Quadrilateral Mesh Generation for Open Surfaces with Negative Euler Characteristics Based on Symmetric Abel Differentials

Submission ID 1011

Abstract

This study introduces a quadrilateral mesh generation method tailored for surfaces with multiple boundaries and negative Euler characteristic numbers, utilizing symmetric Abel differentials. Grounded in robust theoretical principles, this method generates high-quality quad-meshes characterized by exceptional orthogonality and minimal singularity presence. Additionally, the grid lines are strategically aligned with the boundaries, enhancing the mesh's structural integrity and applicability in complex modeling scenarios.

1 Introduction

1.1 Motivation A quadrilateral mesh on a surface refers to a type of polygonal mesh where the surface is discretized into a network of quadrilaterals (four-sided polygons). Each element of the mesh, or "quad," approximates a small portion of the surface, and the collection of all quads provides a comprehensive covering of the entire surface. In a well-constructed quadrilateral mesh, the vertices, edges, and faces are arranged in a manner that respects the geometric and topological properties of the surface.

Quadrilateral elements can more naturally align with directional features of a surface, such as curvature and anisotropy, which is advantageous in complex geometries seen in engineering and biomedical contexts. Quadrilateral meshes are conducive to subdivision schemes used in graphics and geometric modeling [6]. These schemes, which iteratively refine and smooth the mesh, often work better with quads, allowing for the generation of smooth surfaces from coarse base meshes. In computational engineering, such as finite element analysis (FEA) [48], quadrilateral meshes are favored because they tend to have favorable numerical properties. Quads can provide more uniform element quality and better convergence properties compared to triangular meshes in certain simulation contexts.

In general, a good quality quad-mesh should satisfy the following criteria:

- orthogonality: the adjacent edges of each quad-face should be orthogonal to each other;
- boundary alignment: the quad-faces should be aligned with the boundaries, namely the grid lines are either parallel or orthogonal to the boundary curves;
- uniformity: the sizes of the quads should be locally uniform, and globally varies smoothly;
- singularity: the number of singularities should be small and their positions should be controllable;

Because quad-mesh generation plays a fundamental role in CAE fields, researchers have developed many algorithms [34, 3]. However, for surfaces with negative Euler characteristic numbers and multiple boundaries, high-quality quad-mesh generation remains challenging to meet the above requirements.

In this work, we focus on tackling the central problem:

Generating quad-meshes for surfaces with negative Euler characteristic numbers and multiple boundaries that satisfy the criteria of orthogonality, boundary alignment, uniformity, and singularity.

Our method is based on Abel differential theories on Riemann surfaces [9, 30]

1.2 Abel Differential Recently, Lei et al. [7, 24, 47] introduced a novel theoretic framework for surface quad-mesh generation, which bridges the quad-meshes with meromorphic quartic differentials on the Riemann surface, namely a meromorphic section of a special holomorphic line bundle, such that the singularities of the quad-mesh are governed by the Abel-Jacobi theorem. The method introduced in [46] uses a special type of meromorphic quartic differential, the holomorphic differential, to generate high-quality quad meshes on a polyannulus as uniform as possible. But the method can not handle the surface with negative Euler characteristic numbers directly.

This work generalizes the works in [7] and [46] to generate quad-meshes on the surface with multiple boundaries and negative Euler characteristic numbers using Abel differential. A holomorphic differential can only have zero singularities, but an Abel differential may have zeros and poles. Therefore Abel differentials are more flexible, and capable of modeling boundary holes

(or cusps) as poles to ensure the boundary alignment property.

1.3 Contributions This work proposes a novel method for generating quadrilateral meshes on the surfaces with boundaries and negative Euler characteristic numbers, which have merits of high orthogonality, boundary alignment, minimal number of singularities and good uniformity. In detail:

- Propose a framework to compute the first and third types Abel differentials on Riemann surfaces with boundaries;
- Propose a practical algorithm to generate quad-meshes on surfaces with negative Euler characteristic numbers and multiple boundaries.

The proposed work is a natural extension of the method introduced by Gu and Yau [11], which generalized the Finite Element Method to exterior calculus and computed holomorphic 1-forms on closed surfaces based on the Hodge decomposition. This approach was later extended to surfaces with boundaries using the double-covering technique in [42, 12].

Similar to these previous works, the proposed method relies on Hodge decomposition and the double-covering technique. However, the key innovation lies in addressing the symmetry of the holomorphic differential basis. In [42, 12], the holomorphic 1-form basis is computed on the double-covered surface without enforcing symmetry. As a result, the holomorphic 1-forms derived from linear combinations of this basis often lack symmetry. When these asymmetric 1-forms are projected back onto the original input mesh, the boundaries in the parameter domain fail to align with horizontal or vertical directions. While this limitation is acceptable for general texture mapping, as demonstrated in [42], it poses a significant challenge for quad-mesh generation.

To overcome this issue, the current work employs the symmetrization method detailed in subsection 4.5 to ensure the holomorphic 1-form basis is symmetric. This adjustment guarantees proper boundary alignment, which is critical for generating high-quality quad meshes.

2 Previous Work

Quadrilateral mesh generation is a cornerstone technique in science and engineering, prized for its tensor-product structure and its ability to approximate smooth surfaces accurately. The topic is well-documented across extensive literature. For detailed literature reviews, we suggest consulting references such as [4, 34, 3]. Below, we focus on discussing the principal methods of quadrilateral mesh generation.

Converting Triangulation Approach One prevalent technique involves transforming a triangular mesh into a quadrilateral mesh. This transformation typically encompasses steps such as edge matching, vertex insertion, and optimization to refine the properties of the quad mesh. The most straightforward approach combines two adjacent triangles to form a single quadrilateral, thereby creating a quad mesh [14, 35, 38, 40]. However, this method generally yields unstructured quad meshes and is fraught with challenges such as increased complexity, potential degradation in mesh quality, loss of geometric detail, boundary issues, and irregular shapes.

Paving Method Approach The paving method, based on the Advancing Front Technique (AFT) [28], is a key algorithm for quadrilateral mesh generation, known for its adaptability to complex boundaries and high-quality meshes. Starting from boundaries, it progresses inward layer by layer. A direct advancing front method [1] enhances edge alignment and reduces irregular nodes. However, the approach lacks robust theoretical guarantees and is computationally intensive. Improvements include single-element insertion with collision detection for robustness [43] and element size control for smooth transitions [45]. Region decomposition into subdomains [29] enables automatic meshing of complex geometries with constraints, improving efficiency and adaptability.

Patch-based Approach An alternative strategy is the patch-based method, which segments data or regions into smaller patches for streamlined analysis and processing. This approach is frequently adopted in fields such as image processing, computer graphics, and machine learning to efficiently handle complex datasets. In this method, adjacent triangular faces are merged into patches using clustering techniques that may include normal-based and center-based methods [2, 5]. The generation of these patches is often supported by the implementation of poly-cube maps [44, 41, 27, 15].

Parameterization Based Approach Another widely used technique in quad-meshing employs parameterization-based algorithms. This approach involves creating parameterizations or mappings of geometric data, which are crucial for various applications in computer graphics, computer-aided design, and similar areas. Among these, the spectral surface quadrangulation method [10, 16] is implemented on the input mesh. Various techniques such as global conformal parameterization [13], discrete harmonic forms [39], periodic global parameterization [32], branched covering methods [19], and discrete surface Ricci flow [18, 36, 37] all depend on parameterization to effectively generate quad meshes.

Frame Field Approach Among the prominent methods, cross-field guided quad-mesh generation stands out for its application in structuring quad meshes on complex surfaces. This technique utilizes cross fields to dictate the orientation of quads, enhancing the structure necessary for applications like finite element analysis and 3D modeling. Common representations for crosses include N-RoSy [31, 21], period jump [26], and complex value [20]. These methods typically employ energy minimization to achieve smooth cross fields, often measured by discrete Dirichlet energy [17], and generate quad meshes using techniques such as streamline tracing [33] or parameterization [3]. While offering structured meshing and precise alignment, cross-field methods can be computationally intensive and sensitive to algorithms, sometimes requiring manual adjustments. Lei et al. [25] established essential conditions for cross field existence and differentiated cross fields from quad meshes using fiber bundle theory.

Method Based On Abel-Jacobi Approach Chen et al. [7] established the necessary and sufficient conditions for a Riemannian metric induced by a quad-mesh, including the Gauss-Bonnet condition for curvatures, holonomy, boundary alignment, and finite streamline conditions. Lei et al. [24] demonstrated that the holonomy condition is expressible through the Abel-Jacobi equation in algebraic geometry. Subsequently, Zheng et al. [47] developed a practical algorithm to optimize singularity configurations to meet the Abel-Jacobi condition. Their findings suggest that surface quad-meshes correspond to meromorphic quartic differentials and act as meromorphic global sections of a special holomorphic line bundle on Riemann surfaces, with mesh singularities forming the characteristic class of the line bundle. Thus, singularity indices are dictated by the Abel-Jacobi equations, providing a theoretical foundation for structured mesh generation. Additionally, Lei et al. [22, 23] extended these principles to hexahedral mesh generation through surface foliations, a subset of meromorphic quartic differentials.

Our current work is mainly based on this framework, a special type of meromorphic quartic differential is constructed, the symmetric Abel differential, termed for surfaces with boundaries and negative Euler characteristic numbers.

3 Theoretic Background

This section briefly introduces the theoretic foundation for the currently. We refer readers to [9, 30] for thorough treatments.

3.1 Abel Differentials on Riemann Surfaces

Suppose Ω is domain on the complex plane $\Omega \subset \mathbb{C}$, a complex function is $f : \Omega \to \mathbb{C}$ is *holomorphic* , if it satisfies Cauchy-Riemann equation,

$$\partial_{\bar{z}} f(z) = 0, \quad \forall z \in \Omega,$$

where the differential operator $\partial_{\bar{z}} = 1/2(\partial_x + \sqrt{-1}\partial_y)$. f is *biholomorphic*, if it is invertible and f^{-1} is also holomorphic.

A surface S is a two dimensional *manifold* with an open covering $S \subset \bigcup_\alpha U_\alpha$, each open set U_α is associated with a homeomorphism $\varphi_\alpha : U_\alpha \to \mathbb{C}$, $(U_\alpha, \varphi_\alpha)$ forms a *local chart*. For the intersection between two local charts $(U_\alpha, \varphi_\alpha)$, (U_β, φ_β) the *transition function* is defined as

$$\varphi_{\alpha\beta} : \varphi_\alpha(U_\alpha \cap U_\beta) \to \varphi_\beta(U_\alpha \cap U_\beta), \varphi_{\alpha\beta} := \varphi_\beta \circ \varphi_\alpha^{-1}.$$

If all the transition functions are biholomorphic, then S is called a *Riemann surface* and the collection of local charts $\{(U_\alpha, \varphi_\alpha)\}$ is called a *conformal atlas* or a *conformal structure*.

Suppose S is an oriented surface with a Riemannian metric $\mathbf{g}$. For each point $p \in S$, there is a neighborhood U_α, on which we can define the so-called *isothermal coordinates* (x_α, y_α), such that the metric has a special form

$$\mathbf{g} = e^{2\lambda(x_\alpha, y_\alpha)}(dx_\alpha^2 + dy_\alpha^2),$$

where the scalar function $\lambda_\alpha : U_\alpha \to \mathbb{R}$ is called the *conformal factor* function. The union of all isothermal coordinates charts form the conformal atlas of the surface, therefore the oriented metric surface $(S, \mathbf{g})$ is a Riemann surface.

A complex function defined on a Riemann surface is called a *holomorphic* or *meromorphic* function, if on each local chart (U_α, z_α), the local representation of f is $f_\alpha(z_\alpha)$, which is holomorphic or meromorphic. Furthermore, in the intersection of two charts, $f_\alpha(z_\alpha) = f_\beta(z_\beta)$. Namely, the function is globally consistently defined. A meromorphic function defined on the complex plane is called *symmetric*, if it satisfies

$$f(z) = \overline{f(\bar{z})}.$$

Suppose $p \in S$ is a point on the Riemann surface S, choose a local coordinates z of the neighborhood of p, such that $z_\alpha(p) = 0$, if in the neighborhood f_α is

$$f_\alpha(z_\alpha) = z_\alpha^\nu h_\alpha(z_\alpha),$$

where h_α is a holomorphic function, $h_\alpha(0) \neq 0$, $\nu \in \mathbb{Z}$. ν is called the *order* of f at p, denoted as $\nu_p(f)$. When $\nu_p(f) > 0$, p is called a *zero* of f, and $\nu_p(f)$ is called the *order* of the zero; when $\nu_p(f) < 0$, p is called a *pole* of f, $|\nu_p(f)|$ is called the *order* of the pole p.

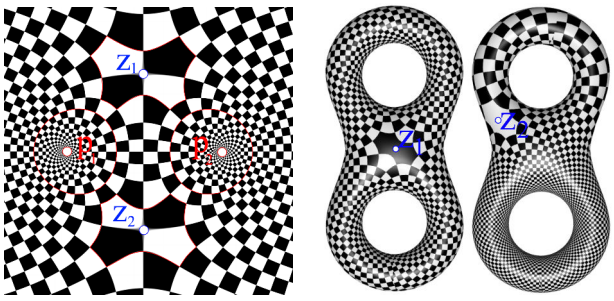


Figure 1: Abel differential (left), holomorphic differential (right) with poles (red circles) and zeros (blue circles).

A complex complex differential 1-form ω is called a *Abel differential*, if on each local chart z_α, its local representation is

$$\omega = f_\alpha(z_\alpha)dz_\alpha,$$

where f_α is a meromorphic function, and on the other chart $\omega = f_\beta(z_\beta)dz_\beta$, where

$$f_\alpha(z_\alpha) = f_\beta(z_\beta(z_\alpha))\frac{dz_\beta}{dz_\alpha}.$$

An Abel differential has local *Laurent series*:

$$\omega = \left(\underbrace{\frac{a_{-n}}{z^n} + \cdots + \frac{a_{-2}}{z^2} + \frac{a_{-1}}{z}}_{principle\ part} + \underbrace{a_0 + \cdots a_k z^k \cdots}_{holomorphic\ part} \right) dz$$
$$= \omega_2 + \omega_3 + \omega_1$$

where

$$Type\ 1: \quad \omega_1 = (a_0 + a_1 + \cdots + a_k z^k + \cdots)dz$$
$$Type\ 2: \quad \omega_2 = \left(\frac{a_{-n}}{z^n} + \frac{a_{-(n-1)}}{z^{n-1}} + \cdots + \frac{a_{-2}}{z^2} \right) dz$$
$$Type\ 3: \quad \omega_3 = \left(\frac{a_{-1}}{z-p} - \frac{a_{-1}}{z-q} \right) dz$$

Fig. 1 left shows an Abel differential with two zeros (blue circles centered at $\pm\sqrt{-1}$) and two simple poles (red circles centered at ± 1), whose local representation is

$$\omega = \left(1 + \frac{1}{z-1} - \frac{1}{z+1} \right) dz,$$

The right frame shows a holomorphic differential (type 1 Abel differential) on a genus two surface with two zero points (blue circles).

3.2 Hodge Decomposition A *differential 1-form* ω on a differential manifold M at each point p assigns a linear functional ω_p on the tangent space T_pM at p. If $(x^1, x^2, \ldots, x^n)$ are local coordinates on M, then ω can be written as

$$\omega = \sum_{i=1}^{n} f_i dx^i$$

where f_i are smooth functions and dx^i are the basis for cotangent space T_p^*M. The *wedge product* of k differential 1-forms $\omega_1, \omega_2, \ldots, \omega_k$ is a anti-symmetric multi-linear function, $\omega_1 \wedge \cdots \wedge \omega_k : T_pM \times \cdots \times T_pM \to \mathbb{R}$, $\forall v_1, \ldots, v_k \in T_pM$,

$$\omega_1 \wedge \cdots \wedge \omega_k(v_1, \ldots, v_k) := \begin{vmatrix} \omega_1(v_1) & \cdots & \omega_1(v_k) \\ \omega_2(v_1) & \cdots & \omega_2(v_k) \\ \vdots & & \vdots \\ \omega_k(v_1) & \cdots & \omega_k(v_k) \end{vmatrix},$$

which is called a k-form, and has the local representation

$$\omega = \sum_{i_1 < \cdots < i_k} f_{i_1 \ldots i_k} dx^{i_1} \wedge \cdots \wedge dx^{i_k}.$$

The linear space of all k-forms is denoted as $\Omega^k(M)$. The *exterior derivative* $d : \Omega^k(M) \to \Omega^{k+1}(M)$ acts in a k-form given by

$$d\omega = \sum_{i_1 < \cdots < i_k} df_{i_1 \ldots i_k} \wedge dx^{i_1} \wedge \cdots \wedge dx^{i_k},$$

The kernel of d is called *closed forms*, the image of d the *exact forms*. The exact forms must be closed, that is, $d^2 = 0$, the difference is the de Rham cohomology group,

$$H^k_{dR}(S, \mathbb{R}) := \frac{\text{Ker } d^k}{\text{Img } d^{k-1}}.$$

Suppose M has a Riemannian metric $\mathbf{g} = (g_{ij})$. Let $\omega = \sum_i \omega_i dx^i$ and $\eta = \sum_j \eta_j dx^j$, the *inner product* is given by

$$g(\omega, \eta) = \sum_{i,j} \omega_i \eta_j g^{ij}$$

where g^{ij} are the components of the inverse metric tensor. For an orthonormal basis $\{e^1, e^2, \ldots, e^n\}$ of the cotangent space at a point, the action of the *Hodge star* on a basis k-form is given by

$$\star(e^{i_1} \wedge e^{i_2} \wedge \cdots \wedge e^{i_k}) = \pm e^{j_1} \wedge e^{j_2} \wedge \cdots \wedge e^{j_{n-k}}$$

where the sign is positive, If $(i_1, i_2, \ldots, i_k, j_1, \ldots, j_{n-k})$ is an even permutation of $(1, 2, \ldots, n)$; negative otherwise. If ω is a k-form, the *codifferential* $\delta\omega$ is given by:

$$\delta\omega = (-1)^{n(k+1)+1} \star d \star \omega,$$

The *Laplace-Beltrami* operator $\Delta : \Omega^k(M) \to \Omega^k(M)$ is defined as $\Delta := d\delta + \delta d$. A differential form ω is called

harmonic, if it satisfies the Laplace-de Rham equation: $\Delta\omega = 0$. The Hodge decomposition theorem plays a fundamental role in the current work.

THEOREM 3.1. (HODGE DECOMPOSITION) *In a compact orientable Riemannian manifold* $(M, \mathbf{g})$*, every de Rham cohomology class has a unique harmonic representative.*

4 Computational Algorithm

In this section, we explain our algorithmic pipeline in detail.

4.1 Doubling the Input Surfaces The input is a compact and oriented surface embedded in $\mathbb{E}^3$ with negative Euler characteristic numbers and multiple boundaries. Fig. 2 shows the surface of the kitten, which is a surface of genus one with two boundaries. The surface is with a triangulation $\mathcal{T}$ and approximated by the polyhedral metric, such that each face is a Euclidean triangle.

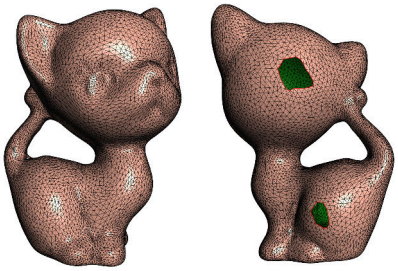


Figure 2: The input genus one surface with two boundaries.

From the input surface S, we construct a closed symmetric surface $\bar{S}$ using the *doubling* technique. Basically, we denote the original surface as S^+, duplicate a copy S^- of S^+, and reverse the orientations of all faces in S^{-1}, which induces a bijective mapping $\psi : S^+ \to S^-$; then we glue both S^+ and S^- along the common boundaries to obtain the doubled surface $\bar{S}$. The mapping ψ is lifted as a convolution map $\varphi : \bar{S} \to \bar{S}$ and $\varphi^2 = \mathrm{id}$. Alg 4.1 gives the details of the algorithm, where the dual map is denoted as ψ, the convolution map as φ, and the covering vertex of v as $\bar{v}$.

ALGORITHM 4.1. (DOUBLING) Surface Doubling

Require: Input mesh S^+ with boundaries
Ensure: The doubled surface $\bar{S}$ and the convolution map $\varphi : \bar{S} \to \bar{S}$
function DOUBLING(S)
for each vertex $v_i^+ \in S^+$ **do**
construct the dual vertex v_i^- in S^-
Set the images of the dual map $\psi(v_i^+)$ as v_i^-
end for
for each face $[v_i^+, v_j^+, v_k^+] \in S^+$ **do**
construct the dual face $[v_j^-, v_i^-, v_k^-] \in S^-$
end for
for each vertex $v_i^+ \in S^+$ **do**
construct a covering vertex $\bar{v}_i^+$ in $\bar{S}$
end for
for each vertex $v_i^- \in S^-$ and $v_i^- \notin \partial S^-$ **do**
construct a covering vertex $\bar{v}_i^-$ in $\bar{S}$
end for
for each boundary vertex $v_i^- \in \partial S^-$ **do**
set the covering vertex $\bar{v}_i^-$ as $\overline{\psi^{-1}(v_i^-)}$ in $\bar{S}$
end for
for each face $[v_i^+, v_j^+, v_k^+] \in S^+$ **do**
construct the covering face $[\bar{v}_j^+, \bar{v}_i^+, \bar{v}_k^+] \in \bar{S}$
end for
for each face $[v_i^-, v_j^-, v_k^-] \in S^-$ **do**
construct the covering face $[\bar{v}_j^-, \bar{v}_i^-, \bar{v}_k^-] \in \bar{S}$
end for
for each vertex $v_i^+ \in S^+$ **do**
set $\varphi(\bar{v}_i^+)$ as $\overline{\psi(v_i^+)}$.
end for
end function

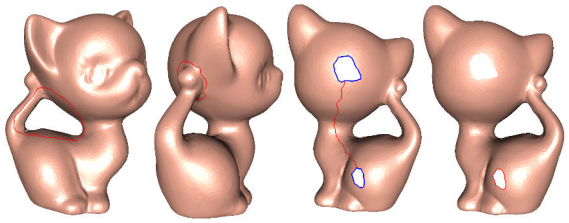


Figure 3: Homology group basis, from left to right: tunnel loop, handle loop, bridge loop and boundary loop.

4.2 Homology Group Basis Suppose that the surface S is of genus g with b boundaries. First, we fill the b bouondaries with b disks to obtain a closed surface $\hat{S}$ embedded in $\mathbb{E}^3$. By adding the $\{\infty\}$ point, $\mathbb{E}^3$ is compactified to $\mathbb{S}^3$, and the surface $\hat{S}$ is embedded in $\mathbb{S}^3$. $\hat{S}$ divides $\mathbb{S}^3$ into two connected components, the interior $\mathbb{I}$ and the exterior $\mathbb{O}$. The exterior contains the infinite point $\infty \in \mathbb{O}$.

Suppose $\gamma : \mathbb{S}^1 \to \hat{S}$ is a closed loop on the surface represented as a continuous map from the unit circle to the surface, if γ is homological to 0 in the interior $\mathbb{I}$, but nontrivial in the exterior $\mathbb{O}$, then γ is called a *tunnel loop*; on the contrary, if the loop γ is trivial in the exterior and nontrivial in the interior, then γ is called a *handle loop*. Each handle h_i on the surface has a pair of handle loops a_i and tunnel loop b_i, $i = 1, 2, \ldots, g$, where g is the genus of $\hat{S}$. Fig. 3 the first and second frames show the tunnel and handle loops of the kitten

model. The handle and tunnel loops can be computed using persistent homology algorithm in [8].

Suppose that the genus g surface S has n boundaries, $\partial S = \{\alpha_0, \alpha_1, \ldots, \alpha_{n-1}\}$, we call α_k as the k-th *boundary loop*. For each boundary loop α_k, $k > 0$, the shortest path between α_0 and α_k is denoted as β_k, and called the k-th *bridge* path.

We denote the tunnel and handle loops on S^+ as $\{a_i^+, b_i^+\}_{i=1}^g$, their dual images on S^- are denoted as

$$a_i^- := \psi(a_i^+), \quad b_i^- := \psi(b_i^+), \quad i = 1, 2, \ldots, g,$$

Their liftings on the doubled mesh $\bar{S}$ are denoted as $\{\bar{a}_i^+, \bar{b}_i^+, \bar{a}_i^-, \bar{b}_i^-\}_{i=1}^g$. It is obvious that

$$\varphi(\bar{a}_i^{\pm}) = \bar{a}_i^{\mp}, \quad \varphi(\bar{b}_i^{\pm}) = \bar{b}_i^{\mp}.$$

The lifting of the boundary loop $\alpha_k^+ \subset \partial S^+$ are denoted as $\bar{\alpha}_k$, $k = 1, 2, \ldots, n-1$, which are invariant under the convolution $\varphi(\bar{\alpha}_k) = \bar{\alpha}_k$. Each bridge path $\beta_k^+ \subset S^+$ has a dual path $\beta_k^- \subset S^-$, the union of their liftings on $\bar{S}$ is a loop, the so-called *bridge loop*:

$$\bar{\beta}_k := \bar{\beta}_k^+ \cup \bar{\beta}_k^-, \quad k = 1, 2, \ldots, n-1.$$

The $4g$ tunnel and handle loops, $n-1$ boundary loops and $n-1$ bridge loops form the basis of the first homology group of the doubled surface $H_1(\bar{S}, \mathbb{Z})$, hence the genus of $\bar{S}$ is $2g+(n-1)$. Fig. 3 shows the homology group basis of the kitten model with 2 holes.

Suppose S is a closed genus g surface, with homology group basis $\{\gamma_1, \gamma_2, \ldots, \gamma_{2g}\}$, we slice the surface along these loops to obtain a surface with $2g$ boundaries, $\{b_1, b_2, \ldots, b_{2g}\}$. Without loss of generality, we can assume b_1 is the longest boundary. For each b_k, $k > 1$, we find the shortest path τ_k from b_k to b_1. Then we further slice the surface along these τ_k's. The resulting surface is called a *fundamental domain* of the surface. The details can be found in Alg. 4.2.

Algorithm 4.2. (Foundamental Domain)
Foundamental Domain

Require: Input closed mesh S with genus g
Require: Basis of $H_1(S, \mathbb{Z})$
Ensure: A fundamental domain of S.

```
function FoundamentalDomain(S)
    for each base loop γ_k of H_1(S, Z) do
        Slice the mesh S along the loop γ_k
    end for
    The surface has 2g boundaries, ∂S = ⋃_{k=1}^{2g} b_k
    for each boundary b_k, k > 1 do
        Compute the shortest path τ_k from b_1 to b_k
    end for
    for each shortest path τ_k, k > 1 do
        Slice the surface along τ_k
    end for
end function
```

4.3 Hodge Decomposition We construct the basis of the cohomology group of the doubled mesh $H^1(\bar{S}, \mathbb{Z})$. Basically, for each base loop γ of $H_1(\bar{S}, \mathbb{Z})$, we slice the surface along γ_k to produce a surface with boundaries $\tilde{S}_k$, such that $\partial \tilde{S}_k = \tilde{\gamma}_k^+ \cup \tilde{\gamma}_k^-$. We then construct a real valued function $\tilde{f}_k : \tilde{S}_k \to \mathbb{R}$, such that the restriction of $\tilde{f}_k$ on $\tilde{\gamma}_k^+$ and $\tilde{\gamma}_k^-$ are 1 and 0 respectively. The the differential of $\tilde{f}_k$, $d\tilde{f}_k$ is a closed differential form well defined on the original surface $\bar{S}$, denoted as $\bar{\eta}_k$, then all such $\bar{\eta}_k$'s form the basis of the $H^1(\bar{S}, \mathbb{R})$.

Algorithm 4.3. (Cohomology Basis)
Cohomology Basis

Require: Input closed mesh S with genus g
Require: Basis of $H_1(S, \mathbb{Z})$
Ensure: Basis of $H^1(S, \mathbb{R})$, $\{\eta_1, \eta_2, \ldots, \eta_{2g}\}$.

```
function CohomologyBasis(S)
    for each base loop γ_k of H_1(S, Z) do
        Slice the mesh S along the loop γ_k to
        obtain S̃_k, ∂S̃_k = γ̃_k^+ − γ̃_k^−
        // Construct the function f̃_k : S̃_k → R
        for each vertex v_i ∈ S̃_k do
            if v_i ∉ ∂S̃_k then
                f̃_k(v_i) ← random
            else if v_i ∈ γ̃_k^+ then
                f̃_k(v_i) ← 1
            else
                f̃_k(v_i) ← 0
            end if
        end for
        // Construct the differential η_k ← df̃_k
        for [v_i, v_j] ∈ S do
            if [v_i, v_j] ∈ γ_k then
                η([v_i, v_j]) ← 0
            else
                η([v_i, v_j]) ← f̃_k(ṽ_j) − f̃_k(ṽ_i)
            end if
        end for
```

According to the Hodge decomposition theorem, each cohomological class has a unique harmonic differential from. Namely, all harmonic differentials form a group $H^1_\Delta(\bar{S}, \mathbb{R})$, which is isomorphic to $H^1(\bar{S}, \mathbb{R})$. For each $\bar{\eta} \in H^1(\bar{S}, \mathbb{R})$, we find a function $\bar{f} : \bar{S} \to \mathbb{R}$, such that $\delta(\bar{\eta} + d\bar{f}) = 0$, this is reduced to a Poisson equation

$$\Delta \bar{f} = -\delta \bar{\eta}. \tag{4.1}$$

The Poisson equation can be solved using the Finite Element Method on exterior calculus as follows. For each edge $e_{ij} = [v_i, v_j]$ on the mesh, it is shared by two faces $f_{ijk} = [v_i, v_j, v_k]$ and $f_{jik} = [v_j, v_i, v_l]$, the corner

angle in f_{ijk} against e_{ij} is θ_k^{ij}, and that in f_{jil} θ_l^{ji}. The *cotangent edge weight* of e_{ij} is defined as

$$w_{ij} := \frac{1}{2}\left(\cot\theta_k^{ij} + \cot\theta_l^{ji}\right). \tag{4.2}$$

The Poisson equation (4.1) is discretized as follows: for each vertex $v_i \in \bar{S}$,

$$\sum_{v_j \sim v_i} w_{ij}(\bar{\eta} + d\bar{f})([v_i, v_j]) = 0,$$

it is simplified to

$$\sum_{v_j \sim v_i} w_{ij}(\bar{f}_j - \bar{f}_i) = -\sum_{v_j \sim v_i} w_{ij}\bar{\eta}([v_i, v_j]). \tag{4.3}$$

This leads to a linear system with the *discrete Laplace Beltrami* matrix $L = (l_{ij})$, where

$$l_{ij} := \begin{cases} w_{ij} & i \neq j \\ -\sum_k w_{ik} & i = j \end{cases} \tag{4.4}$$

The harmonic differential homological equivalent to $\bar{\eta}$ is given by $\bar{\omega} = \bar{\eta} + d\bar{f}$. Details can be found in Alg. 4.4

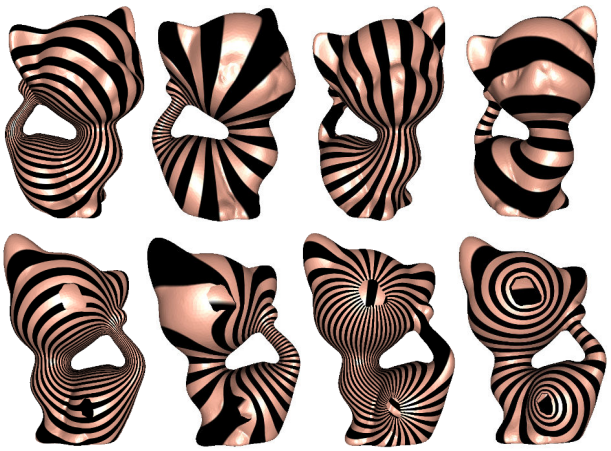


Figure 4: Harmonic differential group basis.

Algorithm 4.4. (Harmonic Differential Basis) Harmonic Differential Basis

Require: Input closed mesh S with genus g
Require: Basis of cohomology group $H_{1(S,\mathbb{Z})}$, $\{\eta_1, \eta_2, \ldots, \eta_{2g}\}$.
Ensure: Basis of harmonic differential group $H^1_\Delta(S, \mathbb{R})$, $\{\omega_1, \omega_2, \ldots, \omega_{2g}\}$.
function HarmonicDifferentialBasis(S)
  for each edge e_{ij} of S **do**
    Find the two adjacent faces f_{ijk} and f_{jil}
    Compute the edge weight using Eqn. 4.2
  end for
  Construct the Laplace-Beltrami matrix Eqn. 4.4
  for each basis differential η_k of $H^1(S, \mathbb{R})$ **do**
    for each basis vertex $v_i \in S$ **do**
      Compute the right hand side of Eqn. 4.3
    end for
    Solve the discrete Poisson equation 4.3 to get f_k
    $\omega_k \leftarrow \eta_k + df_k$
  end for
end function=0

For the purpose of visualizing a harmonic differential ω, we can define a function in the fundamental domain D by integrating ω on it, $f : D \to \mathbb{R}$, and use texture mapping to check the differential. First, we choose a base vertex $v_0 \in D$, then define

$$f(p) := \int_{v_0}^{p} \omega, \tag{4.5}$$

since ω is harmonic, it is closed, therefore the integration value is independent of the choice of the integration path. By mapping the surface to the line segment, we can visualize the level set of f, and the differential ω. Alg. 4.5 gives the details.

Algorithm 4.5. (Integration) Integration

Require: A fundamental domain D of a surface S
Require: A harmonic 1-form ω
Ensure: The integration function $f : D \to \mathbb{R}$ of ω on D
function Integration(D,ω)
  Choose a base vertex $v_0 \in D$,
  Set $f(v_0) \leftarrow 0$, $v_0.access \leftarrow$
  Enqueue v_0 to Q, $Q.push(v_0)$
  while Q is non-empty **do**
    $v \leftarrow$ Q.pop()
    for each edge $[v, w]$ in D **do**
      if not $w.access$ **then**
        $f(w) \leftarrow f(v) + \omega([v, w])$
        Set $w.access \leftarrow$
        Enqueue w to Q, $Q.push(w)$
      end if
    end for
  end while
end function

Fig. 4 shows the basis of the harmonic differential group of the doubled kitten mesh with two holes. The upper row shows the front views and the lower row shows the back views. Each column shows the harmonic differential corresponds to the homology base loop in the same column of Fig. 3.

4.4 Abel Differentials In this step, we construct the first type Abel differentials (holomorphic differentials) on the doubled surface $\bar{S}$, then the second type

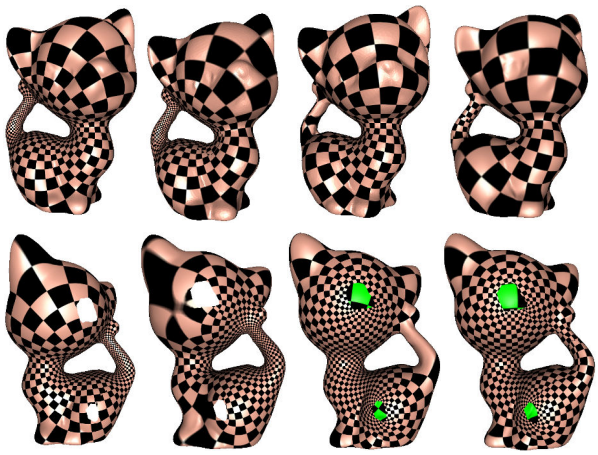


Figure 5: Holomorphic differential group basis.

of Abel differentials on the original surface with boundaries and negative Euler characteristic numbers.

On a compact genus g Riemann surface S, each holomorphic 1-form can be decomposed into a pair of conjugate harmonic 1-forms $\omega + \sqrt{-1}{}^*\omega$. Since ω is harmonic, ${}^*\omega$ is also harmonic and therefore can be represented as a linear combination of the basis of the harmonic differential group $H^1_\Delta(S,\mathbb{R})$. Suppose $\{\omega_1, \omega_2, \ldots, \omega_{2g}\}$ is a basis of $H^1_\Delta(S,\mathbb{R})$, assume ${}^*\omega_k$ can be represented as $\sum_j \lambda_{kj}\omega_j$. We obtain the linear system,

$$\begin{pmatrix} \omega_1 \wedge {}^*\omega_k \\ \omega_2 \wedge {}^*\omega_k \\ \vdots \\ \omega_{2g} \wedge {}^*\omega_k \end{pmatrix} = \begin{pmatrix} \omega_1 \wedge \omega_1 & \cdots & \omega_1 \wedge \omega_{2g} \\ \omega_2 \wedge \omega_1 & \cdots & \omega_2 \wedge \omega_{2g} \\ \vdots & & \vdots \\ \omega_{2g} \wedge \omega_1 & \cdots & \omega_{2g} \wedge \omega_{2g} \end{pmatrix} \begin{pmatrix} \lambda_{k,1} \\ \lambda_{k,2} \\ \vdots \\ \lambda_{k,2g} \end{pmatrix}$$

We can compute the integration of 2-forms on each triangular face. Given a Euclidean triangle $\Delta = [\mathbf{v}_i, \mathbf{v}_j, \mathbf{v}_k]$, the edge vector is $\mathbf{e}_i = \mathbf{v}_k - \mathbf{v}_j$, suppose ω_1 and ω_2 are constant 1-forms on Δ, then

$$\int_\Delta \omega_1 \wedge \omega_2 = \frac{1}{6} \begin{vmatrix} \omega_1(\mathbf{e}_i) & \omega_1(\mathbf{e}_j) & \omega_1(\mathbf{e}_k) \\ \omega_2(\mathbf{e}_i) & \omega_2(\mathbf{e}_j) & \omega_2(\mathbf{e}_k) \\ 1 & 1 & 1 \end{vmatrix} \tag{4.6}$$

Furthermore, the wedge product with the conjugate differential form has

$$\int_\Delta \omega_1 \wedge {}^*\omega_2 = \frac{1}{2} \sum_i \cot\theta_i \omega_1(\mathbf{e}_i)\omega_2(\mathbf{e}_i). \tag{4.7}$$

The conjugate harmonic 1-forms can be evaluated using Alg. 4.6.

Algorithm 4.6. (Conjugate Harmonic Differential) Conjugate Harmonic Differential

Require: A closed genus g surface S

Require: Harmonic differential group basis $\{\omega_k\}_{k=1}^{2g}$
Ensure: Conjugate harmonic differentials $\{{}^*\omega_k\}_{k=1}^{2g}$
function HodgeStar(S,$\{\omega_k\}$)
for each harmonic differential ω_k **do**
for each harmonic differential ω_i **do**
Compute $\int_S \omega_i \wedge \omega_k$ using Eqn. 4.6
Compute $\int_S \omega_i \wedge {}^*\omega_k$ using Eqn. 4.7
end for
Solve the linear system obtain $\lambda_{k,i}$'s
Set ${}^*\omega_k \leftarrow \sum_{i=1}^{2g} \lambda_{k,i}\omega_i$
end for
end function

Each harmonic differential ω_k paired with its conjugate ${}^*\omega_k$ forms a holomorphic differential

$$\Omega_k := \omega_k + \sqrt{-1}{}^*\omega_k, \tag{4.8}$$

$\{\Omega_1, \Omega_2, \ldots, \Omega_{2g}\}$ forms the basis for all holomorphic differentials of S, namely $\Omega(S)$. Fig. 5 shows the holomorphic differentials, whose real parts are the harmonic differentials in Fig. 4 respectively. The upper row shows the front views and the lower row shows the back views.

4.5 Symmetrization In practice, we prefer symmetric holomorphic differentials on the doubled surface $\bar{S}$. A holomorphic differential $h \in \Omega(\bar{S})$ is *symmetric*, if for any point on the boundary of the original surface $p \in \partial S$, or equivalently, $\bar{p} \in \bar{S}$, $\varphi(\bar{p}) = \bar{p}$, there is a neighborhood $U(\bar{p}) \subset \bar{S}$ with local parameter z, $h(z) = f(z)dz$,

$$f(z) = \overline{f(\bar{z})}, \tag{4.9}$$

where $\bar{c}$ means the conjugate of the complex number c. Or globally, for each edge $e \in \bar{S}$,

$$h(e) = \overline{h \circ \varphi(e)}, \tag{4.10}$$

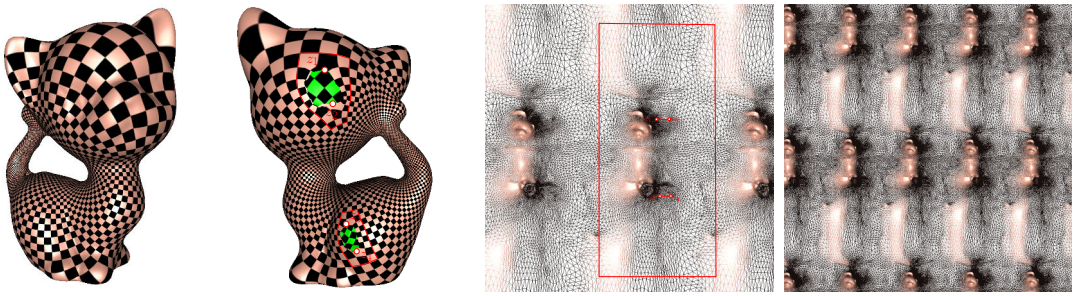
where φ is the convolution map of the doubled mesh $\bar{S}$.

In our construction algorithm, the holomorphic 1-form basis h corresponding to γ is denoted h_γ. Then

$$\bar{h}_{a_i} := \bar{h}_{\bar{a}_i^+} + \bar{h}_{\bar{a}_i^-}, \quad \bar{h}_{b_i} := \bar{h}_{\bar{b}_i^+} + \bar{h}_{\bar{b}_i^-}$$

are symmetric. Fig. 6 shows the symmetric differential corresponding to the tunnel loop h_{a_i}. All zero points are in the boundaries of S as shown in frame (b), two boundaries are mapped to parallel slits in frame (c). The whole surface S can be periodically mapped onto the plane as shown in frame (d). Fig. 7 shows the differential corresponding to the handle loop h_{b_i}. Similarly, the zeros are in the boundaries and the boundaries are mapped to parallel slits.

The holomorphic differential $\bar{h}_{\beta_k}$ corresponds to the bridge loop β_k, for any edge on the boundary $e \in \partial S$,



a. front view b. back view c. one period d. periods

Figure 6: Symmetric Abel differential (corresponds to $\bar{a}_1^+ + \bar{a}_1^-)h = f(z)dz$ satisfying $f(z) = \overline{f(\bar{z})}$ in a local chart near the boundary, the boundaries are mapped to horizontal slits.

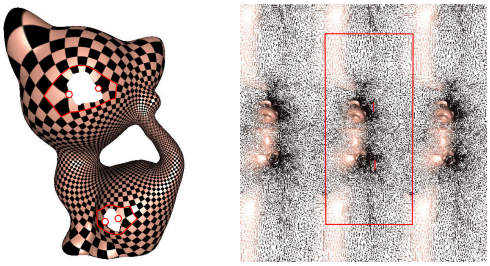


Figure 7: Symmetric Abel differential corresponds to $\bar{b}_1^+$ plus $\bar{b}_1^-$, the boundaries are mapped to vertical slits.

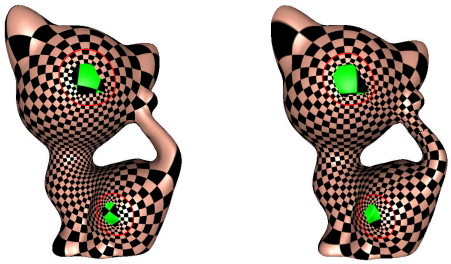


Figure 8: Symmetric Abel differentials correspond to $\bar{\alpha}_1$ and $\bar{\beta}_1$ respectively. The neighborhoods of boundaries are mapped to cylinders.

$\bar{h}_{\beta_k}(e)$ is a real number, $\bar{h}_{\beta_k}$ is symmetric. Fig. 8 left frame shows $\bar{h}_{\beta_1}$, we can see that there is no zero point on the boundaries, hence the neighborhoods of boundaries are mapped to cylinders. For $\bar{h}_{\alpha_k}$ corresponding to a boundary loop α_k, $\bar{h}_{\alpha_k}(e)$ is imaginary. In this case, we multiply $\bar{h}_{\alpha_k}$ by $\sqrt{-1}$ to make it symmetric. Fig. 8 right frame shows $\sqrt{-1}\bar{h}_{\alpha_1}$.

Then we obtain a basis for symmetric holomorphic differentials are the doubled surface $\bar{S}$,

$$\bigcup_{k=1}^{g} \{\bar{h}_{a_k}, \bar{h}_{b_k}\}, \bigcup_{i=1}^{n-1} \{\sqrt{-1}\bar{h}_{\alpha_i}, \bar{h}_{\beta_i}\}.$$

The symmetric holomorphic differential basis on the doubled surface $\bar{S}$ can be projected to the original surface S, which form a holomorphic differential basis of S,

$$\bigcup_{k=1}^{g} \{h_{a_k}, h_{b_k}\}, \bigcup_{i=1}^{n-1} \{\sqrt{-1}h_{\alpha_i}, h_{\beta_i}\}, \tag{4.11}$$

such that these holomorpic differentials restricted on the boundaries of S are real. Furthermore, from Fig. 7 we can see that the $\sqrt{-1}h_{\alpha_i}$ and h_{β_1} are the third type Abel differentials, the boundaries can be extended to cusps to become the poles of the differentials.

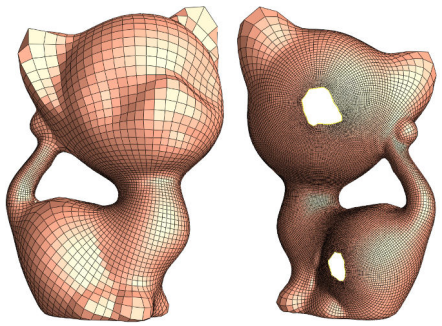


Figure 9: The quad-mesh produced by the symmetric Abel differential h_{alpha_1}.

4.6 Quadrilateral Mesh Generation Given a surface with genus g and n boundaries, we have compute the basis for $H_1(S, \mathbb{Z})$, tunnel and handle loops $\{a_i, b_i\}_{i=1}^g$ and boundary loops $\{\alpha_1, \ldots, \alpha_{n-1}\}$. We relabel them as γ_k's, $k = 1, 2, \ldots, 2g + n - 1$. We also compute the holomorphic differential basis Eqn. 4.11, whose real part form the basis of $H_\Delta(S, \mathbb{R})$, denoted as $\{\omega_1, \omega_2, \ldots \omega_{2g+n-1}\}$. We construct another harmonic differential group basis, $\{\omega_k'\}$, which is dual to the homology basis $\{\gamma_k\}$, namely

$$\int_{\gamma_i} \omega_j' = \delta_{ij}, \quad i, j = 1, 2, \ldots, 2g + n - 1.$$

Assume $\omega_k' = \sum_j \mu_{k,j}\omega_j$, then we construct a linear system

$$\begin{pmatrix} \int_{\gamma_1}\omega_1 & \cdots & \int_{\gamma_1}\omega_{2g+n-1} \\ \int_{\gamma_2}\omega_1 & \cdots & \int_{\gamma_2}\omega_{2g+n-1} \\ \vdots & & \vdots \\ \int_{\gamma_k}\omega_1 & \cdots & \int_{\gamma_{2g+n-1}}\omega_k \\ \vdots & & \vdots \\ \int_{\gamma_{2g+n-1}}\omega_1 & \cdots & \int_{\gamma_{2g+n-1}}\omega_{2g+n-1} \end{pmatrix} \begin{pmatrix} \mu_{k,1} \\ \mu_{k,2} \\ \vdots \\ \mu_{k,k} \\ \vdots \\ \mu_{k,2g+n-1} \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix}$$

By linear combine the basis in Eqn. 4.11 to obtain a holomorphic differential h, with periods $\lambda_k := \int_{\gamma_k} h$, $k = 1, \ldots, 2g+n-1$. In general λ_k's are real numbers, may be irrationals. We choose a positive integer $q \in \mathbb{Z}^+$, and quantize all the periods to be rational numbers in the form p/q. For each k, find the rational number

$$\{q\Re(\lambda_k)\} = q\Re(\lambda_k) - [q\Re(\lambda_k)],$$
$$\{q\Im(\lambda_k)\} = q\Im(\lambda_k) - [q\Im(\lambda_k)]$$

where $[x]$ is the operator of taking the integer part of a real number $x \in \mathbb{R}$, $\Re(z)$, $\Im(z)$ taking the real and the imaginary part of a complex number z respectively. We construct a complex differential,

$$h' = h - \frac{1}{q}\sum_{k=1}^{2g+n-1} \left(\{q\Re(\lambda_k)\} + \sqrt{-1}\,\{q\Im(\lambda_k)\}\right)\omega_k'$$

Then all the periods of h' are rational numbers in the form p/q. The integration of h' on the fundamental domain D maps D to the complex plane using the Alg. 4.5 with h' as the input differential, this gives a parameterization of the surface. On the complex plane, we construct the regular grid with size $1/q$, and pull the grid back onto the surface using the parameterization. Since the periods of h' are p/q, therefore the pull-backed grid on the surface form a quadrilateral tessellation (the grid lines aligned consistently on the surface across the boundaries of the fundamental domain).

Fig. 9 shows the quad-mesh produced by the symmetric holomorphic differential corresponding to the boundary loop α_1. We can see that each cell is similar to the planar unit square, all the corners are close to right angles, the grid lines are either parallel or orthogonal to the boundaries.

5 Experimental Results

We use the fertilty model to validate our proposed algorithms. The model is of the genus 4 and has 3 boundaries. Fig. 11 and Fig. 12 show the computational process. The top row, middle and bottom rows show the basis of the homology group $H_1(S, \mathbb{Z})$, the basis of the harmonic differential group $H^1_\Delta(S, \mathbb{R})$ and the basis of the holomorphic differential group $\Omega^1(S)$. Each frames of the same column correspond to the same homology basis. The upper half of Fig. 11 shows the intermediate results corresponding to the tunnel loops a_1, a_2, a_3 and a_4; the lower half shows those corresponding to the handle loops b_1, b_2, b_3, b_4; The left part of Fig. 12 shows computational process corresponding to the boundary loops α_1, α_2; the right part shows those corresponding to the birdge loops β_1, β_2.

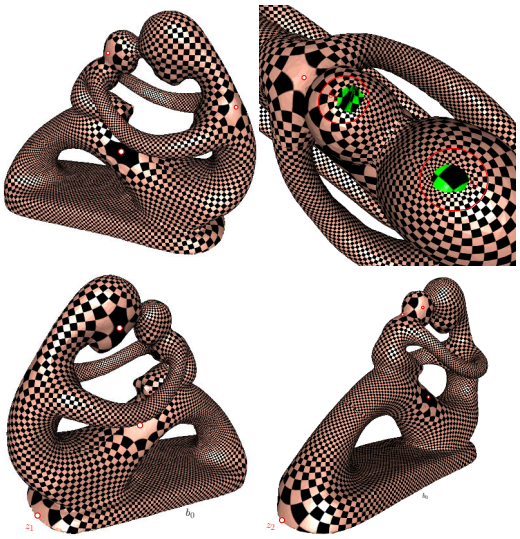


Figure 10: The zeros and poles (top right frame) on a symmetric Abel differential.

Fig. 10 shows the zeros and poles of a symmetric Abel differential ω in the fertilty model. ω is real along the boundaries. There are two zeros z_1, z_2 on the boundary b_0, and no zeros on b_1, b_2. Therefore, the two boundaries b_1 and b_2 are poles of the differential.

Fig. 13 illustrates some quad-meshes induced by symmetric Abel differentials. We can see the grid lines are orthogonal to each other, and either parallel or perpendicular to the boundaries. Furthermore, the quad-meshes are with minimal number singularities.

6 Conclusion

This work proposes a quadrilateral mesh generation method for the surfaces with multiple boundaries and negative Euler characteristic numbers, based on symmetric Abel differentials. The method has solid theoretic foundations and produces high quality quad-meshes with high orthogonality and least number of singularities. Furthermore, the grid lines align with the boundaries.

In the future, we plan to generalize the proposed method to symmetric Abel quadratic differentials to further improve the flexibility.

Figure 11: The tunnel loops (upper half) and handle loops (lower half).

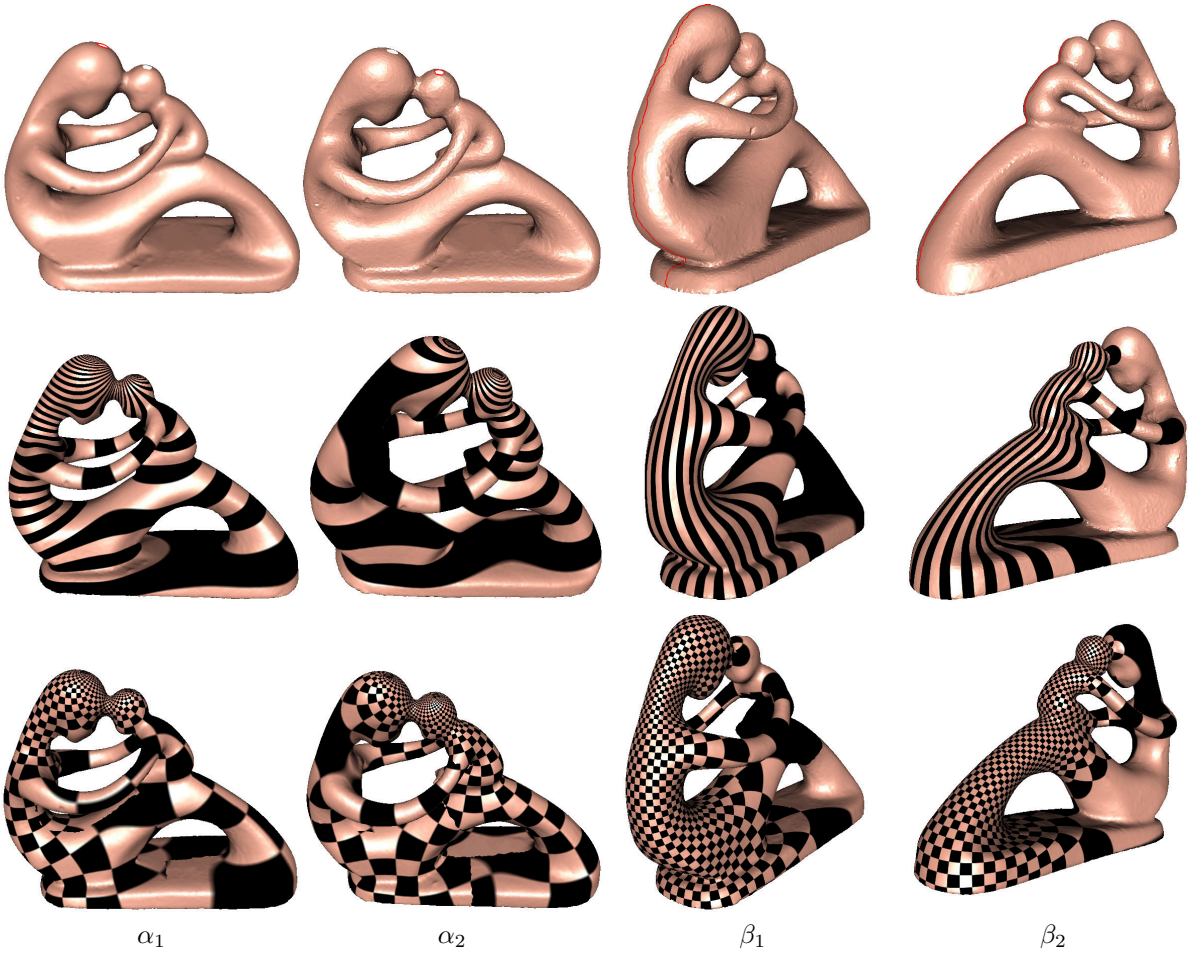


Figure 12: The boundary (left two frames) and the bridge (right two frames) loops of the fertility model.

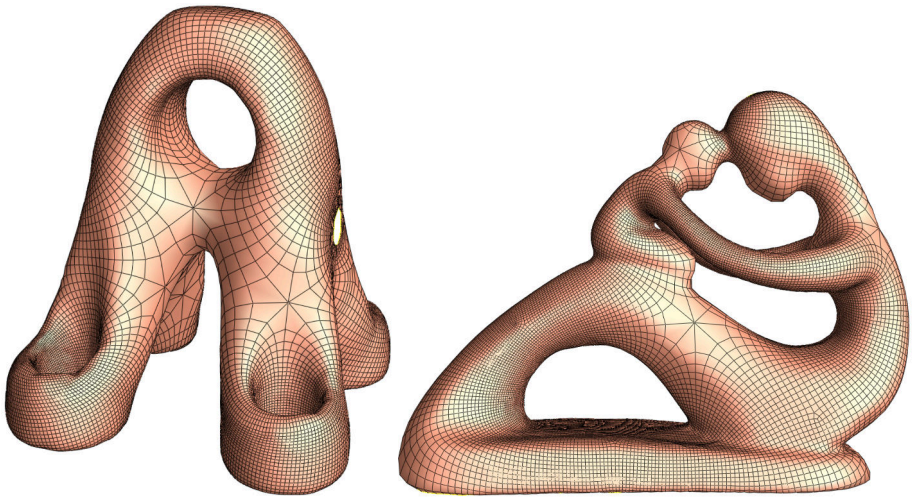


Figure 13: The quad-mesh produced by the symmetric Abel differential.

References

[1] T. D. Blacker and M. B. Stephenson, *Paving: A new approach to automated quadrilateral mesh generation*, International journal for numerical methods in engineering, 32 (1991), pp. 811–847.

[2] I. Boier-Martin, H. Rushmeier, and J. Jin, *Parameterization of triangle meshes over quadrilateral domains*, in Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing, 2004, pp. 193–203.

[3] D. Bommes, B. Lévy, N. Pietroni, E. Puppo, C. Silva, M. Tarini, and D. Zorin, *Quad-mesh generation and processing: A survey*, in Computer graphics forum, vol. 32, Wiley Online Library, 2013, pp. 51–76.

[4] S. A. Canann, J. R. Tristano, and M. L. Staten, *A survey of quadrilateral mesh generation technology*, International Journal for Numerical Methods in Engineering, 40 (1998), pp. 4113–4130.

[5] N. A. Carr, J. Hoberock, K. Crane, and J. C. Hart, *Rectangular multi-chart geometry images*, in Symposium on geometry processing, 2006, pp. 181–190.

[6] E. Catmull and J. Clark, *Recursively generated b-spline surfaces on arbitrary topological meshes*, Computer-aided design, 10 (1978), pp. 350–355.

[7] W. Chen, X. Zheng, J. Ke, N. Lei, Z. Luo, and X. Gu, *Quadrilateral mesh generation i: Metric based method*, Computer Methods in Applied Mechanics and Engineering, 356 (2019), pp. 652–668.

[8] T. K. Dey, F. Fan, and Y. Wang, *An efficient computation of handle and tunnel loops via reeb graphs*, ACM Transactions on Graphics (TOG), 32 (2013), pp. 1–10.

[9] S. Donaldson, *Riemann surfaces*, Oxford University Press, 2011.

[10] S. Dong, P.-T. Bremer, M. Garland, V. Pascucci, and J. C. Hart, *Spectral surface quadrangulation*, in ACM SIGGRAPH 2006 Papers, 2006, pp. 1057–1066.

[11] X. Gu and S.-T. Yau, *Computing conformal structure of surfaces*, arXiv preprint cs/0212043, (2002).

[12] X. Gu and S.-T. Yau, *Computational Conformal Geometry*, vol. 3 of Advanced Lectures in Mathematics, International Press and Higher Education Press, 2007.

[13] ———, *Global conformal surface parameterization*, in First Eurographics Symposium on Geometry Processing (SGP03), Aachen, Germany, June 23-25, 2003, pp. 127–137.

[14] T. Gurung, D. Laney, P. Lindstrom, and J. Rossignac, *Squad: Compact representation for triangle meshes*, in Computer Graphics Forum, vol. 30, Wiley Online Library, 2011, pp. 355–364.

[15] Y. He, H. Wang, C.-W. Fu, and H. Qin, *A divide-and-conquer approach for automatic polycube map construction*, Computers & Graphics, 33 (2009), pp. 369–380.

[16] J. Huang, M. Zhang, J. Ma, X. Liu, L. Kobbelt, and H. Bao, *Spectral quadrangulation with orientation and alignment control*, in ACM SIGGRAPH Asia 2008 papers, 2008, pp. 1–9.

[17] T. Jiang, X. Fang, J. Huang, H. Bao, Y. Tong, and M. Desbrun, *Frame field generation through metric customization*, ACM Transactions on Graphics (TOG), 34 (2015), pp. 1–11.

[18] M. Jin, J. Kim, F. Luo, and X. Gu, *Discrete surface ricci flow*, IEEE Transaction on Visualization and Computer Graphics (TVCG), 14 (2008), pp. 1030–1043.

[19] F. Kälberer, M. Nieser, and K. Polthier, *Quadcover-surface parameterization using branched coverings*, in Computer graphics forum, vol. 26, Wiley Online Library, 2007, pp. 375–384.

[20] N. Kowalski, F. Ledoux, and P. Frey, *A pde based approach to multidomain partitioning and quadrilateral meshing*, in Proceedings of the 21st international meshing roundtable, Springer, 2013, pp. 137–154.

[21] Y.-K. Lai, M. Jin, X. Xie, Y. He, J. Palacios, E. Zhang, S.-M. Hu, and X. Gu, *Metric driven rosy field design and remeshing*, IEEE Transaction on Visualization and Computer Graphics (TVCG), 16 (2009), pp. 95–108.

[22] N. Lei, X. Zheng, J. Jiang, Y.-Y. Lin, and D. X. Gu, *Quadrilateral and hexahedral mesh generation based on surface foliation theory*, Computer Methods in Applied Mechanics and Engineering, 316 (2017), pp. 758–781.

[23] N. Lei, X. Zheng, Z. Luo, and D. X. Gu, *Quadrilateral and hexahedral mesh generation based on surface foliation theory ii*, Computer Methods in Applied Mechanics and Engineering, 321 (2017), pp. 406–426.

[24] N. Lei, X. Zheng, Z. Luo, F. Luo, and X. Gu, *Quadrilateral mesh generation ii: Meromorphic quartic differentials and abel–jacobi condition*, Computer methods in applied mechanics and engineering, 366 (2020), p. 112980.

[25] N. Lei, Y. Zhu, X. Zheng, H. Si, and X. Gu, *Why cross fields are not equivalent to quadrilateral meshes*, CMAME, (2023).

[26] W.-C. Li, B. Vallet, N. Ray, and B. Lévy, *Representing higher-order singularities in vector fields on piecewise linear surfaces*, IEEE Transactions on Visualization and Computer Graphics, 12 (2006), pp. 1315–1322.

[27] J. Lin, X. Jin, Z. Fan, and C. C. Wang, *Automatic polycube-maps*, in Advances in Geometric Modeling and Processing: 5th International Conference, GMP 2008, Hangzhou, China, April 23-25, 2008. Proceedings 5, Springer, 2008, pp. 3–16.

[28] S. Lo, *A new mesh generation scheme for arbitrary planar domains*, International journal for numerical methods in engineering, 21 (1985), pp. 1403–1426.

[29] X. Ma and L. Sun, *An automatic approach to constrained quadrilateral mesh generation*, Engineering Computations, 37 (2020), pp. 929–951.

[30] R. Miranda, *Algebraic Curves and Riemann Surfaces*, vol. 5 of Graduate Studies in Mathematics, American Mathematical Society, Providence, RI, 1995.

[31] J. Palacios and E. Zhang, *Rotational symmetry field design on surfaces*, ACM Transactions on Graphics (TOG), 26 (2007), pp. 55–es.

[32] N. Ray, W. C. Li, B. Lévy, A. Sheffer, and P. Alliez, *Periodic global parameterization*, ACM Transactions on Graphics (TOG), 25 (2006), pp. 1460–1485.

[33] N. Ray and D. Sokolov, *Robust polylines tracing for n-symmetry direction field on triangulated surfaces*, ACM Transactions on Graphics (TOG), 33 (2014), pp. 1–11.

[34] J.-F. Remacle, J. Lambrechts, et al., *Advances in quadrilateral mesh generation: Meshing roundtable review*, in 21st International Meshing Roundtable, Springer, 2012, pp. 129–146.

[35] J.-F. Remacle, J. Lambrechts, B. Seny, E. Marchandise, A. Johnen, and C. Geuzainet, *Blossom-quad: A non-uniform quadrilateral mesh generator using a minimum-cost perfect-matching algorithm*, International journal for numerical methods in engineering, 89 (2012), pp. 1102–1119.

[36] K. M. Shepherd, X. D. Gu, R. R. Hiemstra, and T. J. Hughes, *Quadrilateral layout generation and optimization using equivalence classes of integral curves: theory and application to surfaces with boundaries*, Journal of Mechanics, 38 (2022), pp. 128–155.

[37] K. M. Shepherd, X. D. Gu, and T. J. Hughes, *Feature-aware reconstruction of trimmed splines using ricci flow with metric optimization*, Computer Methods in Applied Mechanics and Engineering, 402 (2022), p. 115555.

[38] M. Tarini, N. Pietroni, P. Cignoni, D. Panozzo, and E. Puppo, *Practical quad mesh simplification*, in Computer Graphics Forum, vol. 29, Wiley Online Library, 2010, pp. 407–418.

[39] Y. Tong, P. Alliez, D. Cohen-Steiner, and M. Desbrun, *Designing quadrangulations with discrete harmonic forms*, in Eurographics symposium on geometry processing, 2006.

[40] L. Velho and D. Zorin, *4–8 subdivision*, Computer Aided Geometric Design, 18 (2001), pp. 397–427.

[41] H. Wang, M. Jin, Y. He, X. Gu, and H. Qin, *User-controllable polycube map for manifold spline construction*, in Proceedings of the 2008 ACM symposium on Solid and physical modeling, 2008, pp. 397–404.

[42] L. Wang, X. Gu, K. Mueller, and S.-T. Yau, *Uniform texture synthesis and texture mapping using global parameterization*, The Visual Computer, 21 (2005), pp. 801–810.

[43] D. R. White and P. Kinney, *Redesign of the paving algorithm: Robustness enhancements through element by element meshing*, in 6th international meshing roundtable, vol. 10, 1997, p. 830.

[44] J. Xia, I. Garcia, Y. He, S.-Q. Xin, and G. Patow, *Editable polycube map for gpu-based subdivision surfaces*, in Symposium on interactive 3D graphics and games, 2011, pp. 151–158.

[45] Y. Zhao, B. Yu, and W. Tao, *An improved paving method of automatic quadrilateral mesh generation*, Numerical Heat Transfer, Part B: Fundamentals, 64 (2013), pp. 218–238.

[46] Z. Zhao, S. Fang, N. Lei, Y. Liu, Y. Zhu, C. Sadasivan, A. Tassiopoulos, S. Chen, and X. Gu, *Optimal Surface Quadrilateral Mesh Generation*, pp. 14–27.

[47] X. Zheng, Y. Zhu, W. Chen, N. Lei, Z. Luo, and X. Gu, *Quadrilateral mesh generation iii: Optimizing singularity configuration based on abel–jacobi theory*, Computer Methods in Applied Mechanics and Engineering, 387 (2021), p. 114146.

[48] O. C. Zienkiewicz, R. L. Taylor, and J. Z. Zhu, *The finite element method: its basis and fundamentals*, Elsevier, 2005.