and go beyond prior work which is based on the analysis of special cases.

1.1 Contributions and Overview

After a recap of isometries, flexions and infinitesimal flexions of meshes in § 2 we present algorithms for basic tasks used in the design of flexible meshes. § 3.1 shows how to optimize two meshes to become isometric to each other and in addition to become infinitesimally flexible. § 3.2 discusses quad-meshing a pair of surfaces such that the resulting meshes enjoy planar faces and are isometric. § 3.3 does the same without the planar faces requirement. In § 3.4 we use algebraic considerations to reduce the problem of finding time-continuous flexions to a time-discrete sequence of isometric meshes. § 3.5 gives tools to create a dense sequence of meshes approximating a continuous flexion. Finally, § 4 combines the tasks listed above and presents computational pipelines for the design of quad mesh mechanism: one is based on remeshing, and another one on iterative refinement. A third approach is based on exploring the configuration space, which is discussed by § 4.3: We design meshes which enjoy some, but not all properties of known classes of flexible

Authors' addresses: Caigui Jiang, Xi'an Jiaotong University, China; Dmitry Lyakhov, KAUST, Saudi Arabia; Florian Rist, TU Wien and KAUST, Austria / Saudi Arabia; Helmut Pottmann, TU Wien and KAUST, Austria / Saudia Arabia; Johannes Wallner, TU Graz, Austria.

0730-0301/2024/12-ART243
https://doi.org/10.1145/3687939

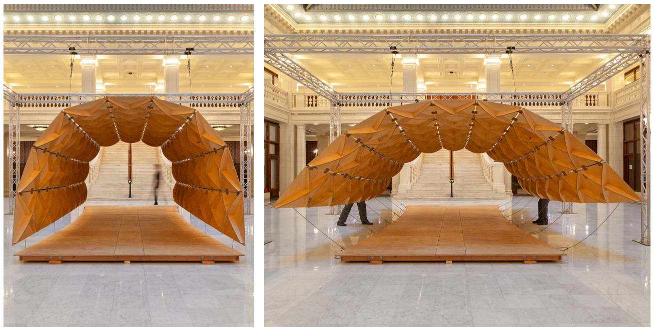


Fig. 2. *Kinematic Sculpture* by Skidmore, Owings & Merrill, an installation for the 2018 Chicago Design week (© SOM).

meshes, such that there is still enough design freedom. Optimization towards flexibility of such results is surprisingly effective. § 5 concludes the paper with a discussion of results, implementation details, physical models, limitations and pointers to future work.

The main result of the present paper is the interplay between algebraic and numerical results: A degree bound allows us to convert the search for a continuous flexion to the computation of a *finite* number of isometric meshes.

1.2 Previous Work

Here we work with mechanisms, with the rigidity and flexibility of bar and joint frameworks, and also the isometric bending of surfaces. There is extensive prior work on each of these subjects.

Flexions of Quad Meshes with Planar Faces. Flexible quad meshes with planar rigid faces have been studied by [Kokotsakis 1933] and follow-up work, e.g. [Stachel 2010, 2011], in particular with a focus on origami [Tachi 2009, 2013; Tachi and Epps 2011]. For regular quad meshes with planar faces, Schief et al. [2008] pointed out that it is essentially sufficient to consider meshes of size 3 × 3. The full classification of all flexible types was obtained by Izmestiev [2017]. The *design* of flexible quad meshes has been treated in special cases only: He and Guest [2020] discuss several elementary operations on meshes which preserve flexibility, including the stitching together of smaller meshes.

Flexions of Quad Meshes with Non-Planar Faces. We are aware of only two contributions, studying “small” examples with non-planar quads, namely by Nawratil [2023] and by Aikyn et al. [2024]. The latter contains a generalization of the so-called Bricard equations to the case of nonplanar faces. This result is directly used in the present paper.

Flexions of Meshes Analogous to Bending of Surfaces. Isometric bending of surfaces has been extensively surveyed, see e.g. [Sabitov 1992]. It is important to appreciate that even if a discrete surface approximates a smooth one, the flexions of the discrete surface do not usually approximate the bendings which the smooth surface is capable of. This happens only in special cases. One example is provided by *Voss surfaces*, defined by the existence of a geodesic conjugate curve network [Voss 1888]. Discrete Voss surfaces are defined via vertex angles and occur as reciprocals of discrete K-surfaces [Sauer 1970; Wunderlich 1951]. They have recently been employed for grid shell applications [Montagne et al. 2020], and they include the well-known unfolding solar panels of Miuro-Ori type as a special case [Miura 1980]. Another class is so-called profile-affine surfaces (which include rotational surfaces). Their discrete analogs are characterized by trapezoidal faces [Sauer 1970]. The design of such T-net mechanisms has been proposed by [Sharifmoghaddam et al. 2021], and their flexions have in more detail been studied by [Izmestiev et al. 2023]. In the special case of surfaces projectively equivalent to profile-affine surfaces, Nawratil [2024] succeeded in classifying all flexible cases.

Flexions Which Include a Flat State. Flexible quad meshes relevant for paper-folding (origami) have a special property, namely existence of an overall flat state. They are well studied, see e.g. [Dudte et al. 2016; Tachi 2009, 2010a,b, 2013; Tachi and Epps 2011]; we also refer to the monograph [Demaine and O’Rourke 2007]. A topic related to both origami and mechanisms is origami with thick panels [Chen et al. 2015]. More recently, there have been contributions to design, based on local analysis [Evans et al. 2015a,b; Feng et al. 2020] or special geometries [Song et al. 2017]. Design of flat-foldable meshes of general shape based on a marching algorithm acting on boundary values has been proposed by [Dang et al. 2022].

Infinitesimal Rigidity and Flexibility. Infinitesimal flexibility of a mechanism has mostly been studied in the general context of bar and joint frameworks; our setting also fits this general description. We refer to [Connelly 1987] for a textbook introduction, and to [Sauer 1970; Schief et al. 2008] for the specific case of quad meshes, which is treated from the viewpoint of discrete differential geometry. The relations between finite flexibility, infinitesimal (1st order) flexibility and higher order flexibility are geometrically interesting, and sometimes not at all obvious [Connelly and Servatius 1994].

Applications and Computing. So far quad mesh mechanisms in our sense have hardly been explored from the viewpoint of geometry processing. The prior work closest to our paper is by Jiang et al. [2021]. They propose a simplified version of the remeshing procedure of § 3.2, in order to find the edges of a quad mesh mechanism which approximates certain isometric mappings between surfaces. On the other hand the isometric mapping of surfaces has been treated successfully, and in different ways, most recently by [Chern et al. 2018; Jiang et al. 2020, 2021]. As to mechanisms, an important application of mechanisms based on meshes is *mechanical metamaterials*. Their deployment typically approximates surface deformations with local scaling factors ≥ 1, not necessarily isometric mappings. We refer to survey articles [Callens and Zadpoor 2018; Zhai et al. 2021]. Mechanical metamaterials might be based on a watertight quad mesh mechanism in our sense, but in this case do not exhibit fairness (see e.g. [Jiang et al. 2019; Miura 1980]). They might be based on meshes where the connectivity is enforced only for part of the edges, or even only in vertices [Dudte et al. 2016; Konaković et al. 2016; Konaković-Luković et al. 2018]. Recently, focus has been on modular and multi-DOF metamaterials [Jamalimehr et al. 2022; Xiao et al. 2022].

One motivation for our work is *transformable designs*, see e.g. [Hoberman 2006]. These are not restricted to rigid parts, and one sometimes finds blueprints for technical solutions in nature [Lienhard et al. 2011; Masselter et al. 2012]. Contributions to computation are diverse: There are shells which emerge by the deployment of elastic beams [Panetta et al. 2019; Pillwein et al. 2020a,b; Soriano et al. 2019], structures which constitute a mixture of rigid and flexible parts [Guseinov et al. 2017; Malomo et al. 2018], and fully flexible structures where rigidity is imposed by folding [Kilian et al. 2017].

2 BASIC CONCEPTS

In § 2 we recall some basic facts on isometric bending of surfaces, as well as on infinitesimal isometries. The latter are going to be used in the initialization of our optimization algorithms. Figure 3 provides an overview of the contents of this section. In our exposition we go into greater detail than strictly necessary, since certain parts of the material are only available in the German text by Sauer [1970]. Other recent work addressing isometric mappings between surfaces and providing a short introduction is [Jiang et al. 2021]. We further discuss the relation between two different isometric positions of surfaces in both the discrete and continuous cases.

2.1 Isometries of smooth surfaces

2.1.1 Bending of a Surface and the Associated Velocity Diagram and Rotation Diagram. We consider a continuous isometric deformation of a smooth surface $x(u,v)$ over time t. “Isometric” means that the inner products of partial derivatives x_u, x_v do not depend on time t. These inner products are stored in the coefficients of the first fundamental form, namely $E = \langle x_u, x_u\rangle$, $F = \langle x_u, x_v\rangle$, $G = \langle x_v, x_v\rangle$. The conditions $\dot{E} = \dot{F} = \dot{G} = 0$ expand to

$$\langle x_u, \dot{x}_u\rangle = 0,\ \langle x_u, \dot{x}_v\rangle + \langle x_v, \dot{x}_u\rangle = 0,\ \langle x_v, \dot{x}_v\rangle = 0. \tag{1}$$

The dot indicates the derivative with respect to time t. Following [Sauer 1970], we consider the *velocity diagram* $\dot{x}(u,v)$ of the surface. It is not difficult to see that Equ. (1) has a geometric implication: A curve $x(u(s), v(s))$ on the surface is orthogonal to its corresponding curve $\dot{x}(u(s), v(s))$ in the velocity diagram. This relation will also be observed in the case of discrete surfaces.

For any fixed choice (u,v), the tangent vectors x_u, x_v attached to $x(u,v)$ move rigidly, thanks to Equ. (1). It is well known that then there is a vector of angular veclocity $c(u,v)$ and a translational veclocity component $\bar{c}(u,v)$ such that the motion of the point $x(u,v)$ and of tangent vectors x_u, x_v is expressed as

$$\dot{x} = \bar{c} + c \times x. \tag{2}$$

$$\dot{x}_u = c \times x_u, \quad \dot{x}_v = c \times x_v. \tag{3}$$

Sauer [1970] calls the surfaces $c(u,v)$ resp $\bar{c}(u,v)$ the *rotation diagram* resp. *translation diagram.*

As to derivatives of the surface $c(u,v)$ observe that $(\dot{x} - \bar{c})_u = (c \times x)_u = c_u \times x + c \times x_u = c_u \times x - \dot{x}_u$, where the last equality used (3). Together with the analogous computation for v this shows that

$$\bar{c}_u = x \times c_u, \quad \bar{c}_v = x \times c_v.$$

This establishes the remarkable symmetry between pairs $x, \dot{x}$ on the one hand and $c, \bar{c}$ on the other hand: Each pair can be seen as rotation diagram plus translation diagram of the other pair.

Differentiating (3) and imposing the condition $\dot{x}_{uv} = \dot{x}_{vu}$ yields

$$c_v \times x_u = c_u \times x_v. \tag{4}$$

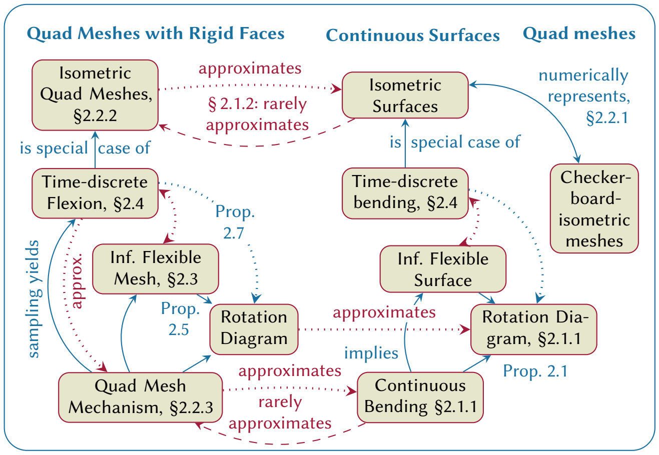


Fig. 3. This diagram lists the notions introduced in § 2 and their relations, with an arrow indicating that one object is determined by another. If all arrows were reversible, the problem of computing quad mesh mechanisms would be easier. Derived concepts like the rotation diagrams are relevant because they can be used to recognize unsolvable cases.

This shows that all four involved vectors c_u, c_v, x_u, x_v are co-planar, and thus corresponding tangent planes of surfaces $x(u,v)$ and $c(u,v)$ are parallel to each other.

2.1.2 Approximating the Bending of Surfaces by Flexion of Meshes. It would be very helpful if continuous bending of surfaces could be approximated by flexions of quad meshes whose edges follow the surface’s parameter lines and whose faces move rigidly. Unfortunately this works only in special cases. The reason for this basically is the fact that isometric bending is concerned with 1st derivatives only, but a face of a quad mesh encodes both 1st and 2nd order discrete derivatives. Consider a quad mesh approximating a smooth parametric surface, and a face that is sampled from the parametrization. It has vertices

$$v_0 = x(u,v),\ v_1 = x(u+h,v),\ v_2 = x(u+h,v+k),\ v_3 = x(u,v+k).$$

Then mixed derivatives and a normal vector can be expressed in terms of these vertices:

$$x_{uv}(u,v) \approx \frac{v_0 - v_1 + v_2 - v_3}{hk}, \qquad n(u,v) \approx \frac{(v_1 - v_3) \times (v_0 - v_2)}{\|(v_1 - v_3) \times (v_0 - v_2)\|}.$$

Rigid motions preserve inner products. Thus, only bendings where

$$M(u,v) := \langle n, x_{uv}\rangle \quad \text{is constant over time} \tag{5}$$

can possibly be approximated by an isometric flexion of the mesh.

The case $M(u,v) = 0$ is well known; these are the conjugate surfaces which correspond to quad meshes with planar faces. In this case there is previous work on flexibility. E.g. a 2nd order infinitesimal isometry of a smooth surface implies existence of a continuous bending of that surface [Schief et al. 2008]. Special cases of flexible meshes such as Voss nets and T-nets have already been discussed by Sauer [1970]. However, flexible quad meshes with non-planar faces are almost unexplored, apart from a result by Sauer [1970] concerning first order flexibility:

Proposition 2.1. [Sauer 1970] *In case $M = const.$, the surface $x(u, v)$ and the rotation diagram $c(u, v)$ are reciprocal-parallel, meaning there are functions $\rho(u, v), \tau(u, v)$ with*

$$c_u = \rho x_v, \; c_v = \tau x_u. \tag{6}$$

Further, the velocity diagram $\dot{x}(u, v)$ is a conjugate surface, i.e.,

$$\det(\dot{x}_u, \dot{x}_v, \dot{x}_{uv}) = 0. \tag{7}$$

Proof. The unit normal vector field is expressed as $n = (x_u \times x_v)/\lambda$ with $\lambda = \|x_u \times x_v\|$. Since $\lambda = \sqrt{EG - F^2}$ and (1) implies $\dot{E} = \dot{F} = \dot{G} = 0$, we also have $\dot{\lambda} = 0$. Thus $\dot{M} = \frac{d}{dt}\langle \frac{1}{\lambda}(x_u \times x_v), x_{uv}\rangle = 0$ is equivalent to $\frac{d}{dt}\langle x_u \times x_v, x_{uv}\rangle = 0$. Expanding this condition yields $\det(\dot{x}_u, x_v, x_{uv}) + \det(x_u, \dot{x}_v, x_{uv}) + \det(x_u, x_v, \dot{x}_{uv}) = 0$. Some manipulations using (3) and (4) yield $\det(c_v \times x_u, x_u, x_v) = \det(c_u \times x_v, x_u, x_v) = 0$. This is possible only if (6) holds. Equ. (7) follows by some more elementary manipulations. □

Later we are going to approximate the bending of a surface by the flexion of a quad mesh. We will make use of the conjugacy expressed by Prop. 2.1 – it is known that conjugate surfaces are discretized by quad meshes with planar faces [Sauer 1970].

2.2 Isometries of discrete surfaces

In our pursuit of quad mesh mechanisms we need to model isometries between surfaces, where those surfaces are represented by quad meshes. For that we employ the "checkerboard" approach proposed by Jiang et al. [2020; 2021] and described by § 2.2.1 below.

For a quad mesh mechanism, different positions of this mechanism should also be called "isometric". This however is a different kind of isometry because here we require that faces moving rigidly. Such isometries with rigid faces are discussed by § 2.2.2. They are much harder to achieve than checkerboard isometries.

2.2.1 Isometric Correspondence and Isometric Bending of Meshes. Following [Jiang et al. 2020], two combinatorially equivalent quad meshes M, M' are checkerboard-isometric, if for corresponding faces $v_0, \ldots, v_3$ and $v'_0, \ldots, v'_3$, the conditions

$$\begin{aligned} &\|v_0 - v_2\|^2 = \|v'_0 - v'_2\|^2, \; \|v_1 - v_3\|^2 = \|v'_1 - v'_3\|^2 \\ &\langle v_0 - v_2, v_1 - v_3\rangle = \langle v'_0 - v'_2, v'_1 - v'_3\rangle, \end{aligned} \tag{8}$$

hold. These equations have a geometric meaning expressed in terms of the parallelogram formed by edge midpoints $m_{01}, m_{12}, m_{23}, m_{30}$, where $m_{ij} = \frac{1}{2}(v_i + v_j)$. (8) means that this parallelogram is congruent to the corresponding parallelogram $m'_{01}, \ldots, m'_{30}$. Jiang et al. [2020; 2021] demonstrate that checkerboard-isometric meshes discretize an isometric mapping between surfaces.

The continuous bending of a continuous surface is approximated by a time-dependent quad mesh (V, E, F) evolving such that it remains in checkerboard-isometric correspondence to a fixed mesh. Then parallelograms inscribed in faces move rigidly, and for each face $f \in F$ there are $c_f, \bar{c}_f$ such that parallelogram vertices $\dot{m}_{ij}$ move according to

$$\dot{m}_{ij} = c_f \times m_{ij} + \bar{c}_f.$$

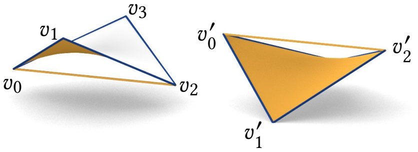


Fig. 4. Quads with the same edge lengths and diagonal lengths, but opposite orientation.

It is not difficult to see that these equations, if seen as a linear system for the variables $c_f, \bar{c}_f$, has the explicit solution

$$c_f = -\frac{(\dot{m}_{12} - \dot{m}_{01}) \times (\dot{m}_{23} - \dot{m}_{01})}{\langle m_{12} - m_{01}, \dot{m}_{23} - \dot{m}_{01}\rangle}, \quad \bar{c}_f = \dot{m}_{01} - c_f \times m_{01}. \tag{9}$$

The c_f's constitute the vertices of a mesh which is a combinatorial dual of the given mesh. It serves as discrete equivalent of the rotation diagram. Similarly, the $\bar{c}_f$'s serve as a discrete translation diagram.

2.2.2 Mesh Isometry with Rigid Faces. Two combinatorially equivalent quad meshes (V, E, F) and (V', E', F') are isometric *with rigid faces*, if for every face f and its corresponding face f' there is a rigid body motion transforming f onto f'. This relation between meshes is much more constrained than the checkerboard-isometric relation defined by § 2.2.1. It is the main focus of this paper.

The 'isometry with rigid faces' constraint implies that within each face, distances between corresponding vertices are the same:

$$c_{\text{len}}^{ij} = \|v_i - v_j\|^2 - \|v'_i - v'_j\|^2 = 0, \quad \text{for } v_i, v_j \in f. \tag{10}$$

Rigid body motions are orientation-preserving (see Fig. 4), so we require that for $f = v_0v_1v_2v_3$, also $c_{\text{vol},f} = 0$, where

$$c_{\text{vol},f} = \det(v_1 - v_0, v_2 - v_0, v_3 - v_0) - \det(v'_1 - v'_0, v'_2 - v'_0, v'_3 - v'_0). \tag{11}$$

2.2.3 Flexion of Quad Meshes — Quad Mesh Mechanisms. A flexion is a deformation of a time-dependent mesh (V', E', F') which stays isometric, with rigid faces, to a fixed mesh (V, E, F), see Fig. 5. Rigid body motions are flexions, they are called trivial. A mesh is flexible if a nontrivial flexion exists. We also say the mesh is a *mechanism* then – see Fig. 5. The key to reducing the complexity of the quad mesh flexibility problem is the following:

Proposition 2.2. *A regular quad mesh is flexible if and only if all its 3×3 submeshes are.*

Schief et al. [2008] proved this in the setting of quad meshes with planar faces, but their proof does not depend on planarity.

For a smooth flexion, each face f is endowed with vectors $c_f, \bar{c}_f$ such that

$$\dot{v}_i = c_f \times v_i + \bar{c}_f, \quad \text{for all } v_i \in f. \tag{12}$$

They can be computed in a way analogous to (9), namely as

$$f = (v_0v_1v_2v_3) \implies c_f = -\frac{(\dot{v}_1 - \dot{v}_0) \times (\dot{v}_2 - \dot{v}_0)}{\langle v_1 - v_0, \dot{v}_2 - \dot{v}_0\rangle},$$

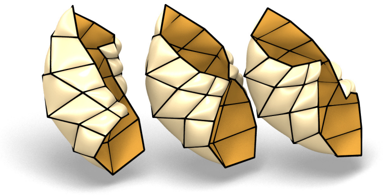


Fig. 5. Flexion of a quad mesh mechanism. The rigid faces are represented by bulky solids to emphasize that they are not planar.

and $\bar{c}_f = \dot{v}_0 - c_f \times v_0$.

A combinatorial dual of the mesh with the c_f's as vertices serves as discrete rotation diagram. Similarly, vectors $\bar{c}_f$ form a translation diagram. The mesh with vertices $\dot{v}_i$ and combinatorially equivalent to the given mesh, is called a velocity diagram. Properties of these auxiliary meshes are nicely similar to certain bendings of continuous surfaces. They are summarized below and are illustrated by Fig. 6.

Proposition 2.3. ([Sauer 1970]; mesh analogue of Prop. 2.1) . *For a smooth flexion of a quad mesh, the associated rotation diagram is reciprocal-parallel to the mesh. Further, the velocity diagram is a discrete-conjugate surface, i.e., its faces are planar. Its edges are orthogonal to the corresponding edges of the original mesh:*

$$\langle v_i - v_j, \dot{v}_i - \dot{v}_j \rangle = 0, \ \textit{whenever}\, v_i, v_j \in f,\ f \in F. \tag{13}$$

Proof. Consider a face f and vertices $v_i, v_j \in f$. Equ. (13) follows by differentiating $\|v_i - v_j\|^2 = \text{const}$. By Equ. (12),

$$v_i, v_j \in f \implies \dot{v}_i - \dot{v}_j = c_f \times (v_i - v_j). \tag{14}$$

Thus c_f is a normal vector of the face of the velocity diagram corresponding to f. As to the rotation diagram, assume that an edge v_0v_1 is the intersection of faces f, f'. We invoke (14) for both f, f':

$$\begin{aligned} \dot{v}_0 - \dot{v}_1 &= c_f \times (v_0 - v_1) = c_{f'} \times (v_0 - v_1) \\ \implies &(c_f - c_{f'}) \times (v_0 - v_1) = 0. \end{aligned}$$

This means that the edge v_0v_1 of the original mesh, and the corresponding edge $c_f c_{f'}$ in the dual rotation diagram are parallel. □

2.3 Infinitesimal Isometries

An infinitesimal bending of a surface is a snapshot of a smooth bending, capturing positions and velocities at a single time instant of interest, allowing the construction of rotation diagrams and similar, without the need for establishing a full time-continuous bending. Similarly an infinitesimal flexion of a mesh is defined. These notions will be important for our algorithmic approach.

Formally, an infinitesimal bending of a smooth surface $x(u, v)$ is a velocity vector field $\dot{x}(u, v)$, such that (1) holds. The constructions performed in § 2.1.1 still work for infinitesimal isometries, since they depend on $\dot{x}(u, v)$ only. There is a rotation diagram $c(u, v)$ and translation diagram $\bar{c}(u, v)$ obeying (2)–(4). Prop. 2.1 is still valid – actually, we could have formulated Prop. 2.1 as a statement about infinitesimal bendings.

Rigid body motions are called trivial bendings, and any infinitesimal bending $\dot{x}(u, v)$ which matches a rigid body motion is called trivial. For discrete surfaces, there is a similar concept:

Definition 2.4. *An infinitesimal flexion of a mesh (V, E, F) is an assignment of velocity vectors $\dot{v}_i$ to vertices v_i, such that* (13) *holds, i.e.,* $\langle v_i - v_j, \dot{v}_i - \dot{v}_j \rangle = 0$ *whenever v_i, v_j belong to the same face.*

The expressions we require to vanish come from the time derivatives of squared lengths $\|v_i - v_j\|^2$ such as occur in the constraints (10). This definition expresses our interpretation as a snapshot of a smooth flexion, capturing positions and first derivatives.

Every mesh has infinitesimal flexions, name the trivial ones originating from a rigid body motion of the whole mesh. They have the form $\dot{v}_i = c \times v_i + \bar{c}$, for certain fixed $c, \bar{c} \in \mathbb{R}^3$. An infinitesimally flexible mesh is one which has nontrivial infinitesimal flexions.

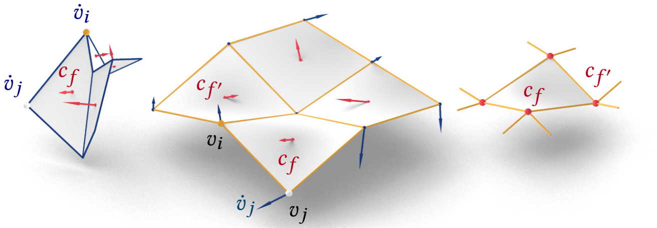


Fig. 6. *Center:* A flexing quad mesh with vertices v_i and angular velocities c_f of faces. *Left:* Velocity diagram with vertices $\dot{v}_i$ and normal vectors c_f of faces. Corresponding edge vectors $\dot{v}_i - \dot{v}_j$ and $v_i - v_j$ are orthogonal. *Right:* The rotation diagram has vertices c_f, for each face f. It is dual to the mesh, with corresponding edges $f \cap f'$ and $c_f, c_{f'}$ being parallel.

The constructions of § 2.2.3, namely of a velocity diagram, rotation diagram and translation diagram work, for infinitesimal flexions directly, because they need only first order derivatives. Prop. 2.3 is still valid. We state it again, with an extension:

Proposition 2.5. [Sauer 1970]. *For an infinitesimal flexion of a mesh, the associated velocity diagram has planar faces, and the associated rotation diagram is reciprocal-parallel to the mesh. Conversely, existence of a nondegenerate reciprocal-parallel mesh implies infinitesimal flexibility.*

Proof. For the converse, assume existence of vectors c_f and define velocities $\dot{v}_i$ by (12). It is easy to see that the infinitesimal flexibility condition follows from the reciprocal-parallel property. The word 'nondegenerate' means that the rotation diagram contains at least two different points. Then rotations associated with faces are not all the same, implying infinitesimal flexibility. □

2nd order flexibility. We also consider higher order infinitesimal flexions. Consider trajectories $v_i(t)$ and $v_j(t)$ of vertices belonging to the same face. In a smooth flexion, their distance remains constant for all times t, so all derivatives of $\|v_i(t) - v_j(t))\|^2$ w.r.t. t vanish:

$$\begin{aligned} c_{\text{fl},1,ij} &= \langle v_i - v_j, \dot{v}_i - \dot{v}_j \rangle = 0, \\ c_{\text{fl},2,ij} &= \langle v_i - v_j, \ddot{v}_i - \ddot{v}_j \rangle + \langle \dot{v}_i - \dot{v}_j, \dot{v}_i - \dot{v}_j \rangle = 0, \end{aligned} \tag{15}$$

and so on. In this paper we do not need order higher than 2. We define a mesh to be 2nd order flexible, if we can assign derivatives $\dot{v}_i, \ddot{v}_i$ to each vertex v_i, such that (15) holds whenever v_i, v_j belong to the same face. We further require that the 1st order derivatives $\dot{v}_i$ are not those of a rigid motion of the entire mesh.

There is debate in the rigidity community if our definition is the "right" one, but it serves for our purposes [Connelly and Servatius 1994]. Note that infinitesimal flexibility does not imply continuous flexibility.

Remark 2.6. In the special case of planar quad meshes, 2nd order flexibility has been nicely elaborated by Schief et al. [2008]. E.g. they show that then the above-mentioned rotation diagrams are discrete Bianchi surfaces with constrained Lelieuvre normal vectors. We do not use their work for our optimization tasks, but directly work with the simpler quadratic constraints of (15).

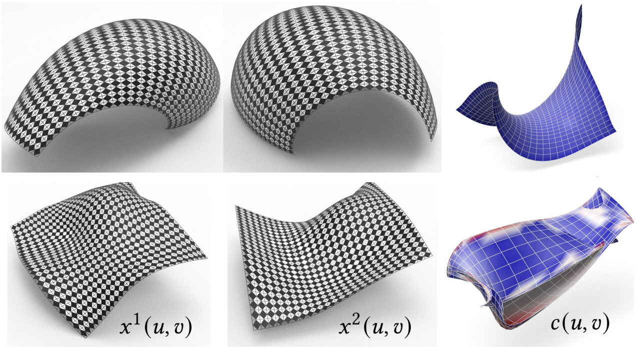


Fig. 7. Illustration of isometric surfaces $x^1(u,v)$ (left hand column) and $x^2(u,v)$ (center) and the associated rotation surface $c(u,v)$ at right. If x^1, x^2 are samples of a smooth bending to be approximated with a flexing quad mesh, $c(u,v)$ must obey further conditions. Thus we visualize Gaussian curvature by color coding (blue for $K<0$ and red for $K>0$).

2.4 Time-Discrete Bendings and Flexions

Above we considered either continuous isometries or infinitesimal isometries. It is equally important to consider sequences of surfaces or meshes which are isometric to each other. Any sample of a continuous bending, or a continuous flexion, yields such a sequence. We start our discussion with two meshes (V^1,E^1,F^1) and (V^2,E^2,F^2) which are isometric in the sense of rigid faces, thus being candidates for being part of a smooth flexion.

To study such mesh pairs, we introduce the vertices v_i^{avg} of the so-called middle mesh and Δv_i of the difference mesh:

$$v_i^{\text{avg}} = \frac{1}{2}(v_i^1 + v_i^2), \quad \Delta v_i = v_i^2 - v_i^1.$$

Distances of corresponding vertices are equal, if they are contained in the same face. Thus we immediately get

$$v_i, v_j \in f \implies \langle v_i^{\text{avg}} - v_j^{\text{avg}}, \Delta v_i - \Delta v_j \rangle = 0.$$

This so-called orthogonality relation is a discrete version of Equ. (13), difference vectors assuming the role of derivatives. It holds because the left hand expression expands to $\frac{1}{2}\left(\|v_i^1 - v_j^1\|^2 - \|v_i^2 - v_j^2\|^2\right)$. It directly follows that also an analogy of (12) holds: for each face f there are vectors $c_f, \bar{c}_f \in \mathbb{R}^3$ such that

$$\Delta v_i = \bar{c}_f + c_f \times v_i^{\text{avg}}. \tag{16}$$

This equation looks as if c_f is a vector of angular velocity, and in fact, if the meshes under consideration come from a dense sampling of a smooth flexion, then c_f approximates the angular velocities which occur in that flexion (up to scaling).

The analogy to the smooth case continues: the mesh with vertices Δv_i (the *difference diagram*) plays the role of the velocity diagram. Further there is a rotation diagram which is a combinatorial dual, having vertices c_f. Since (16) is formally equivalent to (12), the same conclusions hold, in particular the following time-discrete analogue of Prop. 2.3:

Proposition 2.7. *For a pair of quad meshes which are isometric in the rigid-faces sense, consider the middle mesh with vertices v_i^{avg}, the difference diagram with vertices Δv_i, and the associated rotation diagram with vertices c_f. Then middle mesh and rotation diagram are reciprocal-parallel, and the difference diagram has planar faces.*

A similar construction is possible for two smooth surfaces $x^1(u,v)$, $x^2(u,v)$ which are in isometric correspondence. Middle surface and difference surface are defined as $x^{\text{avg}} = \frac{1}{2}(x^1 + x^2)$ and $\Delta x = x^2 - x^1$. We have orthogonality between corresponding curves on $x^{\text{avg}}(u,v)$ and $\Delta x(u,v)$. There is a rotation surface $c(u,v)$ and a translation surface $\bar{c}(u,v)$ which fulfill relations analogous to (2) and (3), namely $\Delta x(u,v) = \bar{c}(u,v) + c(u,v) \times x^{\text{avg}}(u,v)$. In this work (e.g. in Figure 7), an isometric surface pair for computational purposes is represented by a mesh pair enjoying the checkerboard-isometry relation where inscribed parallelograms move rigidly (see § 2.2.1). Also here the mesh pair defines an average mesh and a difference mesh, together with an associated rotation diagram.

The similarity between continuous surfaces and discrete surfaces is relevant for later optimization tasks. E.g. when we are given two isometric surfaces x^1 and x^2 we may compute $\Delta x(u,v)$ and remesh it with planar faces, thus initializing an actual flexion of a quad mesh (see § 3.3). Figure 7 shows examples of rotation surfaces $c(u,v)$.

Since optimization for flexibility is a difficult task, we are interested in ways to identify promising initial values, and to point out situations where we cannot hope optimization to succeed. We use known properties of flexing meshes to do that. An example of this strategy is based on the following property:

Proposition 2.8. *Continuing Prop. 2.7, if the pair of isometric meshes has planar faces, then so does the middle mesh. The rotation diagram has negative curvature, in fact it even has planar vertex stars.*

Proof. For each face f^1, there is a rigid body motion $v_i^1 \mapsto v_i^2 = A_f v_i^1 + u_f$, mapping f^1 to the corresponding face f^2 of the 2nd mesh. The corresponding face of the middle mesh is the image under the affine mapping $v_i^{\text{avg}} = \frac{1}{2}(I + A_f)v_i^1 + \frac{1}{2}u_f$, showing planarity. Planar vertex stars follow directly from the reciprocal-parallel property. □

Figure 7 shows how to use Prop. 2.8. When attempting to use isometric surfaces x^1, x^2 to compute flexing meshes, we can disregard cases where the rotation diagram features positive curvature.

3 BASIC ALGORITHMS

Computing flexing quad meshes is a multi-step process, cf. Fig. 8. Flexibility is eventually established by optimization, but this requires an informed initialization. The preparations for the final step can be rather involved, which is why we dedicate an entire section to them.

In § 3.1 we find meshes which are isometric to each other and in addition are infinitesimally flexible. Such meshes are thought to be good candidates for a time-discrete isometric flexion. A special case (if only 1 mesh is involved) is to minimally deform a mesh such that it becomes infinitesimally flexible.

§ 3.2 deals with a given isometric correspondence between surfaces and converts them into quad meshes with planar faces, which are isometric to each other. These two meshes must follow a uniquely determined frame field and later serve to initialize the computation of a quad mesh mechanism.

§ 3.3 studies the same problem without the additional constraint of planar faces. Here we have more degrees of freedom, so we are

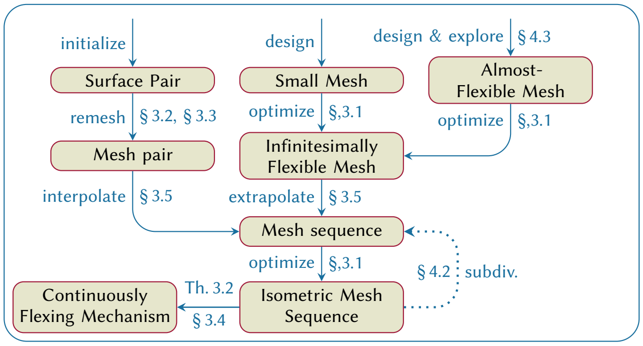


Fig. 8. In § 4 we describe three computational pipelines for producing quad mesh mechanisms. Their individual building blocks are the topic of §3. All pipelines end with an optimization making meshes both infinitesimally flexible individually and isometric to each other. From Theorem 3.2 we infer that we have created a mechanism.

able to find orthogonal cross fields on the given surfaces which determine a pair of isometric meshes. Likewise, they later serve to initialize the computation of a quad mesh mechanism.

§ 3.4 discusses algebraic properties of the configuration space and justifies our basic approach to computing a continuous flexion which is described by § 3.5. We actually compute a time-discrete flexion, which is a dense sequence of mutually isometric meshes which in addition are infinitesimally flexible.

3.1 Achieving Isometry Plus Infinitesimal Flexibility

Infinitesimal flexibility is a necessary condition for flexibility. Actually we are only interested in flexions that depend smoothly on time, with nonzero velocity, so we even consider 2nd order infinitesimal flexibility as necessary conditions. For this reason, we consider achieving infinitesimal flexibility as an essential preparation to initialize optimization towards flexibility. We here show how to solve this preparatory task.

Given a mesh (V, E, F) with vertices v_i, an infinitesimal flexion of order k means assigning derivatives $\dot{v}_i, \ddot{v}_i, \dots$ up to order k which fulfill Equations (15). They represent a quadratic system of constraints. To avoid zero solutions, we require that

$$c_{\text{fl,norm}} = \frac{1}{|V|} \sum\nolimits_{i\in V} \|\dot{v}_i\|^2 - 1 = 0. \tag{17}$$

To find a nontrivial infinitesimal flexion which does not correspond to a mere Euclidean motion, we fix a selected face f_0 by setting all derivatives to zero:

$$v_i \in f_0 \implies c^1_{\text{fix},i} = \dot{v}_i = 0, \quad c^2_{\text{fix},i} = \ddot{v}_i = 0, \tag{18}$$

up to order k. When modifying the given mesh such that it becomes infinitesimally flexible, a typical scenario is that we allow vertices to glide along a given shape Φ (which may be the initial mesh itself). To maintain closeness we need to be able to perform closest point projection onto Φ. We use the constraint

$$c_{\text{close}} = \sum\nolimits_{v_i\in V} \langle v_i - v_i^*, n_i^* \rangle^2 = 0, \tag{19}$$

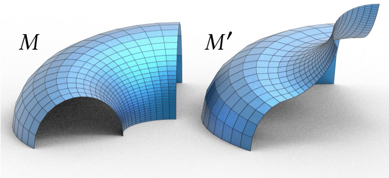


Fig. 9. Isometric positions of a mesh are not always connected by a continuous flexion, e.g in this case where part of M snaps upwards to form M'.

where $v_i^* \in \Phi$ is the point closest to v_i, and $n_i^* = \frac{v_i - v_i^*}{\|v_i - v_i^*\|}$ is a unit normal vector of Φ in that point. We do not treat v_i^*, n_i^* as variables in the optimization, but recompute them in each round of iteration (see e.g. [Tang et al. 2014]). If planarity of faces is to be maintained (this depends on the application) we introduce normal vectors n_f of faces and use the constraints

$$c^f_{\text{planar}} = \sum_{v_i, v_i\in f} \langle v_i - v_j, n_f \rangle = 0, \quad c^f_{\text{norm}} = \|n_f\|^2 - 1 = 0. \tag{20}$$

The objective function for optimization is a sum of squares of constraints, namely c_{len}, c_{vol}, $c_{\text{fl},m}$, $c_{\text{fl,norm}}$, c_{close}, c_{planar}, and c_{norm}. They are defined by Equations (10), (11) , (15), (17), (19), and (20), respectively. These constraints do not only involve derivatives, but also distances which express isometry of meshes. We therefore use the following general setup of energies.

$$\begin{aligned}
E_{\text{iso}}(M, M') &= \lambda_{\text{len}} \sum\nolimits_{f\in F} \sum\nolimits_{v_i,v_j\in f} (c^{ij}_{\text{len}})^2 + \lambda_{\text{vol}} \sum\nolimits_{f\in F} c^2_{\text{vol},f} \\
E_{\text{geom}}(M) &= \lambda_{\text{pl}} \sum\nolimits_{f} c^2_{\text{planar},f} + \lambda_{\text{n},1} \sum\nolimits_{f} c^2_{\text{norm},f} + \lambda_{\text{cl}} c^2_{\text{close}} \\
E_{\text{flex},k}(M) &= \sum\nolimits_{m=1}^{k} \lambda_{\text{fl},m} \sum\nolimits_{f\in F} \sum\nolimits_{v_i,v_j\in f} c^2_{\text{fl},m,ij} \\
&\quad + \lambda_{\text{fix}} \sum\nolimits_{m=1}^{k} \sum\nolimits_{v_i\in f_0} (c^m_{\text{fix},i})^2 + \lambda_{\text{n},2} c^2_{\text{fl,norm}}.
\end{aligned} \tag{21}$$

$E_{\text{iso}}(M, M')$ is zero exactly if M, M' are isometric to each other. The energy $E_{\text{geom}}(M)$ expresses planarity of faces of M, if required, and also proximity to a reference surface (for the proximity term we cannot expect zero residual). The energy $E_{\text{flex},k}(M)$ expresses that M is infinitesimally flexible of order k. Depending on the application, a certain linear combination of energies is minimized, using a Levenberg-Marquardt method. The following are typical application scenarios:

Initialize derivatives for a fixed mesh. Here we are given a mesh whose vertices v_i are considered fixed. We wish to find derivative vectors $\dot{v}_i, \ddot{v}_i, \dots$ up to order k such that infinitesimal flexibility constraints are fulfilled in the least-squares sense. If $k = 1$, these constraints are even linear and the optimization task is simple. Otherwise we minimize the energy $E_{\text{flex},k}$. It turns out that for this optimization, initial values have little influence. We use vertex normal vectors to initialize derivatives.

Optimization of one or more meshes. A task we need later is to optimize a mesh to become infinitesimally flexible, minimizing $E_{\text{geom}}(M) + E_{\text{flex},k}(M)$, typically for $k = 1$ or $k = 2$. Another particular application is the simultaneous optimization of a pair of meshes M, M' such that these two are both infinitesimally flexible and isometric to each other. We do this by minimizing $E_{\text{iso}}(M, M') + E_{\text{geom}}(M) + E_{\text{flex},k}(M) + E_{\text{flex},k}(M')$. Figure 9 shows a result. It turns out that in this case M is not able to continuously flex to reach M'. Only by using force, it snaps to the configuration M'. Such *bistable* pairs M, M' might be useful for applications in their own right, but they are not a focus of the present paper.

3.2 Flexibility Remeshing: Planar Quad Case

Here we aim at solving the following problem: Given are two geometric shapes Φ, Φ' which are connected by an isometry $f\colon \Phi \longrightarrow \Phi'$. Approximate both of them by quad meshes with planar faces which are isometric in the strict sense of rigid faces. This procedure, which is a first step in computing a continuous flexion from Φ into Φ', entails remeshing combined with shape optimization. The remeshing part was already proposed by [Jiang et al. 2021].

In order to better understand the nature of the isometry constraints, we look at a continuous analogy: Assuming fairness, a quad mesh with planar faces is a discrete version of a conjugate curve network on a surface. Equivalently, the edges of a quad mesh with planar faces represent a conjugate frame field on a surface: If vectors $t(p), \bar{t}(p)$ attached to a point $p \in \Phi$ locally represent this frame field, the conjugacy condition is

$$\mathrm{II}_p(t, \bar{t}) = 0,$$

where II_p is the symbol for the 2nd fundamental form of the surface Φ. The required pair of quad meshes then is a discrete analogy of two conjugate cross fields in Φ resp. Φ' which are mapped onto each other by the given isometry f.

For purposes of computation, isometric surfaces Φ, Φ' are represented by checkerboard-isometric quad meshes according to § 2.2.1. A 2nd fundamental form for this specific setting has been proposed by Ceballos Inza et al. [2023].

3.2.1 *A Checkerboard Second Fundamental Form.*

Fig. 10 shows two corresponding faces $f = v_0v_1v_2v_3$ and $f' = v_0'v_1'v_2'v_3'$ of quad meshes which are in the checkerboard-isometric relation. The inscribed parallelograms are also emphasized. These faces are endowed with two basis vectors, namely

$$b_1 = \frac{1}{2}\big((v_2 - v_3) + (v_1 - v_0)\big),\ b_2 = \frac{1}{2}\big((v_3 - v_0) + (v_2 - v_1)\big) \quad (22)$$

in f and analogous for f'. The basis vectors indicate diagonals in the inscribed parallelogram. The neighbouring face in the direction of b_1 is temporarily labelled f_1; the opposite neighbour is f_{-1}. Similarly we have faces f_2 and f_{-2}, see Fig. 11. These faces just mentioned have unit normal vectors $n, n_1, n_{-1}, n_2, n_{-2}$, respectively.

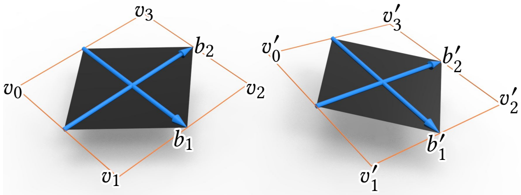


Fig. 10. Basis vectors b_1, b_2 associated with a face f and corresponding basis vectors b_1', b_2' associated with the corresponding face f'. They are diagonals in the respective inscribed parallelograms. If f, f' are related via a checkerboard isometry, there is a rigid body motion mapping basis vectors b_1, b_2 onto b_1', b_2'.

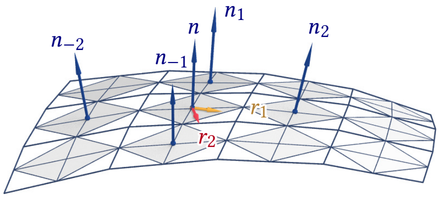


Fig. 11. Setting up the definition of the 2nd fundamental form by Equ. (22) to (25).

The 2nd fundamental form proposed by Ceballos Inza et al. [2023] is constructed from normal vectors. They argue why b_1 is conjugate to r_1 and b_2 is conjugate to r_2, where

$$r_1 = n \times (n_1 - n_{-1}), \quad r_2 = n \times (n_2 - n_{-2}). \quad (23)$$

Vectors r_1, r_2 can be expressed in the basis $\{b_1, b_2\}$ as

$$r_1 = \alpha_1 b_1 + \alpha_2 b_2, \quad r_2 = \beta_1 b_1 + \beta_2 b_2. \quad (24)$$

Evaluation of the 2nd fundamental form is now defined in terms of coordinates w.r.t. the basis $\{b_1, b_2\}$ as

$$\mathrm{II}_f(a, b) = a^T \cdot \Lambda_f \cdot b, \text{ where } \Lambda_f = \lambda \begin{bmatrix} \alpha_2\beta_2 & -\alpha_1\beta_2 \\ -\alpha_1\beta_2 & \alpha_1\beta_1 \end{bmatrix}. \quad (25)$$

We see that indeed $\mathrm{II}_f(b_1, r_1) = \mathrm{II}_f(b_2, r_2) = 0$. The factor λ is not relevant, since we only ever employ II_f to check if it evaluates to zero or not.

3.2.2 *Remeshing Guided By Isometric Conjugate Nets.*

Consider two quad meshes Φ, Φ' related by a checkerboard isometry. For each corresponding pair f, f' of faces, compute basis vectors, matrices Λ_f, Λ_f' and the second fundamental forms II_f, II_f' according to the previous paragraph. The basis (22) associated with f and the one associated with f' are connected by a rigid body motion, which follows directly from our definition of checkerboard isometry. In this way, any tangent vector t_f associated with f can canonically be moved to a corresponding tangent vector t_f' associated with f'.

We now look for a cross field, defined by *conjugate* tangent vectors $t_f, \bar{t}_f$, such that also its corresponding cross field is conjugate, i.e.,

$$\mathrm{II}_f(t_f, \bar{t}_f) = 0, \quad \mathrm{II}_f'(t_f', \bar{t}_f') = 0. \quad (26)$$

This is a generalized eigenvalue problem: We first solve for coefficients λ, μ such that $\det(\lambda\Lambda_f + \mu\Lambda_f') = 0$ and find vectors $t_f, \bar{t}_f$ in the kernel of this matrix. In case of positive curvature ($\det \Lambda_f > 0$) we always have a solution, but in case of negative curvature this is not guaranteed. For this reason we re-formulate our task as an optimization problem where also the vertices are variables. We employ the following variables:

- The vertices of two combinatorially equivalent quad meshes S, S' which are going to be isometric in the checkerboard way.
- Unit normal vectors n_f, n_f' in corresponding faces f, f'.
- For each face f, tangent vectors r_1, r_2, stored via their local coordinates according to § 3.2.1. Analogously, tangent vectors r_1', r_2' in the corresponding face f'.
- For each pair of corresponding faces f, f', the matrices Λ_f, Λ_f' of the respective 2nd fundamental forms.
- For each face, tangent vectors $t_f, \bar{t}_f$, are likewise stored via their local coordinates.

We impose the following constraints:

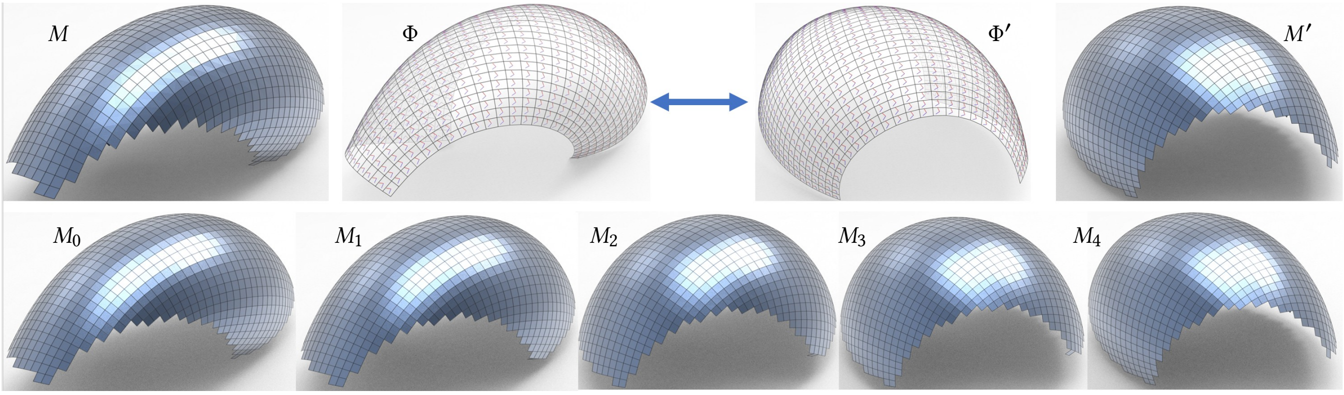


Fig. 12. Computation pipeline for quad mesh mechanisms with planar faces. A given isometry between surfaces Φ, Φ' is the basis for the computation of cross fields on these surfaces which guide quad meshes M, M' approximating Φ, Φ' (top row). A second round of optimization establishes a sequence of meshes $M_0, M_1, M_2, \dots, M_n$ which are isometric to each other in the sense of rigid planar faces. The sequence $\{M_j\}$ approximates a continuous bending of Φ.

— For each face, Equ. (8), expressing a checkerboard isometry.
— Basis vectors b_1, b_2 per face are introduced by (22). Equ. (22) serves as constraint. Similar for corresponding vectors b'_1, b'_2. Constraints (8) are equivalent to the statement that there is a rigid body motion mapping the basis b_1, b_2 to b'_1, b'_2.
— Normal vectors n_f are initialized by the basis vectors of (22) as their normalized cross product. In the course of optimization we maintain constraints $\langle n_f, b_1\rangle = \langle n_f, b_2\rangle = 0$, $\|n_f\|^2 = 1$.
— For each face f, vectors r_1, r_2 obey (23). For this purpose, we express r_1, r_2 as a linear combination of basis vectors, using (24) as constraint. Vectors r'_1, r'_2 associated with the corresponding face f' fulfill analogous constraints.
— The matrices related to 2nd fundamental forms are defined by the constraint (25); the conjugacy constraint (26) applies.
— Normalization constraints $\|t_f\|^2 = \|\bar{t}_f\| = 1$ prevent vectors $t_f, \bar{t}_f$ from fulfilling (26) by simply converging to zero.

Having computed the cross field $\{t_f, \bar{t}_f\}$, we extract a quad mesh M whose edges follow the cross field by the method of [Ebke et al. 2013]. The given isometry transfers it to a quad mesh M' approximating Φ'. Meshes M, M' are now candidates for being isometric in the sense of rigid planar faces. They serve as input for further optimization. An example is shown by Fig. 12.

3.3 Flexibility Remeshing: Skew Case

Similar to § 3.2, we are given an isometry between surfaces Φ_1, Φ_2 and want to find quad meshes M_1, M_2 which approximate Φ_1, Φ_2 and which are isometric with rigid faces in the sense of § 2.2.2. The difference to § 3.2 is that we do not require the faces to be planar. The computational setup is similar: Surfaces Φ_1, Φ_2 are given as quad meshes connected by a checkerboard isometry. Further we assume that faces are equipped with a 2nd fundamental form according to § 3.2.1, so we can check if vectors attached to a face are conjugate.

Our method is based on Prop. 2.7. The quad meshes we look for are guided by cross fields on Φ_1, Φ_2 which correspond in the given isometry. The cross field on Φ_1 is represented by two vectors $t_f, \bar{t}_f$ per face f. Prop. 2.7 involves the difference surface $\Phi' = \Phi_2 - \Phi_1$. This difference is computed vertex-wise. We use (22) to associate basis vectors b_1, b_2 to each face of Φ_1 and corresponding basis vectors b'_1, b'_2 to the corresponding face of Φ', thus defining a linear mapping between corresponding faces f, f'. Prop. 2.7 states that vector fields $t_f, \bar{t}_f$ in Φ_1 can guide the mesh M_1 only if the corresponding vector fields $t'_f, \bar{t}'_f$ in Φ' are conjugate. We thus require that

$$\langle t_f, \bar{t}_f\rangle = 0, \quad \mathrm{II}'_f(t'_f, \bar{t}'_f) = 0. \tag{27}$$

Similar to (26), this leads to an eigenvalue problem, which here is always solvable because the matrix of the inner product is positive definite.

The requirement that $t_f, \bar{t}_f$ are orthogonal expresses our wish that the edges of M_1 intersect nicely at right angles. We compute $t_f, \bar{t}_f$ by optimization in the same way as we computed vector fields in § 3.2.2, only the conjugacy constraint (26) is replaced by (27).

Similar to § 3.2, we extract a quad mesh M_1 from the cross field $\{t_f, \bar{t}_f\}$. The given isometry transfers this to M_2. Quad meshes M_1, M_2 are now candidates for being isometric in the sense of rigid faces and can serve as input for further optimization. An example is shown by Fig. 13.

We argue that this procedure yields sensible initial data for subsequent optimization, despite the face that Prop. 2.7 does not give a sufficient condition for isometry, only a necessary one: If surfaces Φ_1, Φ_2 are close, the difference diagram is a scaled version of a velocity diagram. In that case, Prop. 2.5 applies and reveals Φ_1 to be infinitesimally flexible. We therefore have good reason to believe that M_1, M_2 are indeed good initial data for a continuous flexion. Numerical experiments confirm this.

3.4 The Configuration Space of Flexible Meshes

3.4.1 The Configuration Space of Flexible 3×3 *Meshes With Planar Faces.* Izmestiev [2017] gave a complete classification of the flexible 3×3 quad meshes with planar faces. The knowledge of these 8 types (and their subcases) is relevant for various reasons. Specifically, in our design pipeline, we use 3×3 meshes to initialize the coarse-to-fine method of § 4.2.

All flexible cases are described by constraints imposed on the 16 angles $a_i, \dots, d_i$, $i = 1, 2, 3, 4$ which occur between the edges

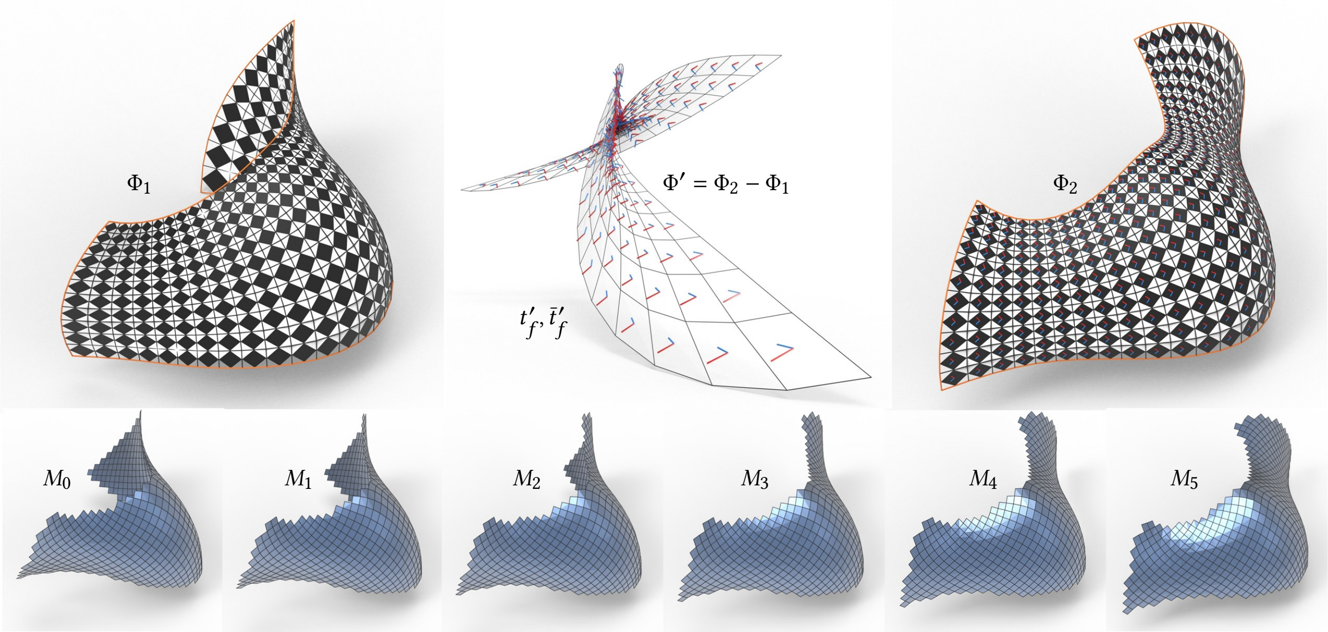


Fig. 13. Computation pipeline for a quad mesh mechanism in the general case (skew faces). Given are two isometric surfaces Φ_1, Φ_2, where the isometry is encoded as a checkerboard isometry (the figure illustrates the isometric inscribed parallelograms which occur in this situation). We compute the difference surface $\Phi' = \Phi_2 - \Phi_1$ and compute appropriate cross fields on Φ_1, Φ_2, Φ' according to § 3.3 (top row). These cross fields are the basis of remeshing; the resulting meshes undergo a round of optimization to produce a sequence $M_0, \ldots, M_n$ of meshes which are isometric with rigid faces. The sequence $\{M_k\}$ approximates a continuous flexion which approximately connects the given surfaces Φ_1, Φ_2.

incident with the central face, see Fig. 15, left. Apart from the well-known cases of T-nets and V-nets (types 1 and 2), finding a specific instance of a case typically requires the solution of a certain nonlinear system of equations in those 16 variables. This is harder than it looks because the solution varieties have different dimensions. For some of the cases, the solutions might not be suitable for practical applications (e.g. a certain sub-type of type 7, “chimeras”, essentially behave like a 2×2 mesh, exhibiting an almost rigid 2×2 sub-mesh). It turned out that certain subtypes (e.g., type 3, “equimodular elliptic”) are not straightforward to find numerically. Most of the results obtained by optimization exhibit type 5 (“linear compound” type). We mention that Erofeev and Ivanov [2020] explicitly construct special cases of type 3.

Remark 3.1. A way to test flexibility of an $n \times m$ quad mesh is based on Prop. 2.2: One needs to check if all $(n-2)(m-2)$ submeshes of size 3×3 are flexible. In the case of planar faces, we could use the classification of Izmestiev [2017] for that purpose. However, in our experience, for numerical reasons, this does not work if n, m exceeds, say, 7. Such a check is particularly hard if the mesh approximates a smooth surface in a fair way, because dihedral angles tend to be small, and opposite angles in vertex stars tend to be equal. We were able to benefit from the existing classification results only for small meshes. see § 4.3.1.

3.4.2 *The Algebraic Problem of Flexibility.*

All our computational pipelines to obtain flexible quad meshes are based on the computation of a sequence $M_0, M_1, \ldots, M_n$ of quad meshes which are isometric in the sense of rigid faces. We look only for flexions of meshes which approximate the isometric bending of a continuous surface. This has two reasons: Firstly the applications we have in mind demand fairness of the mesh, and secondly we want to use methods of differential geometry. To emphasize the analogy even more, we require that each mesh M_i is infinitesimally flexible of degree 2 – we imagine that a 2nd order infinitesimal flexion directly corresponds to the 1st and 2nd derivative vectors experienced by points in the surface while bending.

The question if a certain mesh allows a continuous flexion is algebraic in nature, since distance constraints are quadratic. There is hope that its degree is independent of the mesh size since Prop. 2.2 reduces the question to flexibility of every 3×3 submesh. Indeed, we are able to prove:

Theorem 3.2. *Assume that a quad mesh is given in N mutually isometric positions which enjoy property (A) as specified by Def. A.2. If N is large enough we can infer that the mesh is flexible.*

If the mesh is of generic dimensions and we have generic positions, $N = 33$ is sufficient. This number reduces to $N = 17$ in case of planar faces, and further to $N = 9$ and even $N = 6$ if the given positions are infinitesimally flexible of 1st order or even 2nd order.

This result is immediately relevant. It justifies our method of computing a flexion via a sequence of isometric meshes, it also says how long that sequence has to be. In practice we used $N = 20$ unless the theorem requires a higher number. The proof in the appendix uses an algebraic formulation of flexibility involving dihedral angles, which has recently been generalized to quad meshes with skew faces

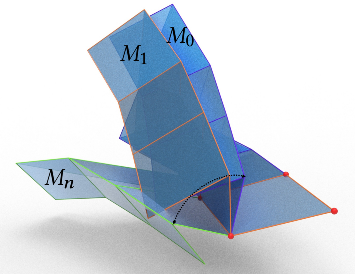


Fig. 14. Here the handle-driven method of § 3.5.2 is used to find further positions $M_1, \ldots, M_n$ of a 3×3 mesh M_0 already known to be flexible from the classification of [Izmestiev 2017].

by Aikyn et al. [2024]. If generic positions are not assumed, N is larger, see Prop. A.3–A.5.

3.4.3 Numerically computing flexions. Our method to find flexions of a mesh rests on the computation of a sequence M_k of mutually isometric meshes. Theorem 3.2 says that a flexion exists if sufficiently many meshes M_k are found. We think of those meshes as discrete samples of a continuous flexion $M(t)$.

A quad mesh mechanism has 1 degree of freedom in its flexion. It represents an algebraic curve in the configuration space. More degrees of freedom are only possible for trivial examples. Algebraic curves may have different branches. Likewise, a quad mesh mechanism might experience different flexions which branch off from a singular position, but we do not know if such examples exist (except for trivial cases). Our method of computing a sequence of meshes M_k certainly favors the case where all M_k are samples of a single smooth flexion. This is because we involved 1st and 2nd order derivatives in our algorithm. We therefore expected that standard numerical interpolation methods applied to M_k plus derivatives indeed yield a good approximation of $M(t)$. Our numerical experiments confirm this. In fact we saw that we basically get the same result whether or not we include the information on infinitesimal flexions. This is evidence for interpolation to yield good results.

3.5 Time-Discrete Flexions

Motivated by § 3.4, our approach to finding a continuous flexion is to establish a time-discrete flexion, i.e., a dense sequence $M_0, M_1, \ldots$ of meshes which are mutually isometric to each other and which themselves are infinitesimally flexible. Below, we list several methods for creating such sequences. In particular, we discuss interpolation and an iterative extrapolation method, which uses infinitesimally flexible meshes.

3.5.1 Extrapolation Based on Infinitesimal Flexibility. We start with a mesh $M_0 = (V, E, F)$ which is 2nd order flexible (e.g. computed by the method of § 3.1). For each vertex $v_i \in V$, 1st and 2nd order derivatives $\dot{v}_i, \ddot{v}_i$ are given. Then, the new mesh $M(t)$ with vertices $v_i(t) = v_i + t\dot{v}_i + \frac{t^2}{2}\ddot{v}_i$ is almost isometric to M_0 in the sense that distances $\|v_i(t) - v_j(t)\|$ of vertices are not exactly constant, but their 1st and 2nd derivatives vanish. We now choose discrete time instances $t_1, t_2, \ldots,$ define meshes $M_i = M(t_i)$ and perform a 2nd round of optimization such that all M_i are isometric to M_0, using the procedure of § 3.1.

3.5.2 Handle-Driven Flexion. A method suitable to compute the flexion of a small mesh is to select vertices v_i, v_j, not in the same face, and force their distance to change. We use the optimization procedure of § 3.1, augmented by the additional constraint $\|v_i - v_j\|^2 = d_{ij}^2$, for different values of d_{ij}. In this way, we simultaneously compute two or more meshes which are mutually isometric, but which are not related by a Euclidean motion — see Fig. 14.

3.5.3 Interpolation. A third method is based on two initial positions M, M' which have been optimized for isometry. Linear interpolation of vertices yields a sequence of intermediate positions $M_0 = M, M_1, M_2, \ldots, M_n = M'$ which are being simultaneously optimized for being isometric, using energies $E_{\text{iso}}(M_0, M_1) + \cdots + E_{\text{iso}}(M_0, M_n)$. Optimization is applied to all involved meshes simultaneously. We cannot keep the initial meshes fixed here, because then we would already assume that the given mesh M is a mechanism.

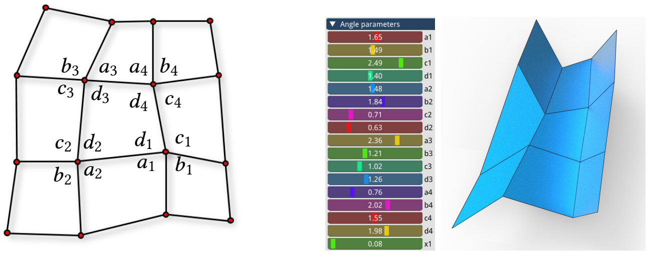
4 DESIGN PIPELINES FOR QUAD MESH MECHANISMS

4.1 The Remeshing Pipeline

This pipeline starts with finding an isometric correspondence between surfaces Φ, Φ' which for computational purposes is represented in the way described by § 2.2.1. It has three steps in total:

1. Find isometric surfaces Φ, Φ', e.g., using the method of [Jiang et al. 2020].
2. Remeshing (§§ 3.2, 3.3) creates isometric meshes M, M'
3. Interpolation followed by optimization (§ 3.5) yields a time-discrete flexion

Examples are shown by Fig. 12 (which has planar faces) and Fig. 13 (skew faces). However, this pipeline is not always successful. Its limitations are the following. Firstly, remeshing for quad meshes does not always work. One has to check if the condition of Prop. 2.8 (see Fig. 7) is fulfilled. Otherwise, the remeshing procedure will not be able to find the necessary frame field (a certain quadratic equation will have only complex solutions). Secondly, we might not be able to find a time-discrete flexion from meshes M, M': The simultaneous optimization of an interpolating sequence M_k might not work at all or create meshes too far away from the initial surfaces Φ, Φ'. The fact of a smaller than 100% success rate is actually to be expected since quad mesh mechanisms are rare. It is certainly not the case that any isometric pair Φ, Φ' is close to a mechanism. The pipeline above is based on necessary conditions that might, or might not, let the final round of optimization succeed.



▼ Angle parameters

1.65	a1
1.49	b1
2.49	c1
1.40	d1
1.48	a2
1.84	b2
0.71	c2
0.63	d2
2.36	a3
1.21	b3
1.02	c3
1.26	d3
0.76	a4
2.02	b4
1.55	c4
1.98	d4
0.08	x1

Fig. 15. This image shows a user interface of our design tool for flexible 3 by 3 quad meshes with planar faces. The parameters shown on the sliders are the 16 angles incident with the central face – see figure at left. The 17th parameter shown is a dihedral angle and can be used to drive the flexion.

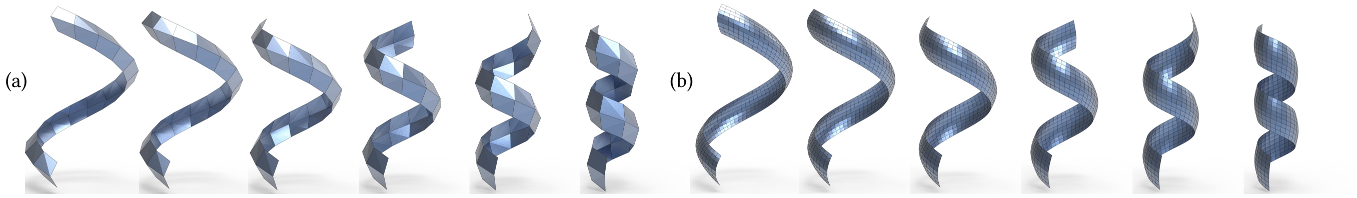


Fig. 16. A quad mesh of size 2 × n is naturally flexible. We start with 6 discrete samples of a continuous flexion (a) and apply two rounds of subdivision plus optimization. This yields a sequence of isometric positions of a quad mesh, indicating the existence of a continuous flexion (b).

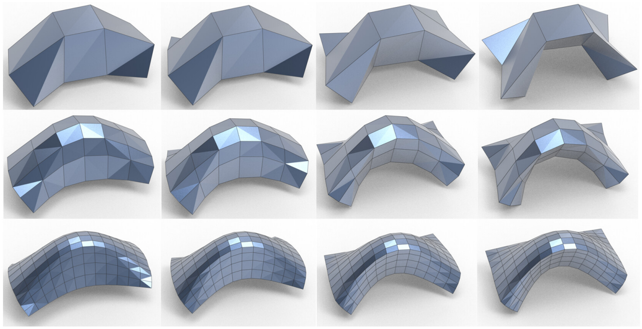


Fig. 17. Isometric positions of meshes generated by refinement plus optimization. The top row shows four isometric positions of a 3 × 3 quad mesh with nonplanar faces generated by interpolation (see § 3.5.3). We then apply Catmull-Clark subdivision followed by optimization to create further rows.

4.2 The Coarse-To-Fine Pipeline

The general idea behind this approach is that optimization of meshes towards specific properties is generally easier for small meshes. A refinement operation produces a denser mesh and generally destroys this property. It is subsequently restored by another round of optimization, and this procedure can be iterated. Examples are shown by Figures 16 and 17, where a sequence of mutually isometric small meshes undergoes Catmull-Clark subdivision and is again optimized for being isometric. After two rounds, we get a sequence of finer meshes. In Fig. 16 we started from meshes of size 2×n which are always flexible and used the method of § 3.5.2 to find additional positions. The starting sequence of Fig. 17 has been generated by applying the optimization procedure of § 3.1.

4.3 Exploring the Configuration Space of Flexible Meshes

Flexible meshes with planar faces have been the object of study for some time, being also embedded into the framework of Discrete Differential Geometry.

4.3.1 Flexible 3×3 *Meshes With Planar Faces.* In § 3.4.1 we described the classification of flexible meshes with planar faces by Izmestiev [2017].

Fig. 15 shows examples of type 6, sub-type 1 (“linear conjugate, antideltoid”). The corresponding system of constraints is shown in Appendix B. Such a system, written as $F(X) = 0$, is solved by a regularized Newton method, moving an initial guess X_0 close to the solution variety in an iterative way. Linearization $F(X_0 + h) \approx F(X_0) + \frac{\partial F}{\partial X} \cdot h$, where $\frac{\partial F}{\partial X}$ is the Jacobi matrix of F, leads to the linear system $\frac{\partial F}{\partial X} h = -F(X_0)$. In this particular case, it represents 11 linear equations in 16 variables. We regularize by solving for $\|h\| \longrightarrow \min$, which amounts to solving the system by applying the Moore-Penrose pseudoinverse of $\frac{\partial F}{\partial X}$ to the right-hand side.

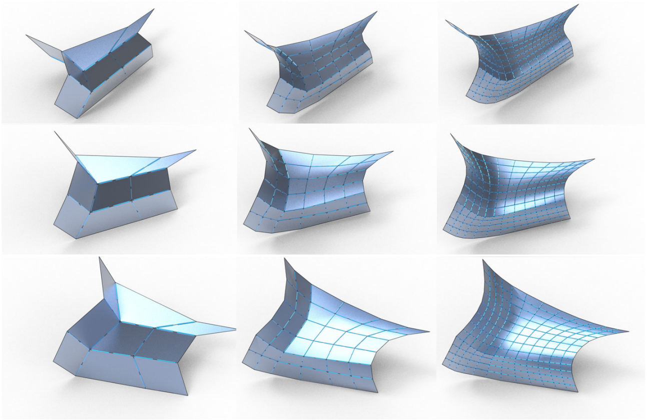


Fig. 18. A quad mesh mechanism created by our coarse-to-fine computational pipeline applied to a flexible 3 × 3 quad mesh with planar faces (left column). Both planarity and flexibility are lost when subdivision is applied; flexibility is restored by optimization (middle and right hand columns). The initial 3 × 3 mesh has been found with an interactive design tool based on the known classification of flexible 3 × 3 meshes with planar faces.

The design tool visualized by Fig. 15 allows the user to specify the 16 angle variables and to optimize towards a certain type of flexible mesh. The result can serve as input for the coarse-to-fine design pipeline. Fig. 18 shows an example.

4.3.2 Elementary Manipulations of Meshes with Planar Faces. For meshes with planar faces, flexibility only depends on the angles in vertices, which for a 3 × 3 mesh are labeled in Fig. 15. It follows that such a mesh is flexible if and only if all its parallel meshes (Combescure transforms) are [Schief et al. 2008, Th. 3.1]. This property is proved by reduction to the 3 × 3 case via Prop. 2.2, and some elementary geometry. It yields quite some design freedom: One can select two transverse mesh polylines and arbitrarily scale the edge vectors on them. Fig. 21 shows an example of this.

4.3.3 Optimization-driven Exploration. The explicitly known classes of flexible meshes are too unwieldy and too restricted to be directly useful for interactive design, e.g. for a freeform architecture application one does not want to work with T-meshes and their trapezoidal faces only. On the other hand, flexibility is a very strong condition,

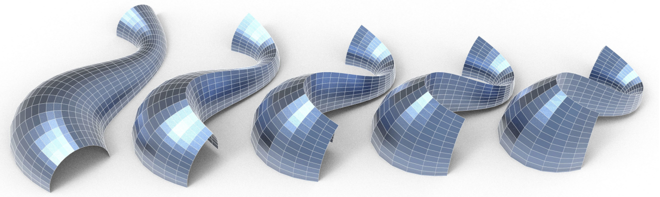


Fig. 19. Quad mesh mechanism created with the method of § 4.3.3. A mesh has been designed such that faces are planar and one family of mesh polylines lies in vertical planes. This is about half of the conditions needed for a T-net, so we cannot expect flexibility. With the method of § 3.5.2, we created a sequence of meshes which was successfully optimized simultaneously. The result is a flexible quad mesh which is not a T-net. It apparently develops a crease during its flexion, similar to the behaviour of some T-nets, cf. [Izmestiev et al. 2023, Fig. 18].

and can usually not be achieved by optimization of general meshes. For this reason we experimented with the idea of designing meshes which fulfill some, but not all of the constraints that define a flexible mesh class. In the configuration space of all meshes we have thus moved away from the sub-variety of flexible meshes, but not too much. Flexibility can be restored by optimization. We found this procedure to work much better than initially expected. Figures 1 and 19 show examples. Also the small example of Figures 5 and 23 has been created by the same idea: Stachel [2011] found flexible meshes inscribed in a cylinder of revolution. Applying a gentle nonlinear transformation (in this case, a Möbius transform) destroys flexibility, which is subsequently restored by optimization.

5 DISCUSSION

The previous sections already showed results obtained by the computational pipelines described above. To verify our results, we built physical models of quad meshes we computed and checked their flexibility experimentally, see Figures 22 and 23. We also demonstrated that our methods work on meshes with combinatorial singularities, see Fig. 20.

Experimental evidence shows clearly that the results we obtained do not fall into the “easy” cases of Voss nets and T-nets, see Fig. 21.

Implementation Details. For optimization, we employ a Levenberg-Marquardt method according to [Madsen et al. 2004, §3.2], using a damping parameter of 10^{-6}. Our implementation in C++ uses the data structures of *OpenMesh* [Botsch et al. 2002] and the Taucs library for sparse linear solvers [Toledo 2003]. Table 1 gives statistics on the number of variables, parameters used in the optimization, the quality of results, and computation times. The latter refers to an Intel Xeon W-2225 4.1GHz processor with 32G RAM, without parallel processing or other acceleration techniques.

In our experience, our optimization is not very sensitive to the choice of parameters $\lambda_{\text{len}}, \ldots,$ as long as the terms in the target functional which correspond to individual constraints are of the same order or magnitude. Evidence of this is that we could use the same values in all examples. Some constraints, like those related to planarity of faces, do not apply to all examples, in which case the respective parameters are set to zero. Proximity to a reference geometry is a soft constraint, and consequently the parameter λ_{cl} is smaller by 2 orders of magnitude than the others.

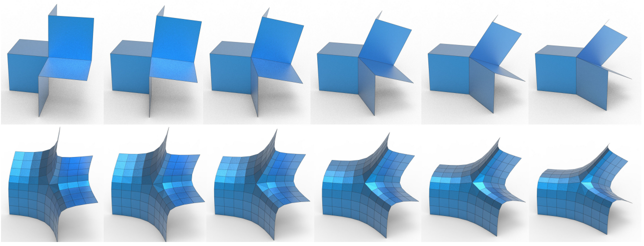


Fig. 20. A quad mesh mechanism with a combinatorial singularity. The first row shows samples of a continuous flexion of a single vertex star consisting of 6 quadrilateral faces (which is always flexible). We apply two rounds of Catmull-Clark subdivision plus optimization for isometry. This yields a dense sequence of isometric positions of a quad mesh, indicating the existence of a continuous flexion.

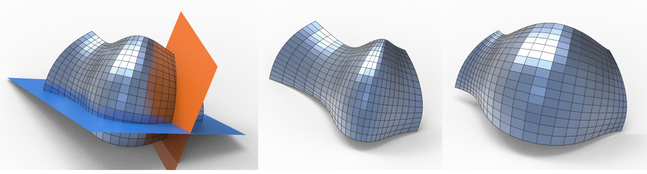


Fig. 21. *Left:* A flexing quad mesh with planar faces generated using the remeshing pipeline of § 4.1. *Center, Right:* Interactive modification via parallel meshes (Combescure transforms). The image at left also displays that mesh polylines are not contained in planes (both the orange and blue planes are spanned by 3 successive vertices). This shows the mesh is no T-net, which is one of the easily generated flexible types.

Ablation Study. We used the example of Fig. 12 to investigate the effects of weights in the combined optimization energy proposed in § 3.1. By setting certain weights to zero we obtain a less than optimal result. E.g. letting $\lambda_{\text{cl}} = 0$ yields a result which is not as close to the reference surface as otherwise, and similar for other changes which have an effect on the planarity of faces, and the isometric correspondence between meshes. The details and statistics are displayed in Table 2.

Limitations. A main limitation of our methods is that we have only a posteriori numerical evidence that they work in each single instance. This is particularly the case for the remeshing pipeline, which is based only on necessary conditions for flexibility, not on sufficient ones. Its limitations are in detail described at the end of § 4.1. The coarse-to-fine pipeline in our experience is more reliable. The limitations are that in some cases indeed it does not work (i.e., the optimization does not succeed), and in other cases, the result may be significantly different from the input data. Apart from these fundamental limitations, which lie in the difficulty of the problem itself, there are also limitations of our implementation. E.g. we did not implement collision detection.

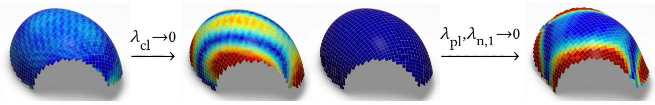
Failure cases. The problem of computing quad mesh mechanism is difficult because there is no theoretical result on shapes that are realizable as mechanisms (such a result would be a significant

Table 1. Statistics on the number of vertices and faces, the weights used in optimization, the number of iterations, the total time T in seconds, and the quality of the results. We measure isometry between two meshes M, M' by means of distances $l_{ij} = \|v_i - v_j\|$ of vertices which belong to the same face. Lengths pertaining to M resp. M' are stored in vectors L, L'. We consider the relative ℓ^2 error $E_2 = \|L - L'\|_2 / \|L\|_2$ and also the maximum $E_{\max}$ of $|l_{ij} - l'_{ij}|/l_{ij}$. In figures with multiple instances $M_0, \ldots, M_n$ of a flexing mesh, we compare M_0 to $M_1, \ldots, M_n$.

Fig	$\lvert V\rvert$	$\lvert F\rvert$	λ_{len}	λ_{vol}	λ_{pl}	$\lambda_{\text{n,1}}$	λ_{cl}	$\lambda_{\text{fl,1}}$	$\lambda_{\text{fl,2}}$	λ_{fix}	#it	T[s]	$E_{\max}$	E_2
18a	16	9											$2.4\cdot10^{-07}$	$5.8\cdot10^{-08}$
18b	49	36	1	1	0	0	0	1	1	1	10	0.12	$4.4\cdot10^{-05}$	$1.2\cdot10^{-05}$
18c	169	144	1	1	0	0	0	1	1	1	10	0.80	$1.1\cdot10^{-04}$	$1.3\cdot10^{-05}$
9	496	450											$1.5\cdot10^{-06}$	$2.8\cdot10^{-07}$
12	1048	962	1	1	1	1	.01	1	1	1	10	25.6	$6.0\cdot10^{-05}$	$5.2\cdot10^{-06}$
13	745	657	1	1	0	0	.01	1	1	1	10	10.4	$1.8\cdot10^{-05}$	$2.1\cdot10^{-06}$
16a	51	32											$6.0\cdot10^{-07}$	$2.8\cdot10^{-07}$
16b	585	512	1	1	0	0	0	1	1	1	10	4.47	$7.0\cdot10^{-05}$	$1.0\cdot10^{-05}$
16c	51	32											$6.5\cdot10^{-07}$	$3.2\cdot10^{-07}$
16d	165	128	1	1	1	1	0	1	1	1	10	0.51	$4.9\cdot10^{-05}$	$2.5\cdot10^{-05}$
17a	16	9	1	1	0	0	0	1	1	1	10	0.04	$8.2\cdot10^{-07}$	$4.5\cdot10^{-07}$
17b	49	36	1	1	0	0	0	1	1	1	10	0.11	$4.5\cdot10^{-05}$	$1.3\cdot10^{-05}$
17c	169	144	1	1	0	0	0	1	1	1	10	0.78	$6.2\cdot10^{-05}$	$8.7\cdot10^{-06}$
20a	13	6											$1.6\cdot10^{-05}$	$8.6\cdot10^{-08}$
20b	121	96	1	1	0	0	0	1	1	1	10	0.37	$1.2\cdot10^{-04}$	$2.4\cdot10^{-05}$
22	16	9	1	1	0	0	0	1	1	1	10	0.04	$1.7\cdot10^{-05}$	$4.2\cdot10^{-06}$
5,23l	36	25	1	1	0	0	0	1	1	1	10	0.18	$6.4\cdot10^{-05}$	$1.8\text{e}\cdot10^{-06}$
23r	441	400	1	1	0	0	0	1	1	1	10	3.82	$1.3\cdot10^{-04}$	$2.8\cdot10^{-05}$

Table 2. Data for an ablation study based on the example of Fig. 12. We measure how properties of computed meshes $M_1, M_2, \ldots$ deteriorate when setting corresponding weights to zero. Isometry is measured as described in in Table 1. Planarity of meshes is measured by the distance of diagonals in faces, normalized by average diagonal length. Closeness to a reference surface Φ is measured by distances of vertices to Φ, normalized by bounding box diagonal. The table below gives the ℓ^2 average and also the maximum values. Data for closeness and planarity are visualized by color-coding below.

weight change	effect	ℓ^2 average	maximum values
$\lambda_{\text{len}}, \lambda_{\text{vol}} \rightarrow 0$	isometry	$5.2\cdot10^{-6} \rightarrow 7.8\cdot10^{-3}$	$6.0\cdot10^{-5} \rightarrow 2.5\cdot10^{-2}$
$\lambda_{\text{pl}}, \lambda_{\text{n,1}} \rightarrow 0$	planarity	$9.2\cdot10^{-6} \rightarrow 6.3\cdot10^{-3}$	$4.9\cdot10^{-5} \rightarrow 2.8\cdot10^{-2}$
$\lambda_{\text{cl}} \rightarrow 0$	closeness	$7.2\cdot10^{-4} \rightarrow 2.2\cdot10^{-3}$	$2.6\cdot10^{-3} \rightarrow 6.6\cdot10^{-3}$
$\lambda_{\text{fl},i}, \lambda_{\text{fix}} \rightarrow 0$	isometry	$5.2\cdot10^{-6} \rightarrow 7.3\cdot10^{-4}$	$6.0\cdot10^{-5} \rightarrow 3.5\cdot10^{-3}$


$\xrightarrow{\lambda_{\text{cl}} \rightarrow 0}$
$\xrightarrow{\lambda_{\text{pl}}, \lambda_{\text{n,1}} \rightarrow 0}$

breakthrough). In our experiments we worked with surfaces of disk topology which have no or only a few simple features, such as the architectural example of Fig. 1. Mechanisms tend to possess hinges in direction of feature curves, ruling out complex arrangements of those. Likewise, several bump-like features are problematic as well. With too complex reference surfaces, optimization will either not succeed at all or converge to a different, simpler, shape. Our different ways of initialization have the purpose of finding good input meshes, and the presented optimization is well suited for shape exploration by moving away from known classes of quad mesh mechanisms (see § 4.3.3).

Future Research. Our numerical results point to interesting theoretical questions of real algebraic geometry. For the case of planar faces, an unsolved question is how to build larger meshes from 3 × 3 building blocks, where partial results have been achieved by He and Guest [2020]. Generally, the topic of computational design of flexible structures is by no means exhausted. Even if we feel we have made a significant contribution, the question of approximability of given shapes by flexible quad meshes is still unsolved. The generalization to other kinds of flexible structures is wide open. The mesh mechanism problem is easier in the degenerate metric of isotropic geometry, and the classification of isotropic-flexible meshes is a topic of a forthcoming publication. Apparently they can be efficiently used to initialize the computation of quad mesh mechanisms.

Conclusion. In this work we combine algebraic and numerical methods to compute quad meshes which exhibit a continuous isometric flexion where faces stay rigid and edges act as hinges. A novel degree bound is the basis for time-discretization, reducing the problem to computing a feasible finite number of isometric meshes. On the other hand, we use insights from numerical computations as pointers to interesting cases and to future research.

ACKNOWLEDGMENTS

This research was partially funded by the National Natural Science Foundation of China under grant number 62088102. Helmut Pottmann was supported by KAUST baseline funding.

REFERENCES

Alisher Aikyn, Yang Liu, Dmitry A Lyakhov, Florian Rist, Helmut Pottmann, and Dominik L Michels. 2024. Flexible Kokotsakis meshes with skew faces: Generalization of the orthodiagonal involutive type. *Computer-Aided Design* 168 (2024), 103669.

Thomas Bächler, Vladimir Gerdt, Markus Lange-Hegermann, and Daniel Robertz. 2012. Algorithmic Thomas decomposition of algebraic and differential systems. *Journal of Symbolic Computation* 47, 10 (2012), 1233–1266.

Eric Baldwin. 2018. SOM Designs Kinematic Sculpture for Chicago Design Week. ArchDaily (Jan 19). https://www.archdaily.com/904506

Mario Botsch, Stephan Steinberg, Stephan Bischoff, and Leif Kobbelt. 2002. OpenMesh: A Generic and Efficient Polygon Mesh Data Structure. Proc. OpenSG Symposium. https://graphics.uni-bielefeld.de/publications/openmesh.pdf

Sebastien Callens and Amir Zadpoor. 2018. From flat sheets to curved geometries: Origami and kirigami approaches. *Materials Today* 21, 3 (2018), 241–264.

Victor Ceballos Inza, Florian Rist, Johannes Wallner, and Helmut Pottmann. 2023. Developable Quad Meshes and Contact Element Nets. *ACM Trans. Graph.* 42, 6 (2023), 183:1–13.

Yan Chen, Rui Peng, and Zhong You. 2015. Origami of thick panels. *Science* 349 (2015), 396–400. (Issue 6246).

Albert Chern, Felix Knöppel, Ulrich Pinkall, and Peter Schröder. 2018. Shape from Metric. *ACM Trans. Graph.* 37, 4 (2018), 63:1–17.

Robert Connelly. 1987. Infinitesimal Rigidity. in: Theory of rigid structures (unpublished collection). http://pi.math.cornell.edu/~connelly/rigidity.chapter.2.pdf

Robert Connelly and Herman Servatius. 1994. Higher-order rigidity — What is the proper definition? *Discrete Comput Geom* 11 (1994), 193–200.

Xiangxin Dang, Fan Feng, Paul Plucinsky, Richard D. James, Huiling Duan, and Jianxiang Wang. 2022. Inverse design of deployable origami structures that approximate a general surface. *Intl. J. Solids and Structures* 234-235, Article 111224 (2022), 18 pages.

Erik Demaine and Joseph O'Rourke. 2007. *Geometric Folding Algorithms.* Cambridge Univ. Press.

Levi H Dudte, Etienne Vouga, Tomohiro Tachi, and L Mahadevan. 2016. Programming curvature using origami tessellations. *Nature materials* 15, 5 (2016), 583.

Hans-Christian Ebke, David Bommes, Marcel Campen, and Leif Kobbelt. 2013. QEx: Robust Quad Mesh Extraction. *ACM Trans. Graph.* 32, 6, Article 168 (2013), 10 pages.

Ivan Erofeev and Grigory Ivanov. 2020. Orthodiagonal anti-involutive Kokotsakis polyhedra. *Mechanism and Machine Theory* 146, Article 103713 (2020), 18 pages.

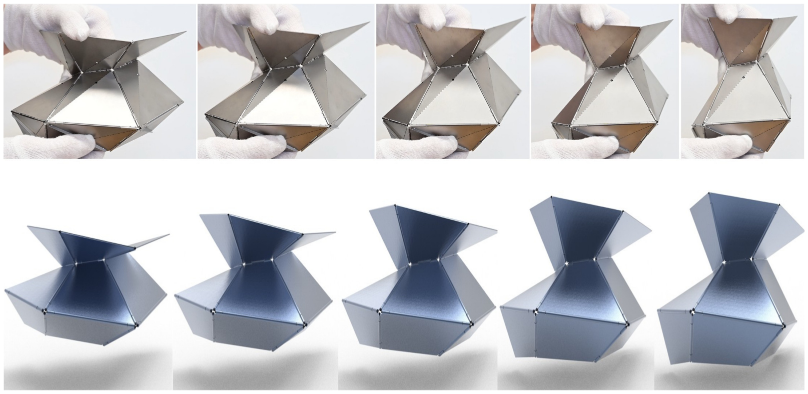


Fig. 22. Precision model of a 3×3 flexible mesh. Optimization was based on the interpolation method of § 3.5.3. In the rendering, skew faces are filled in by a bilinear interpolant. The built mechanism has a size 20cm×20cm×15cm. Faces were laser cut from 0.8 mm hard-rolled stainless steel and manually bent along the diagonal until they fit a bracing element of the same material. The knuckle and pin parts of the hinges were CNC cut (from a 2.5 mm cold drawn stainless steel seamless capillary pipe, resp. 1.5 mm carbon spring steel wire). Bracing and knuckles were fused to faces by pulse TIG welding. As to tolerances, the knuckle/pin fit has a clearance between 0 and 64 μm. The resulting mechanism is stiff, yet easy to move, without any discernible backlash or play in the joints. Its flexibility is solely due to geometric properties.

Thomas A. Evans, Robert J. Lang, Spencer P. Magleby, and Larry L. Howell. 2015a. Rigidly foldable origami gadgets and tessellations. *R. Soc. Open Sci.* 2, 9, Article 150067 (2015), 18 pages.

Thomas A. Evans, Robert J. Lang, Spencer P. Magleby, and Larry L. Howell. 2015b. Rigidly foldable Origami Twists. In *Origami* ^{6}. Vol. 1. American Math. Soc, 119–130.

Fan Feng, Xiangxin Dang, Richard D. James, and Paul Plucinsky. 2020. The designs and deformations of rigidly and flat-foldable quadrilateral mesh origami. *J. Mechanics and Physics of Solids* 142, Article 104018 (2020), 23 pages.

Ruslan Guseinov, Eder Miguel, and Bernd Bickel. 2017. CurveUps: Shaping Objects from Flat Plates with Tension-Actuated Curvature. *ACM Trans. Graph.* 36, 4 (2017), 64:1–12.

Zeyuan He and Simon D. Guest. 2020. On rigid origami II: quadrilateral creased papers. *Proc. R. Soc. A* 476, Article 20 (2020), 19 pages.

Chuck Hoberman. 2006. Transformation in Architecture and Design. In *Transportable Environments 3.* Taylor & Francis, 70–79.

Ivan Izmestiev. 2017. Classification of flexible Kokotsakis polyhedra with quadrangular base. *Int. Math. Res. Not.* 3 (2017), 715–808.

Ivan Izmestiev, Arvin Rasoulzadeh, and Jonas Tervooren. 2023. Isometric Deformations of Discrete and Smooth T-surfaces. arxiv:2302.08925.

Amin Jamalimehr, Morad Mirzajanzadeh, Abdolhamid Akbarzadeh, and Damiano Pasini. 2022. Rigidly flat-foldable class of lockable origami-inspired metamaterials with topological stiff states. *Nature Commun.* 13, Article 1816 (2022), 14 pages.

Caigui Jiang, Klara Mundilova, Florian Rist, Johannes Wallner, and Helmut Pottmann. 2019. Curve-pleated structures. *ACM Trans. Graph.* 38, 6 (2019), 169:1–13.

Caigui Jiang, Cheng Wang, Florian Rist, Johannes Wallner, and Helmut Pottmann. 2020. Quad-mesh based isometric mappings and developable surfaces. *ACM Trans. Graph.* 39, 4 (2020), 128:1–13.

Caigui Jiang, Hui Wang, Victor Ceballos Inza, Felix Dellinger, Florian Rist, Johannes Wallner, and Helmut Pottmann. 2021. Using isometries for computational design and fabrication. *ACM Trans. Graphics* 40, 4 (2021), 42:1–42:12.

Martin Kilian, Aron Monszpart, and Niloy J. Mitra. 2017. String Actuated Curved Folded Surfaces. *ACM Trans. Graph.* 36, 3 (2017), 25:1–13.

Antonios Kokotsakis. 1933. Über bewegliche Polyeder. *Math. Ann.* 107 (1933), 627–647.

Mina Konaković, Keenan Crane, Bailin Deng, Sofien Bouaziz, Daniel Piker, and Mark Pauly. 2016. Beyond Developable: Computational Design and Fabrication with Auxetic Materials. *ACM Trans. Graph.* 35, 4 (2016), 89:1–11.

Mina Konaković-Luković, Julian Panetta, Keenan Crane, and Mark Pauly. 2018. Rapid Deployment of Curved Surfaces via Programmable Auxetics. *ACM Trans. Graph.* 37, 4 (2018), 106:1–13.

Julian Lienhard, Simon Schleicher, Simon Poppinga, Tom Masselter, Markuks Milwich, Thomas Speck, and Jan Knippers. 2011. Flectofin: a hingeless flapping mechanism inspired by nature. *Bioinspir. Biomim.* 6, Article 045001 (2011), 7 pages.

Kaj Madsen, Hans Bruun Nielsen, and Ole Tingleff. 2004. *Methods for non-linear least squares problems* (2nd ed.). Technical Univ. Denmark.

Luigi Malomo, Jesús Pérez, Emmanuel Iarussi, Nico Pietroni, Eder Miguel, Paolo Cignoni, and Bernd Bickel. 2018. FlexMaps: Computational Design of Flat Flexible Shells for Shaping 3D Objects. *ACM Trans. Graph.* 37, 6 (2018), 231:1–14.

Tom Masselter, Simon Poppinga, Julian Lienhard, Simon Schleicher, and Thomas Speck. 2012. The flower of Strelitzia reginae as concept generator for the development of a technical deformation system for architectural purposes. In *Proc. 7th. Plant Biomechanics Int. Conf.* INRIA, 389–392.

Toby Mitchell, Arek Mazurek, Christian Hartz, Masaaki Miki, and William Baker. 2018. Structural Applications of the Graphic Statics and Static-Kinematic Dualities: Rigid Origami, Self-Centering Cable Nets, and Linkage Meshes. In *Graphic statics.* Proc. IASS Symposia, Vol. 16.

Koryo Miura. 1980. Method of Packaging and Deployment of Large Membranes in Space. In *31st Congress Int'l Astronautical Federation (Tokyo 1980).*

Nicolas Montagne, Cyril Douthe, Xavier Tellier, Corentin Fivet, and Olivier Baverel. 2020. Voss surfaces: A design space for geodesic gridshells. *Journal of the IASS* 61, 4 (2020), 255–263.

Georg Nawratil. 2023. Generalizing continuous flexible Kokotsakis belts of the isogonal type. In *Proc. 20th Int. Conf. Geom. & Graphics*, L.-Y. Cheng (Ed.). Springer, 115–126.

Georg Nawratil. 2024. From axial C-hedra to general P-nets. In *Advances in Robot Kinematics 2024*. Springer. to appear. Extended version see arxiv:2401.04649.

Julian Panetta, Mina Konaković-Luković, Florin Isvoranu, Etienne Bouleau, and Mark Pauly. 2019. X-shells: A new class of deployable beam structures. *ACM Trans. Graph.* 38, 4 (2019), 83:1–15.

Stefan Pillwein, Johanna Kübert, Florian Rist, and Przemyslaw Musialski. 2020a. Design and Fabrication of Elastic Geodesic Grid Structures. In *Proc. SCF 2020: ACM Symposium on Computational Fabrication.* Article 2, 11 pages.

Stefan Pillwein, Kurt Leimer, Michael Birsak, and Przemyslaw Musialski. 2020b. On Elastic Geodesic Grids and Their Planar to Spatial Deployment. *ACM Trans. Graph.* 39, 4 (2020), 125:1–12.

Idzhad Kh. Sabitov. 1992. Local Theory of Bendings of Surfaces. In *Geometry III.* Springer, 179–256.

Robert Sauer. 1970. *Differenzengeometrie.* Springer.

Wolfgang Schief, Alexander Bobenko, and Tim Hoffmann. 2008. On the integrability of infinitesimal and finite deformations of polyhedral surfaces. In *Discrete differential geometry*, A. Bobenko et al. (Eds.). Oberwolfach Seminars, Vol. 38. Springer, 67–93.

Kiumars Sharifmoghaddam, Georg Nawratil, Arvin Rasoulzadeh, and Jonas Tervooren. 2021. Using Flexible Trapezoidal Quad-Surfaces for Transformable Design. In *Transformable Structures.* Proc. IASS Symposia, Vol. 28. 7:1–13.

Keyao Song, Xiang Zhou, Shixi Zang, Hai Wang, and Zhong You. 2017. Design of rigid-foldable doubly curved origami tessellations based on trapezoidal crease patterns. *Proc. R. Soc. A* 473, Article 20170016 (2017), 18 pages.

Enrique Soriano, Ramon Sastre, and Dionis Boixader. 2019. G-shells: Flat collapsible geodesic mechanisms for gridshells. In *Elastic Gridshells and Bending Active Systems.* Proc. IASS Annual Symposia, Vol. 11. 6:1–8.

Hellmuth Stachel. 2010. A kinematic approach to Kokotsakis meshes. *Comp. Aided Geom. Design* 27 (2010), 428–437.

Hellmuth Stachel. 2011. What lies betwen rigidity and flexibility of structures. *Serbian Architectural J.* 3 (2011), 102–115.

Tomohiro Tachi. 2009. Generalization of rigid foldable quadrilateral mesh origami. *J. Int. Ass. Shell & Spatial Structures* 50 (2009), 173–179.

Tomohiro Tachi. 2010a. Freeform rigid-foldable structure using bidirectionally flat-foldable planar quadrilateral mesh. In *Advances in Architectural Geometry 2010*, C. Ceccato et al. (Eds.). Springer, 87–102.

Tomohiro Tachi. 2010b. Geometric Considerations for the Design of Rigid Origami Structures. In *Proc. IASS Symposium 2010.* China Architecture & Building Press, 771–782.

Tomohiro Tachi. 2013. Composite Rigid-Foldable Curved Origami Structure. In *Proc. 1st Transformables Conf.*, F. Escrig and J. Sanchez (Eds.). Starbooks, Sevilla, 6 pp.

Tomohiro Tachi and Gregory Epps. 2011. Designing One-DOF Mechanisms for Architecture by Rationalizing Curved Folding. In *Proc. ALGODE Symposium*, Y. Ikeda (Ed.). Arch. Institute Japan, Tokyo, 14 pp.

Chengcheng Tang, Xiang Sun, Alexandra Gomes, Johannes Wallner, and Helmut Pottmann. 2014. Form-finding with Polyhedral Meshes Made Simple. *ACM Trans. Graph.* 33, 4 (2014), 70:1–9.

Sivan Toledo. 2003. TAUCS, A Library of Sparse Linear Solvers. Tel Aviv University. www.tau.ac.il/~stoledo/taucs

Aurel Voss. 1888. Über diejenigen Flächen, auf denen zwei Scharen geodätischer Linien ein conjugirtes System bilden. *Sitzungsber. Bayer. Akad. Wiss., math.-naturw. Klasse* 18 (1888), 95–102.

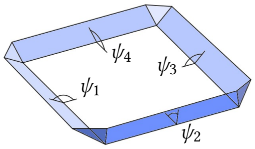
Walter Wunderlich. 1951. Zur Differenzengeometrie der Flächen konstanter negativer Krümmung. *Sitzungsber. Österr. Ak. Wiss. II* 160 (1951), 39–77.

Kai Xiao, Zihe Liang, Bihui Zou, Xiang Zhou, and Jaehyung Ju. 2022. Inverse design of 3D reconfigurable curvilinear modular origami structures using geometric and topological reconstructions. *Nature Commun.* 13, Article 7474 (2022), 9 pages.

Zirui Zhai, Lingling Wu, and Hanqing Jiang. 2021. Mechanical metamaterials based on origami and kirigami. *Appl. Phys. Rev.* 8, Article 04131 (2021), 22 pages.

A APPENDIX: ALGEBRAIC POSITION THEORY

Our aim is to establish a degree bound on the algebraic problem of flexibility of quad meshes with regular combinatorics. The key to a bound that is independent of the mesh size is Proposition 2.2, stating the mesh is flexible if and only all 3 by 3 submeshes are.



The case of planar faces. Consider a 3×3 quad mesh with planar faces consisting of the 1-ring neighbourhood of a central face. The four edges of this face have dihedral angles ψ_j. We introduce variables $w = \tan(\psi_1/2)$, $z = \tan(\psi_2/2)$, $v = \tan(\psi_3/2)$, $u = \tan(\psi_4/2)$. The configuration space of 3×3 meshes is now defined by 1 constraint per vertex which involves the two dihedral angles adjacent to this vertex [Izmestiev 2017, Lemma 4.3, p. 748]:

$$\begin{aligned} T_1(z,w) &:= A_{22}z^2w^2 + A_{20}z^2 + A_{02}w^2 + 2zw + A_{00} = 0,\\ T_2(v,z) &:= B_{22}v^2z^2 + B_{20}v^2 + B_{02}z^2 + 2vz + B_{00} = 0,\\ T_3(u,v) &:= C_{22}u^2v^2 + C_{20}u^2 + C_{02}v^2 + 2uv + C_{00} = 0,\\ T_4(w,u) &:= D_{22}w^2u^2 + D_{20}w^2 + D_{02}u^2 + 2uw + D_{00} = 0. \end{aligned} \tag{28}$$

The coefficients $A_{ij}, \ldots$ depend on the shape of the faces involved. The coefficients of the mixed terms are nonzero, so without loss of generality they equal 2. Generically, leading coefficients are nonzero:

$$A_{22} \neq 0,\ B_{22} \neq 0,\ C_{22} \neq 0,\ D_{22} \neq 0. \tag{29}$$

To eliminate the variable z from the equations $T_1 = T_2 = 0$, we use a standard tool: Consider T_1, T_2 as polynomials in the indeterminate z and observe that by (29), leading coefficients do not vanish:

$$A_{22}w^2 + A_{20} \neq 0, \quad B_{22}v^2 + B_{02} \neq 0. \tag{30}$$

We then compute the resultant of T_1, T_2. It has the form

$$F(w,v) = C_4(v)w^4 + \cdots + C_1(v)w + C_0(v) = 0 \tag{31}$$

where $C_4(v) = (A_{02}^2B_{22}^2 - 2A_{02}A_{22}B_{02}B_{22} + A_{22}^2B_{02}^2)v^4 + (2A_{02}^2B_{20}B_{22} - 2A_{02}A_{22}B_{00}B_{22} - 2A_{02}A_{22}B_{02}B_{20} + 2A_{22}^2B_{00}B_{02} + 4A_{02}A_{22})v^2 + (A_{02}^2B_{20}^2 - 2A_{02}A_{22}B_{00}B_{20} + A_{22}^2B_{00}^2)$.

Lemma A.1. *The coefficient $C_4(v)$ does not vanish if*

$$B_{20} \neq 0, \quad B_{00} \neq 0, \quad A_{02} \neq 0. \tag{32}$$

Proof. We employ methods of Symbolic Computation. The equations "$C_4 = 0$" are polynomial in the indeterminates A_{ij}, B_{ij}. We compute a Thomas decomposition of this system [Bächler et al. 2012] and arrive at $A_{02} = B_{00} = B_{20} = 0$. By (32) there is no solution. □

In view of this lemma, we make another genericity assumption, namely (32). It is our aim to show that a quad mesh is flexible if it has a sufficient number of individual positions which are all isometric (in the sense of rigid faces). Having computed a dense sequence of isometric meshes, we want to be sure that there is actually a continuous flexion connecting them. We therefore typically apply the result in a situation where the dihedral angles ψ_i are not arbitrary. It is thus reasonable to restrict to the following situation:

Definition A.2. *A set of N isometric configurations of a quad mesh is said to satisfy property (A) if all dihedral angles $\psi_1, \ldots, \psi_4$ assume N different values, and if each angle assumes values either in the interval $(0, 180°)$ or in the interval $(-180°, 0)$.*

We proceed in our task and eliminate the variable u from equations $T_3 = T_4 = 0$ by computing the resultant, analogous to above:

$$G(w,v) = D_4(v)w^4 + D_3(v)w^3 + D_2(v)w^2 + D_1(v)w + D_0(v).$$

We interpret both T_3 and T_4 as polynomials in u and observe that by (29), the leading coefficients do not vanish:

$$C_{22}v^2 + C_{20} \neq 0, \quad D_{22}w^2 + D_{02} \neq 0. \tag{33}$$

Existence of a one-parameter solution of the system $T_1 = T_2 = T_3 = T_4 = 0$ is now connected to the resultant of $F(w,v)$ and $G(w,v)$ with respect to the variable w. Our genericity assumptions ensure that the leading coefficients of F, G are nonzero,

$$C_4(v) \neq 0, \quad D_4(v) \neq 0. \tag{34}$$

cf. Lemma A.1 and the analogous statement for $D_4(v)$. Elimination of w leads to the resultant $H(v)$ which is a polynomial of degree at most 32 involving only even orders. Actually already the original system $T_1 = T_2 = T_3 = T_4 = 0$ is obviously symmetric w.r.t. to the substitution $(u,v,z,w) \mapsto (-u,-v,-z,-w)$. We conclude:

$$H \text{ has 17 pos. zeros (counting multiplicities)} \implies H = 0. \tag{35}$$

After these preparations, we are ready to state:

Proposition A.3. *Assume a quad mesh with planar faces of generic shape, such that* (29) *and* (32) *are fulfilled. Assume further that we have $N = 25$ mutually isometric positions of this mesh which fulfill property (A) as stated by Def. A.2. Then this mesh exhibits a continuous nontrivial isometric flexion.*

If we assume the given positions are generic samples, $N = 17$ is sufficient for the same conclusion.

Proof. By Prop. 2.2, it is sufficient to consider meshes of size 3×3. We show that there are continuous functions $w(v)$, $z(v)$, $u(v)$ such that for all v contained in a suitable interval, the original constraints $T_1 = \ldots = T_4 = 0$ are fulfilled, when substituting these functions for variables w, z, u. Since the dihedral angle v is moving in its interval, this represents a nontrivial continuous flexion. Observing degrees, we see that at most 8 of the assumed N solutions violate (30), (33), (34), so at least $N - 8$ lead to positive solutions of $H(v) = 0$. Since

$$N - 8 \geq 17, \tag{36}$$

Equ. (35) implies that $H = 0$. It follows that every (partial) solution v over the field of complex numbers lifts to solutions $v \longrightarrow (w,v)$ of $F_4(w,v) = G_4(w,v) = 0$, and further to a lift $(w,v) \longrightarrow (w,z,v,u)$ of solutions of $T_1 = \ldots T_4 = 0$. Generically however, (30), (33), (34) are not violated anyway, so (35) simply reduces to $N \geq 17$.

To discuss how many solutions are real, we use well-known properties of quadratic equations. Since the N assumed solutions

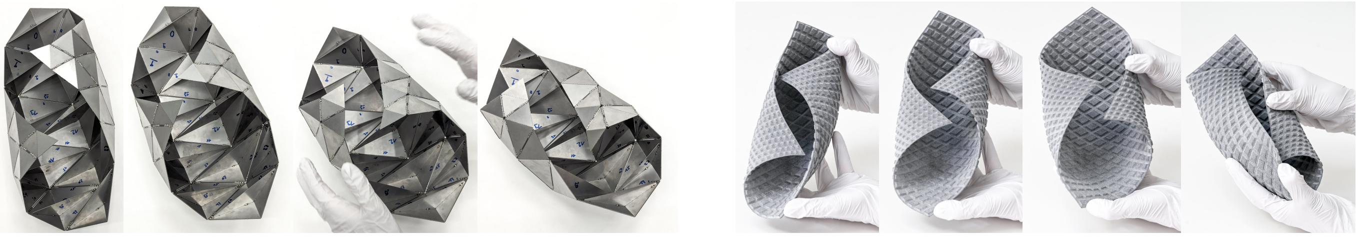


Fig. 23. Two flexing meshes (nonplanar quad case) approximating the same bending surface. Optimization here was based on the method of § 4.3.3. On the left we show the flexion of a 5 × 5 precision model fabricated from faces with hinges in the way described by Fig. 22. The model with 20 × 20 faces on the right has been 3D printed on a HP Multi Jet Fusion 3D Printing system using thermoplastic polyurethane. The material is flexible (Shore A hardness 88) but harder than the rubber of a typical automobile tire (Shore A hardness 70). The model has a thickness of 4 mm that is reduced to 0.6 mm at the joints. The resulting local reduction in stiffness creates a well-defined flexure bearing. Note that the 3D printed version still exhibits only the flexibility that was computed on the basis of rigid faces and hinges in the edges. This is shown by our actuating the model on the right hand side only and observing that the left hand side automatically moves as computed.

of $T_1 = \ldots = T_4 = 0$ are real, the discriminant of the equation $T_2(v, z) = 0$, seen as a quadratic equation in z, is nonnegative there:

$$\Delta_2(v) = 4v^2 - 4(B_{20}v^2 + B_{00})(B_{22}v^2 + B_{02}) \geq 0. \qquad (37)$$

At least $N-2$ of the N given values of v have $\Delta_2(v) > 0$, and around each there is a small interval $(v-\epsilon, v+\epsilon)$ where $\Delta_2(v) > 0$. It follows that there is a continuous real solution branch $z(v)$. The argument can be repeated for $u(v)$ and $w(v)$, using T_3 and T_4, respectively. □

Proposition A.4. *Further to Prop. A.3, if the given isometric positions of the mesh are infinitesimally flexible, $N = 17$ is sufficient to conclude existence of a continuous flexion. If they are even 2nd order infinitesimally flexible, $N = 14$ is sufficient.*

If we assume that the given positions are generic samples, $N = 9$ resp. $N = 6$ are sufficient.

Proof. Infinitesimal flexibility means that not only does (w,z,v,u) fulfill $T_i = 0$ for $i = 1, \ldots, 4$, but there is also a nonzero derivative vector $(\dot{w}, \dot{z}, \dot{v}, \dot{u})$ such that $\langle \nabla T_i, (\dot{w}, \dot{z}, \dot{v}, \dot{u})^T \rangle = 0$. It is known that the resultant F of T_1, T_2, see (31), can be expressed as $F(v, w) = a(w, z, v)T_1(z, w) + b(w, z, v)T_2(v, z)$. Observing that in the solution variety $T_1 = T_2 = 0$, we compute partial derivatives $F_w = aT_{1w}$, $F_v = bT_{2v}$ and $F_z = aT_{1z} + bT_{2z}$. Observe also that $F_z = 0$ because the variable z does not occur in F. Some elementary manipulations now show $\langle \nabla F, (\dot{w}, \dot{v})^T \rangle = F_w\dot{w} + F_v\dot{v} = 0$. Similarly, $\langle \nabla G, (\dot{w}, \dot{v})^T \rangle = 0$. Geometrically, this means that curves $F(w, v) = 0$ and $G(w, v) = 0$ have the same tangent and therefore intersection multiplicity ≥ 2.

Their resultant $H(v)$ then has a double zero, replacing Equ. (36) by $2(N - 8) \geq 17$. In case of 2nd order flexibility, an analogous argument yields $3(N - 8) \geq 17$. Generically, the same argument as above yields $2N \geq 17$ resp. $3N \geq 17$. □

The skew quads case. For each face consider the tetrahedron which is the convex hull of its four vertices. In a 3 × 3 piece of quad mesh, we can still introduce angles $\psi_1, \ldots \psi_4$ as above, but these angles are now measured between faces of tetrahedra. The configuration space is defined by equations very similar to (28), see [Aikyn et al. 2024]: $T_1(z, w)$ is replaced by $\tilde{T}_1(z, w) := T_1(z, w) + A_{21}z^2w + A_{12}w^2z + A_{10}z + A_{01}w$ and similar for T_2, T_3, T_4. We get:

Proposition A.5. *A generic quad mesh, which has $N = 49$ mutually isometric positions enjoying property (A) as stated by Def. A.2, does also exhibit a continuous nontrivial flexion. If the given positions are generic samples, then $N = 33$ is sufficient.*

Proof. The counting arguments we used in the planar faces case still work, leading to a resultant H of degree 32. However, equations are not symmetric, and (35) is replaced by the statement hat more than 32 zeros of H imply that $H = 0$. Equation (36) is replaced by $N - 16 \geq 33$, leading to the number $N = 49$. In the generic case we get the equation $N \geq 33$. We omit the details. □

Propositions A.3 to A.5 together yield Th. 3.2. The proofs above assumed generic shape of the mesh. If these assumptions are not fulfilled, then the resultants will actually have a lower degree, and a smaller number of mutually isometric mesh positions already implies a continuous flexion.

B APPENDIX: A SAMPLE TYPE OF FLEXIBLE 3X3 MESH

We here summarize one of the types of flexible meshes with planar quadrilaterals that have been found by Izmestiev [2017], namely the so-called linear conjugate type, anti-deltoid subtype. The 16 angles around the vertices of the inner face are labeled a_i, b_i, c_i, d_i, $i = 1, 2, 3, 4$ according to Fig. 15. With the notation

$$\lambda_i := \frac{\tan d_i + \tan a_i}{\tan d_i - \tan a_i}, \ \mu_i := \frac{\tan d_i + \tan c_i}{\tan d_i - \tan c_i},$$
$$\kappa_i \in \left\{ \frac{\sin \frac{1}{2}(a_i - b_i)}{\sin \frac{1}{2}(a_i + b_i)}, \frac{\cos \frac{1}{2}(a_i - b_i)}{\cos \frac{1}{2}(a_i + b_i)} \right\}.$$

introduced by Def. 2.8 and Th. 2.4 of [Izmestiev 2017], the variables $a_i, \ldots, d_i$ obey $d_1 + d_2 + d_3 + d_4 = 2\pi$ (expressing planarity of the central face), further the linear constraints $a_1 + d_1 = b_1 + c_1 = a_3 + b_3 = c_3 + d_3 = a_2 + c_2 = b_2 + c_2 = a_4 + c_4 = b_4 + c_4 = \pi$, and the nonlinear constraints $\kappa_4 \frac{\mu_1 - 1}{\cos \delta_1} = \kappa_2 \frac{\lambda_3 - 1}{\cos \delta_3}$, $\kappa_4^2 \mu_1 = \lambda_3$. The linear constraints express the fact that opposite vertices No. 2 and 4 are of the anti-deltoid type and vertices No. 1 and 3 are of the anti-isogram type [Izmestiev 2017, § 3.6].