Structured Orthogonal Near-Boundary Voronoi Mesh Layers for Planar Domains

Vladimir Garanzha, Liudmila Kudryavtseva, and Valeriia Tsvetkova

Abstract We consider the problem of constructing a purely Voronoi mesh where the union of uncut Voronoi cells approximates the planar computational domain with piecewise-smooth boundary. Smooth boundary fragments are approximated by the Voronoi edges and Voronoi vertices are placed near summits of sharp boundary corners. We suggest a self-organization meshing algorithm which covers the boundary of domain by an almost-structured band of non-simplicial Delaunay cells. This band consists of quadrangles on the smooth boundary segment and convex polygons around sharp corners. The dual Voronoi mesh is a double-layered orthogonal structure where the central line of the layer approximates the boundary. The overall Voronoi mesh has a hybrid structure and consists of high quality convex polygons in the core of the domain and orthogonal layered structure near boundaries. We introduce refinement schemes for the Voronoi boundary layers, in particular near sharp corners and discuss problems related to the generalization of the suggested algorithm in 3d.

V. Garanzha (✉)
Dorodnicyn Computing Center FRC CSC RAS, Moscow, Russia

Moscow Institute of Physics and Technology, Moscow, Russia
e-mail: garan@ccas.ru; http://www.ccas.ru/gridgen/lab

L. Kudryavtseva
Dorodnicyn Computing Center FRC CSC RAS, Moscow, Russia

Moscow Institute of Physics and Technology, Moscow, Russia

Keldysh Institute of Applied Mathematics RAS, Moscow, Russia

V. Tsvetkova
Keldysh Institute of Applied Mathematics RAS, Moscow, Russia

V. A. Garanzha et al. (eds.), *Numerical Geometry, Grid Generation and Scientific Computing*, Lecture Notes in Computational Science and Engineering 131,
https://doi.org/10.1007/978-3-030-23436-2_2

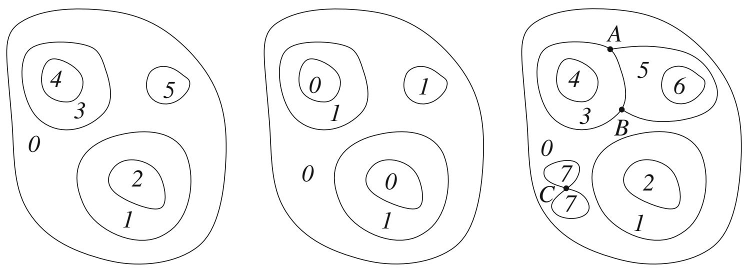
1 Introduction

Construction of hybrid polyhedral meshes in complicated 3d domains is an interesting and actively developing field of mesh generation. A well established approach to polyhedral meshing is based on construction of a tetrahedral mesh and its approximate dualization [1, 2]. In most cases, this technique produces high quality polyhedra. Unfortunately, near the boundary, it creates a number of cut cells which should be optimized in order to obtain an acceptable mesh. Optimality criteria in most cases are contradictory. Hence, costly multicriterial optimization is needed with uncertain outcome. A good solution might be a construction of a Voronoi polyhedral mesh with full uncut Voronoi cells near the boundary. However, we are not aware of such an algorithms. Hence, the goal of the paper is to try to construct an algorithm which solves the above problem in 2d, at least in the practical sense, before treating the more complicated 3d case.

Note that domain approximation by Voronoi tilings and their generalization has a rich history, especially in surface reconstruction problems [3]. Many algorithms for construction and optimization of Voronoi meshes were suggested, see [4–7]. Unfortunately, these algorithms are not suitable to build Voronoi meshes with regular Voronoi layers near boundaries, which is the topic of the present research.

1.1 Definition of a Multimaterial Implicit Domain

Consider a bounded domain Ω which is partitioned into N subdomains Ω_i, $i = 0, \ldots, N-1$. Intuitively, imagine a body of different materials glued together. We assume that the boundary of each subdomain is piece-wise smooth and Lipschitz continuous. The simplest case of multimaterial domain is based on two assumptions: (a) the boundary of each subdomain is a manifold and (b) multimaterial vertices with neighborhoods containing more than two materials are absent. An example of such a domain is shown in Fig. 1a.



(a) (b) (c)

Fig. 1 Domain examples

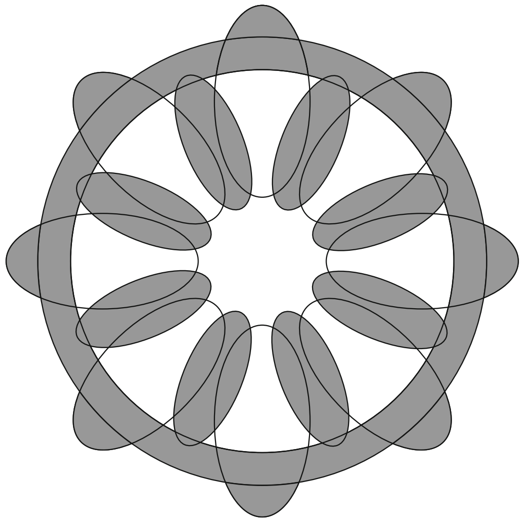


Fig. 2 Model "wheel": construction of an implicit domain using Boolean operations

The mesh generation problem in this multimaterial domain is equivalent to the mesh generation problem in the bimaterial domain in Fig. 1b. Such a domain van be modelled by a single scalar function $u(x)\colon \mathbb{R}^d \to \mathbb{R}$ which is negative inside Ω_1, positive inside Ω_0, and the zero isosurface of this function is the boundary. A more complicated case is presented in Fig. 1c. Here, the two multimaterial vertices A, B and a non-Lipschitz vertex C are present. Potentially, the meshing algorithm described below can be applied in this case as well but we have not tested such a configurations yet.

Boolean operations can be used to build quite complicated domains from primitives. Figure 2 shows (in gray) a planar domain which we use as a test case for the meshing algorithm.

It is assumed that the function $u(x)$ is piecewise smooth, Lipschitz continuous, and its derivatives along certain vector field transversal to internal boundary Γ are not equal to zero in a finite layer around the boundary. In fact, it is assumed that the behavior of the implicit function resembles that of the signed distance function. In particular, we always assume that the norm of $\nabla u(x)$, when defined, is bounded from below and from above in a certain layer around Γ.

1.2 Voronoi Mesh in an Implicit Domain

Consider a planar mesh $\mathscr{D}$ consisting of convex polygons D_i inscribed into circles B_i, as shown in Fig. 3b. D_i is a convex envelope of all mesh vertices lying on ∂B_i. Each circle is empty in a sense that it does not contain any mesh vertices inside. Such a mesh is called a Delaunay mesh (Delaunay partitioning). Considering convex envelope of all centers c_i of circles B_i passing through the Delaunay vertex p_k we

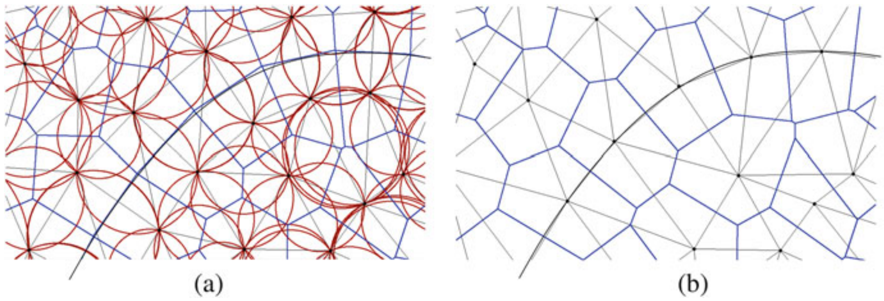


Fig. 3 Domain boundary approximation: (**a**) by Delaunay edges, (**b**) by Voronoi edges

get the Voronoi cell V_k. The set of Voronoi cells constitutes a partition which is generally called a Voronoi diagram. Since, in our setting, the outer boundary is not approximated, we are not interested in infinite Voronoi cells, so we just call the resulting object a Voronoi mesh. Internal boundaries can be approximated by using a Delaunay mesh, as shown in Fig. 3a, or by a Voronoi mesh, see Fig. 3b.

Let us briefly explain the difference. Piecewise smooth boundary Γ is approximated by a system of polylines. It is assumed that, with mesh refinement, polylines converge to Γ in the following sense: (a) distance from each straight edge of polyline to a certain distinct simple arc of Γ should be small; (b) deviation of the normal to the straight edge from the exact normals on the arc should be small; (c) sharp vertices on Γ are approximated by sharp vertices on the polyline. For the Delaunay mesh, this polyline is built from Delaunay edges while for the Voronoi mesh polyline is constructed from Voronoi edges. Delaunay edges, dual to the boundary Voronoi edges, are orthogonal to the boundary. For smooth fragment of the boundary Delaunay cells should be quadrilaterals which make up a band covering the boundary. The midline of this band, consisting of Voronoi edges, approximates the boundary as shown in Fig. 3b. It is well known that state-of-the-art algorithms generate Delaunay triangulations and not general Delaunay partitions. But the edges which split boundary Delaunay cells into triangles have zero dual Voronoi edges and do not influence the Voronoi mesh.

The typical behaviour of a Delaunay-Voronoi mesh around a sharp boundary vertex is shown in Fig. 4. Regular Delaunay bands consisting of quads are glued together through a convex polygonal Delaunay cell. The number of sides in this polygon depends on the sharp vertex angle.

2 Voronoi Meshing Algorithm Based on Self-organization of Elastic Network

In order to build Voronoi meshes in domains with non-smooth boundary we adapt the algorithm from [8] which was originally developed for Delaunay meshing in 2d and 3d implicit domains with piecewise-smooth boundaries. The unknowns in the

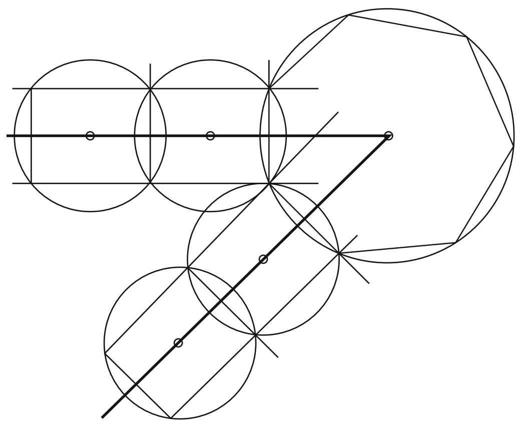


Fig. 4 Band of polygonal Delaunay cells and dual Voronoi edges on the domain boundary

presented algorithm are Delaunay mesh vertices which are considered as material points repulsing each other, thus modelling elastic medium. Repulsive forces are applied to each pair of vertices belonging to Delaunay edges, i.e., edges with circumferential open balls not containing any other vertices. Each Delaunay edge is treated as a compressible strut which tries to expand until prescribed length is reached. At each step, the dual Voronoi mesh is constructed and partitioned into two subdomains according to the value of implicit function in the Delaunay vertices. The Delaunay mesh is split into three subdomains: subdomain 0, subdomain 1, and a set of bands covering the boundary. All Delaunay triangles with circumcenters close to Γ are added to the bands. An approximate Voronoi boundary polyline is constructed. At this stage, mesh refinement is applied, provided that local minimum of the energy is attained. The idea of mesh refinement is to try to eliminate long Voronoi edges which are not orthogonal to the boundary, as explained in Fig. 5.

With each Voronoi edge we associate the "sharpening energy" and the "boundary attraction potential". Boundary attraction potential is used as a penalty term for obvious condition that each boundary Voronoi edge is tangential to Γ and touches it in a certain "touching point". Sharpening energy is minimized when the Voronoi edge e is orthogonal to the vector ∇u at the "touching point" for e. We use a special variant of preconditioned gradient search method to make one minimization step. It is convenient to call directions vectors in the minimization technique "elastic forces". When, due to point displacement under elastic forces, the edge loses Delaunay property, it should be excluded from the list of struts and new Delaunay edges should be created. Hence, the Voronoi mesh should be rebuilt as well. These steps are repeated until boundary is approximated with reasonable accuracy and correct topology of the near-boundary layers is recovered.

The outcome of the algorithm is a certain "equilibrium" mesh where elastic forces acting on each point sum to zero. As suggested in [9], we build the equilibrium mesh in the slightly compressed state.

2.1 Elastic Potential

Suppose that the system of points $\mathscr{E} = \{p_1, p_2, \ldots, p_n\}$ in $\mathbb{R}^2$ is prescribed. Let us denote by $\mathscr{T}(\mathscr{E})$ its Delaunay triangulation. We denote by $\mathscr{T}_e$ the set of triangulation edges and by $\mathscr{F}_b$ the set of the Delaunay edges crossing Γ. All vertices constitute $2 \times n$ matrix P with i-th column equal to p_i. We denote the set of the near-boundary Delaunay vertices by P_Γ. Voronoi mesh dual to $\mathscr{T}$ is denoted by $\mathscr{V}$, and the set of the Voronoi edges detected as a current guess to polyline approximating Γ is denoted by $\mathscr{E}_v$.

With each mesh $\mathscr{T}$ we associate the following elastic potential

$$W(P) = \theta_r W_r(P) + \theta_s W_s(P) + \theta_a W_a(P), \tag{1}$$

where $W_r(P)$ is the repulsion potential, $W_s(P)$ is the sharpening potential which serves to align Voronoi boundary edges along isolines of function u, $W_a(P)$ is the sharp edge attraction potential.

The repulsion potential is written as follows

$$W_r(P) = \sum_{e \in \mathscr{T}_e} w_r(e),$$

$$w_r(e) = \begin{cases} L_0^2 \left(\frac{L}{L_0} - 1 - \log\left(\frac{L}{L_0}\right)\right), & \text{when } L < L_0, \\ 0, & \text{when } L \geq L_0, \end{cases}$$

where

$$L = |p_i - p_j|$$

is the length of the edge e, and $L_0(e)$ is the target length of this edge defined by

$$L_0(e) = Mh\left(\frac{1}{2}(p_i + p_j)\right).$$

Here, $h(\cdot)$ is the relative sizing function, which is dimensionless, while the scalar multiplier M defines the actual length. As suggested in [9], the parameter M has the meaning of the average mesh edge length and may change slightly in the process of mesh self-organization. In practice, we use $L_0(e) = M(h(p_i) + h(p_j))/2$ in order to diminish the number of sizing function calls.

The sharpening functional is written as

$$W_s(P) = \sum_{e_v \in \mathscr{E}_v} w_s(e_v),$$

where the contribution from the boundary Voronoi edge e_v with vertices c_1 and c_2 looks like

$$w_s(e_v) = \frac{1}{2}|c_1 - c_2|\Big(n^T(c_2 - c_1)\Big)^2,$$

where

$$n = \frac{1}{|\nabla u(v^*)|}\nabla u(v^*) \tag{2}$$

and v^* is the current approximation of the touching boundary point for the Voronoi edge e_v. The simplest choice of v^* is the projection of the middle point of e_v onto Γ,

$$c = \frac{1}{2}(c_1 + c_2).$$

The Voronoi edge boundary attraction term is written as

$$W_a(P) = \sum_{e_v \in \mathscr{E}_v} w_a(e_v),$$

where

$$w_a(e_v) = \frac{1}{2}\left(\frac{L_0}{L}\right)^2 u^2(c)$$

Here, L is the length of the Delaunay edge dual to e_v. Hence, the energy assigned to shorter Delaunay edges is larger. Since unstable Delaunay edges, which serve to triangulate near-boundary, approximate Delaunay polygons in general are longer compared to stable edges, they have a smaller contribution to the total energy and have small influence on vertex positions.

2.2 "Elastic Forces" and Practical Iterative Algorithm

It is convenient to introduce the notions of "repulsive forces", "sharpening forces", and "boundary attraction forces" which denote the contribution to the direction vector from the repulsion, sharpening, and boundary attractions terms, respectively.

Roughly speaking, these "forces" are introduced as

$$\begin{aligned} \delta p_i^k &= -\frac{\theta_r}{d_{r_i}^k}\frac{\partial W_r}{\partial p_i}(P^k) - \frac{\theta_s}{d_{s_i}^k}\frac{\partial W_s}{\partial p_i}(P^k) - \frac{\theta_a}{d_{a_i}^k}\frac{\partial W_a}{\partial p_i}(P^k) \\ &= F_r(p_i^k) + F_s(p_i^k) + F_a(p_i^k), \end{aligned} \tag{3}$$

Here, k is the iteration number, p_i is the i-th vertex in the Delaunay mesh P^k, and ${d_r}_i^k$, ${d_s}_i^k$, and ${d_a}_i^k$ are the scaling factors.

Since Newton law is not used to describe the motion of mesh vertices, these "forces" are speculative and used just to facilitate the intuitive understanding of the algorithm.

In order to present precise formulae for computation of forces, it is convenient to introduce the following notations. Let $\mathrm{star}_e(p_i)$ denote the set of the mesh edges originating from the vertex p_i, while $\mathrm{star}(p_i)$ will denote the set of vertices of these edges excluding p_i. In all cases we assume that every boundary star is ordered, i.e., its entities are numbered counterclockwise around p_i looking from outside the domain. Below we omit the upper index k.

For the internal vertex p_i, the repulsive "force" looks like

$$F_r(p_i) = -\frac{\theta_r}{d_i} \sum_{p_j \in \mathrm{star}\, p_i} \phi_r(p_i, p_j)(p_i - p_j), \qquad d_{ri} = \sum_{p_j \in \mathrm{star}\, p_i} \phi_r(p_i, p_j),$$

where

$$\phi_r(p_i, p_j) = \left(\frac{L_0}{L} - 1\right)\frac{L_0}{L}, \qquad L = |p_i - p_j|, \qquad L_0 = \frac{Mh}{2}(p_i + p_j).$$

The sharpening force can be written as

$$F_s(p_i) = -\frac{\sum\limits_{e_v:\ p_i \in \mathrm{dual}\, e_v} \Pi_r(q|c_1 - c_2|n^T(c_1 - c_2))}{\sum\limits_{e_v: p_i \in \mathrm{dual}\, e_v} |c_1 - c_2||q|^2}.$$

Here, c_1 and c_2 are vertices of the edge e_v, $c = (c_1 + c_2)/2$, vector n is defined in (2), and

$$q = (C_2 - C_1)^T n, \qquad C_1 = \frac{\partial c_1}{\partial p_i}, \qquad C_2 = \frac{\partial c_2}{\partial p_i}.$$

In order to write down expression for matrix C_1, consider the Delaunay triangle T_1 with counterclockwise ordered vertices p_i, p_j, p_k whose circumcenter is c_1. Then,

$$C_1^T = (c_1 - p_i c_1 - p_i)(p_j - p_i p_k - p_i)^{-1}$$

The formula for C_2 is similar.

A nonlinear operator Π_r is responsible for interaction between repulsive force and sharpening force. Consider the contribution to $F_s(p_i)$ from Voronoi edge e_v. Let $e = (p_j - p_i)/|p_j - p_i|$, where p_i and p_j are vertices of Delaunay edge dual to e_v. If

$$e^T F_r(p_i) e^T q n^T (c_1 - c_2) < 0,$$

then

$$\Pi_r(qn^T(c_1 - c_2)) = qn^T(c_1 - c_2) - ee^T qn^T(c_1 - c_2).$$

Otherwise,

$$\Pi_r(qn^T(c_1 - c_2)) = qn^T(c_1 - c_2).$$

After local corrections for the sharpening terms, the assembled sharpening force at the vertex p_i is used in order to correct the repulsive force F_r:

$$F_r \leftarrow F_r - \frac{1}{2|F_s|^2} F_s (F_s^T F_r - |F_s^T F_r|).$$

The attraction force looks like

$$F_a(p_i) = - \sum_{e_v : p_i \in \text{dual}\, e_v} \frac{1}{2} \left(\frac{L_0}{L} \right)^2 u(c)(C_1 + C_2)^T \frac{\nabla u(c)}{|\nabla u(c)|}.$$

The displacement of Delaunay vertices is done in two steps. The first step is

$$\tilde{p}_i^0 = p_i^k + w_r \tau_r F_r + w_s \tau_s F_s, \qquad w_r = \frac{1}{20}, \qquad w_s = \frac{1}{2},$$

$$\tau_r = \min\left(1, \frac{L_0}{5 w_r F_r}\right), \qquad \tau_s = \min\left(1, \frac{L_0}{5 w_s F_s}\right).$$

After this displacement, we use M iterations with the attraction force to project Voronoi edges to the boundary,

$$\tilde{p}_i^{m+1} = \tilde{p}_i^m + \tau_a F_a(\tilde{p}_m^l), \qquad \tau_a = \frac{1}{10}.$$

Finally,

$$p_i^{k+1} = \tilde{p}_i^M.$$

3 Numerical Experiments

We ran series of numerical experiments with artificially constructed domains. The complexity of the tests is well represented by the model "wheel" shown in Fig. 2. In this model multiple sharp vertices are present on the boundary.

Numerical evidence suggests that algorithm recovers internal boundaries quite fast. However, this guess contains approximation defects and topological defects when near-boundary Voronoi edges are not orthogonal to boundary. The origin of these errors is simple: a Delaunay vertex does not have a good mirror vertex across the boundary. Hence, most of the topological errors can be eliminated by a reasonable Delaunay vertex insertion, as shown in Fig. 5. We consider a polygon P being the closest guess to the Delaunay polygon build upon two stable Delaunay edges e_1 and e_2 crossing the boundary. We build a quadrilateral cell upon these two edges and add new vertices at the middle of virtual opposite edges. The approximate Delaunay hexagon is resolved by inserting two vertices, while the approximate Delaunay pentagon is resolved by adding a single vertex. In our test cases there was no need to consider more complex polygons.

Figure 6 illustrates the Voronoi mesh evolution for am enlarged fragment of the "wheel" model.

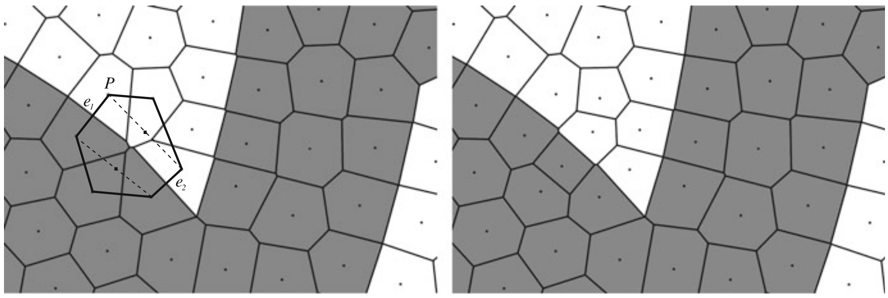


Fig. 5 Left: a fragment of a Voronoi mesh with non-orthogonal edges. Right: correct connectivity is attained by adding a new Delaunay vertex

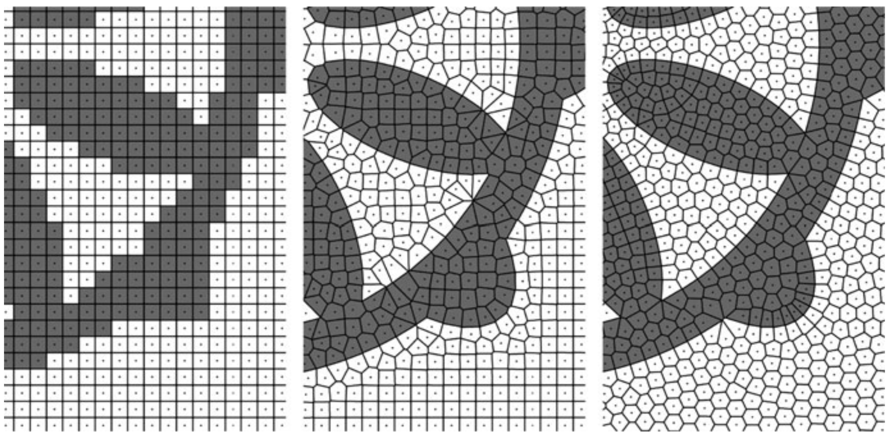


Fig. 6 Fragment of an initial Voronoi mesh, result after few iterations, and stabilized Voronoi mesh

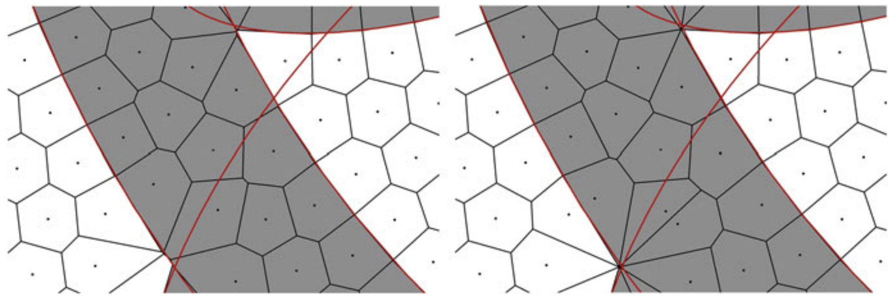


Fig. 7 Elimination of Voronoi faults: enlarged view

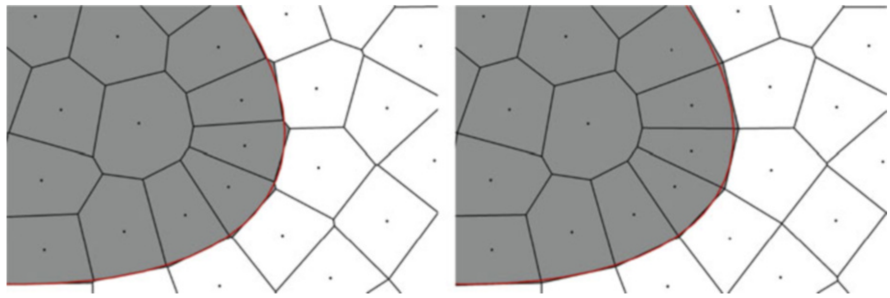


Fig. 8 Elimination of Voronoi faults: enlarged view

Figures 7 and 8 demonstrate that the elimination of short Voronoi edges does not lead to deterioration of the boundary approximation quality. To this end, we glue together nearby Delaunay circles and find new Delaunay vertices as intersections of corrected circles. We call the short boundary edges "Voronoi faults" by analogy with geology. The elimination of faults creates final mesh where the internal boundaries are approximated by Voronoi edges and the normals to the boundary are approximated by discrete normals.

Figure 9 illustrates the importance of corrections of corner Delaunay polygons. As one can see, after reduction of the Delaunay polygons to triangles, quadrilaterals, and pentagons, large Delaunay circles disappear and deviation from orthogonality for near-boundary Voronoi cells is reduced.

Figure 10 (center) shows a fragment of the final Voronoi mesh with boundary Delaunay circles, while the set of bands of Delaunay cells is shown on the right.

As soon as the basic layered structure of the Delaunay-Voronoi mesh is constructed, one can try to build anisotropic orthogonal Voronoi mesh layers using the following anisotropic refinement algorithm:

- new couple of vertices is added symmetrically on each Delaunay edge crossing the boundary;
- special refinement schemes are applied to the corner Delaunay polygons;

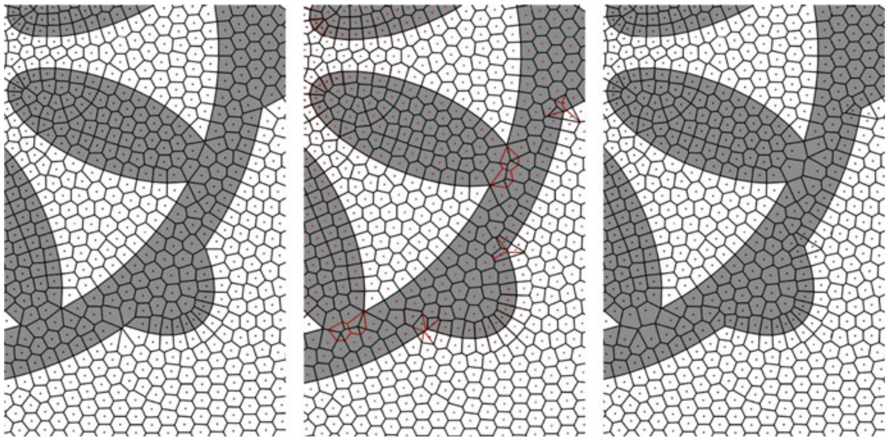


Fig. 9 (Left) Voronoi mesh before correction of corner Delaunay polygons, (right) Voronoi mesh after correction of corner Delaunay polygons, (center) overlapped meshes

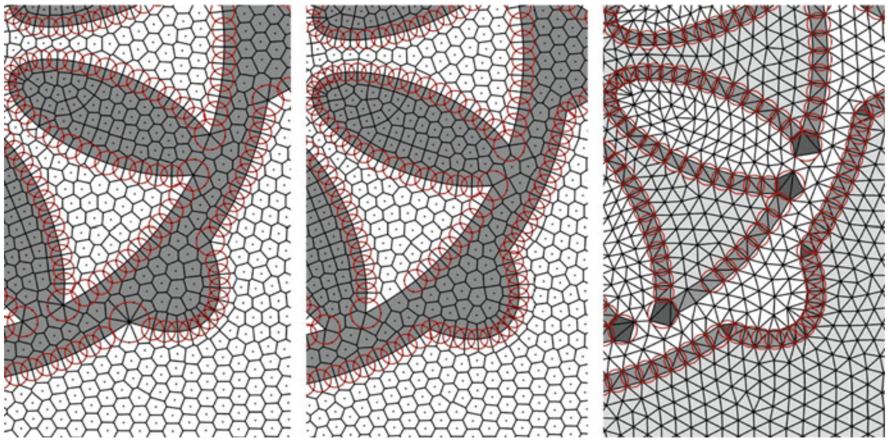


Fig. 10 Mesh after elimination of short Voronoi edges, Voronoi mesh after correction of Delaunay polygons, and a Delaunay layer covering boundary

- new Voronoi cells are computed, boundary Voronoi edges are projected onto the boundary;
- this inward-directed splitting procedure is repeated until required mesh compression rate is attained;
- the Voronoi fault filtering procedure based on close Delaunay circles gluing is applied starting from the boundary outward to the core of the domain;
- near corners the constrained Delaunay circle gluing procedure is applied.

The refinement of a regular Delaunay band creates "faults", namely, short Voronoi edges which are misaligned with the direction of the approximated curves.

In order to eliminate faults, we compute all Delaunay circles for a quasi-regular band of Delaunay cells. Short Voronoi edges correspond to circles which almost coincide. We simply interpolate circle centers and radii and get a regular band of circles. New potential Delaunay vertices are computed as intersections points of two adjacent circles. The circle interpolation procedure should take into account constrained vertices near sharp corners. We repeat this procedure for all layers until the outer boundary is reached. This procedure proved to be quite stable, provided the initial quasi-regular Delaunay mesh is built by structured refinement of a coarse regular Delaunay band. A detailed view of refined hybrid meshes is shown in Figs. 11, 12, and 13.

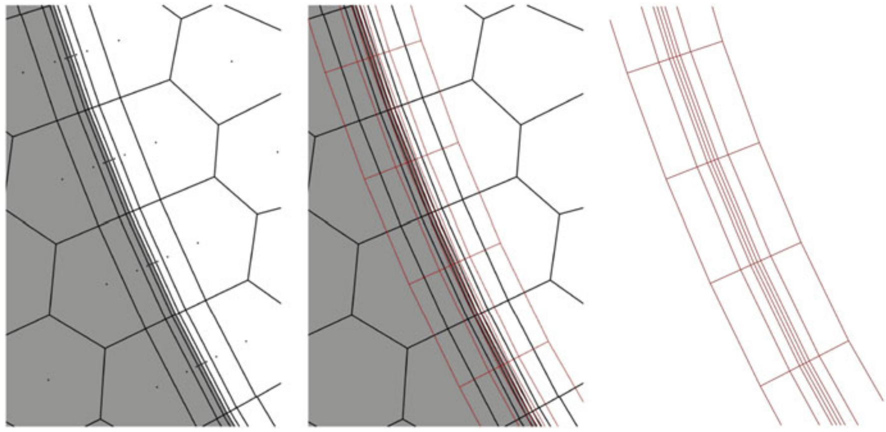


Fig. 11 (Left) Anisotropic Voronoi mesh layer, (right) dual anisotropic Delaunay layer, and (center) overlapped meshes

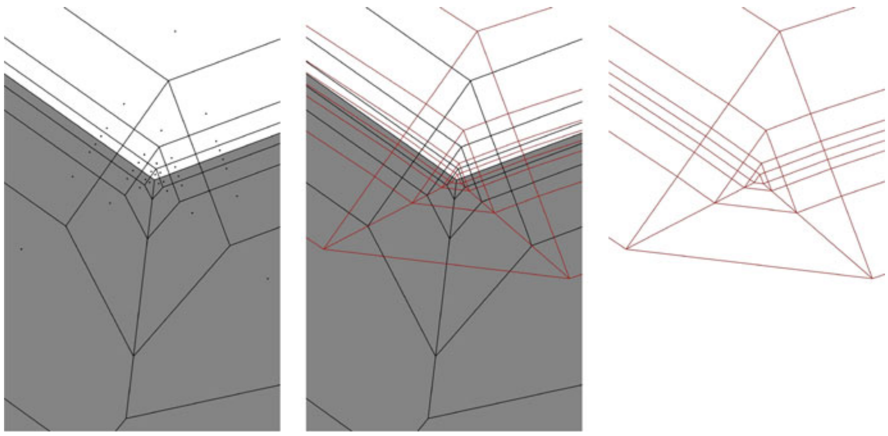


Fig. 12 Refinement scheme for corner Delaunay pentagon: (left) anisotropic Voronoi mesh layer, (right) dual anisotropic Delaunay layer, and (center) overlapped meshes

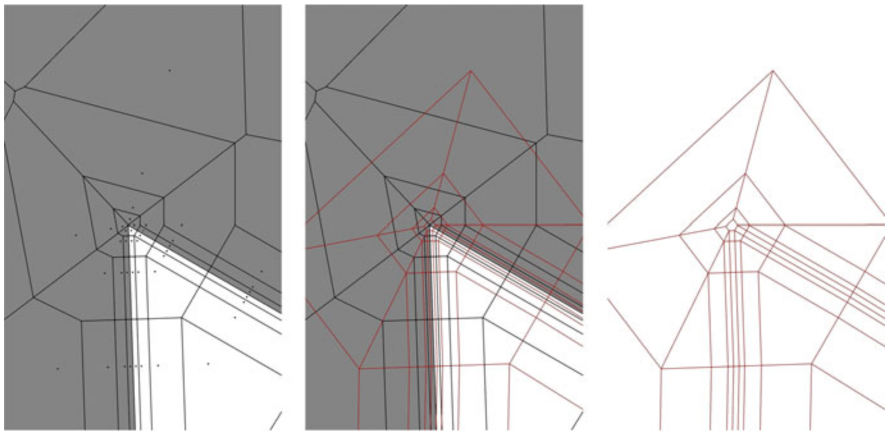


Fig. 13 Refinement scheme for corner Delaunay triangle: (left) anisotropic Voronoi mesh layer, (right) dual anisotropic Delaunay layer, (center) overlapped meshes

As one can see, the algorithm can produce Voronoi layers consisting of almost orthogonal highly anisotropic quadrilaterals.

Figure 14 shows the large fragment of hybrid Voronoi mesh before and after fault elimination.

It should be noted that the refinement schemes for corner Delaunay pentagons shown in Fig. 12, and those for corner Delaunay triangles in Fig. 13 produce Voronoi bumps around corners. We were not able to find anisotropic Delaunay/Voronoi refinement schemes which do not create such bumps. Note that the problem appears only for Voronoi cells, while Delaunay edges round off sharp corner in a nice and smooth manner. Figures 15 and 16 illustrate the behaviour of refinement schemes which are isotropic near corners.

Such refinement schemes for a Delaunay triangle, quadrilateral, and pentagon are very close in spirit to the refinement of the block-structured meshes when the blocks are glued together using the well-known C-type and H-type meshes. Here, the Voronoi faults are not eliminated in order to clarify the amount of deviation from structured mesh introduced by refinement schemes. Note that in order to get the final mesh one should add a smooth transition from each corner to the core Voronoi mesh.

4 Discussion

To our knowledge, we present the first algorithm for construction of hybrid planar Voronoi meshes which demonstrates the ability to build orthogonal layers of Voronoi cells near internal boundaries with correct resolution of sharp vertices. In order to make suggested algorithm more universal, the following problems should be addressed: nonuniform and curvature-sensitive meshing test cases should be

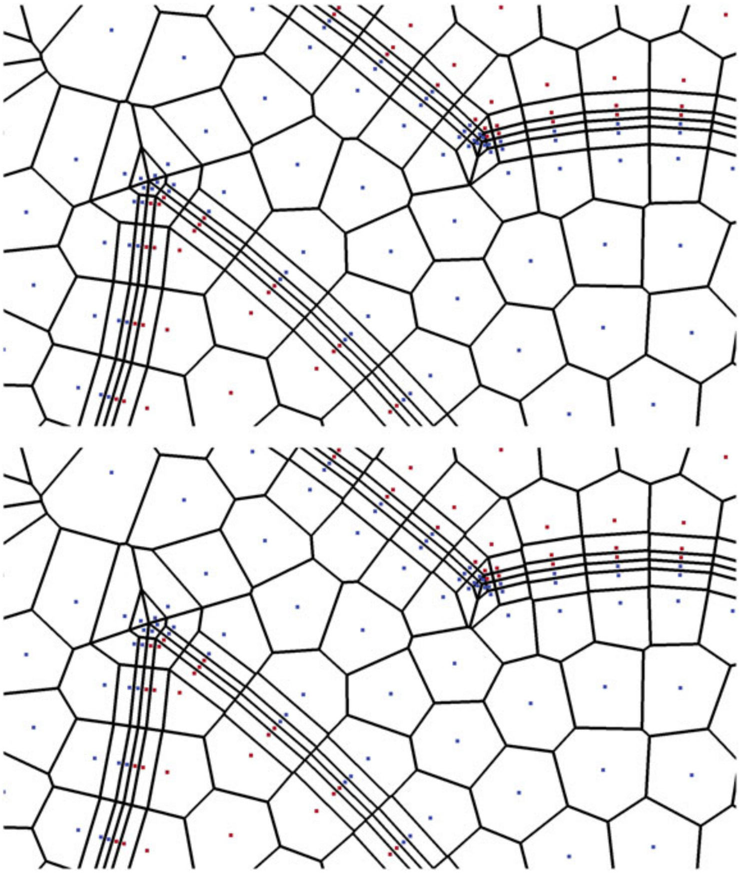


Fig. 14 Voronoi mesh after refinement before and after cleaning

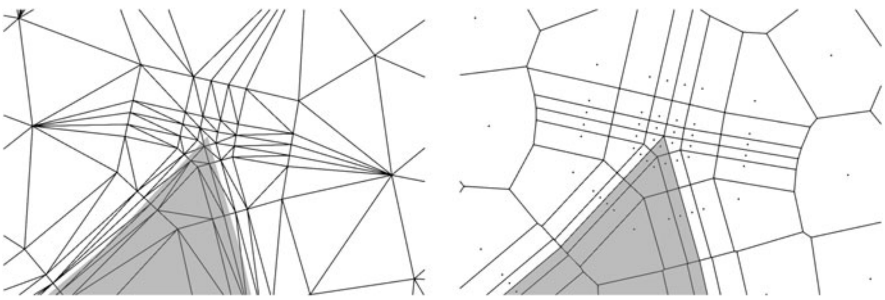


Fig. 15 (Left) Delaunay mesh after refinement, Voronoi fault cleaning is not applied; (right) Voronoi mesh

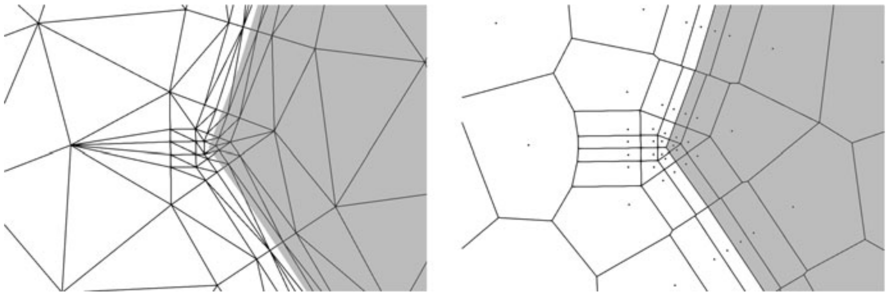


Fig. 16 (Left) Delaunay mesh after refinement, Voronoi fault cleaning is not applied; (right) Voronoi mesh

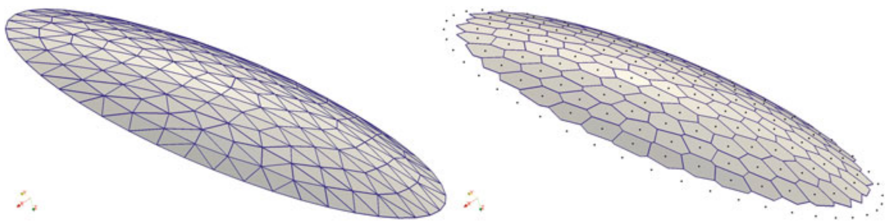


Fig. 17 Primal and dual polyhedral approximations for a convex surface

considered, the case of multimaterial vertices and thin material layers should be addressed, and, most important, the generalization to the 3d case has to be investigated. From the theoretical and practical point of view, Voronoi meshing in 3d is an unsolved problem. It is tightly related to the problem of polyhedral approximation of piecewise smooth surfaces. A number of approximation algorithms was suggested which can be referred to as "dual" methods since they attempt to approximate surface by faces which are dual either to certain vertices or to edges. Dual methods are well suited for approximation of piecewise smooth surfaces. Among those methods one can refer to the class of "dual contouring" algorithms [10] and to the primal-dual surface mesh optimization with sharpening [11]. The dual faces in these algorithms can be non-planar. Algorithm suggested in [12] can approximate quite general surfaces by polyhedral surfaces with flat faces. These faces are nonconvex in general. In [13], primal and dual polyhedral approximations were combined to obtain the discrete counterpart of a spherical mapping and its gradient. It was shown that behaviour of dual cells for the surface of positive Gaussian curvature ($K > 0$) and negative Gaussian curvature ($K < 0$) is intrinsically different. Consider a surface triangulation with vertices lying on the regular surface with strictly positive curvature. It is assumed that the triangulation is regular in a sense that it defines a strictly convex polyhedral surface, as shown in Fig. 17 (left).

The dual surface is defined by tangent planes to the surface at the vertices of triangulation. Each dual face approximates affine image of 2d Voronoi cell for projections of a local set triangulations vertices onto the tangent plane [13]. One can model 3d Voronoi mesh by replacing each vertex of the surface triangulation by a close pair of mirror vertices (positive and negative seeds), such that the middle plane orthogonal to Delaunay edge connecting the vertices is exactly the tangent plane. Constructing Voronoi faces dual to the pairs of seeds from different families, one obtains polyhedral Voronoi approximation of the surface. The Voronoi surface tends to contain faults which are more pronounced for surfaces with anisotropic curvatures, as shown in Fig. 18. Note that the dual faces can be quite anisotropic while Voronoi faces tend to be isotropic and require the faults to fit together.

An even more complicated case is related to saddle surfaces ($K < 0$). In [13], it was shown that, in this case, any regular dual face a is quadrilateral domain with polygonal concave edges, as shown in Fig. 19. Similar results were obtained in [14].

In this case, the Voronoi faults are smaller, since the anisotropy is reduced to make illustration more clear. Still these faults cannot be eliminated by simply gluing together close Delaunay balls, since topology of Voronoi and dual surface is inherently different, as shown in Fig. 20 (right).

For saddle surfaces, the only possibility for the dual faces to become convex is to become quadrilaterals with straight edges, which requires very special vertex arrangements on the surfaces. Figure 21 (left) shows a dual mesh which is planar quadrilateral one (a pq-mesh, see [15]).

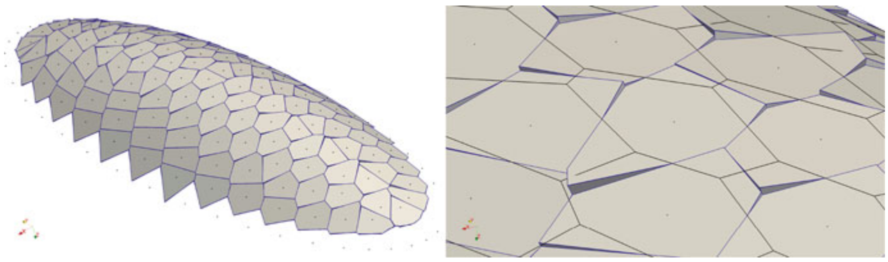


Fig. 18 Voronoi surface approximation and a fragment of overlapped dual and Voronoi polyhedral surfaces

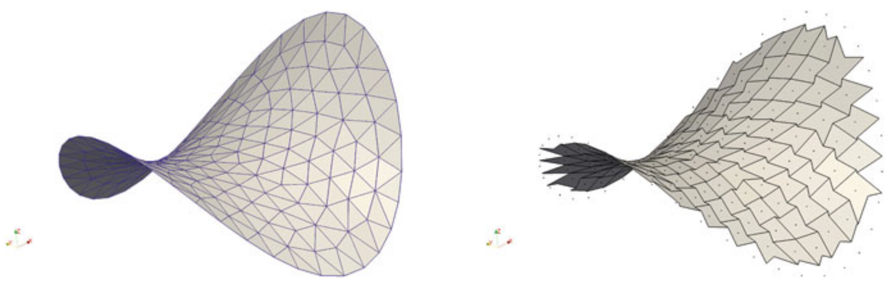


Fig. 19 Primal and dual polyhedral approximations for a saddle surface

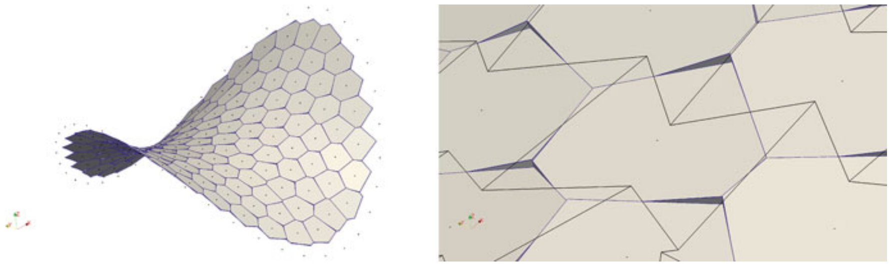


Fig. 20 Voronoi approximation of a saddle surface and a fragment of overlapped dual and Voronoi polyhedral surfaces

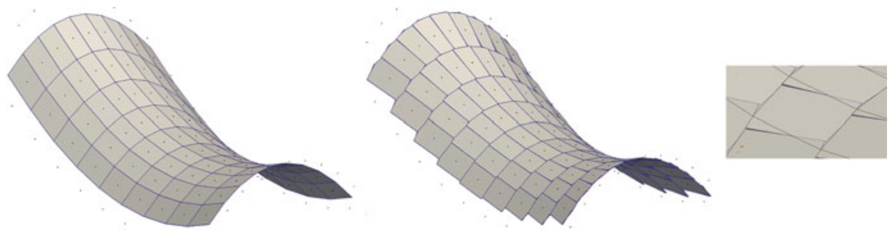


Fig. 21 Planar quadrilateral dual mesh, Voronoi surface, and superimposed view

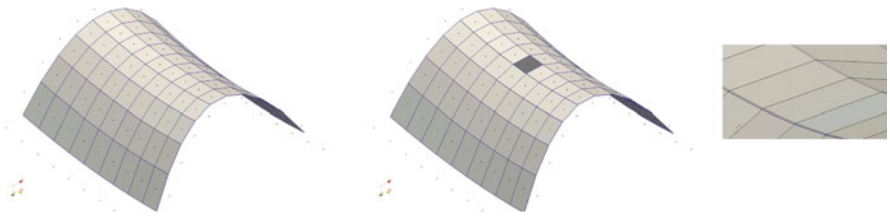


Fig. 22 Curvature line-based dual mesh, Voronoi mesh, and its superposition with Voronoi surface

The primal mesh is also a pq-mesh but its quadrilaterals are quite skewed hence Voronoi surface (Fig. 21 (center)) essentially differs from dual surface and faults cannot be eliminated but Delaunay ball gluing. Creation of new seed arrangement new boundary is necessary. It seems that one has a chance to build fault-free Voronoi surface when primal approximation is based on the circular pq-cells [15], when each flat primary cell lies on the equatorial section of certain empty sphere. Approximations to circular pq-meshes can be constructed using network of lines of curvature of the surfaces for primal vertices. Figure 22 shows dual surface, which consists of convex quadrilaterals.

When seed vertices follow the curvature lines, the Voronoi faces are almost optimal in a sense that faults are absent or quite small, as shown in Fig. 22, and can be easily suppressed via gluing close Delaunay balls.

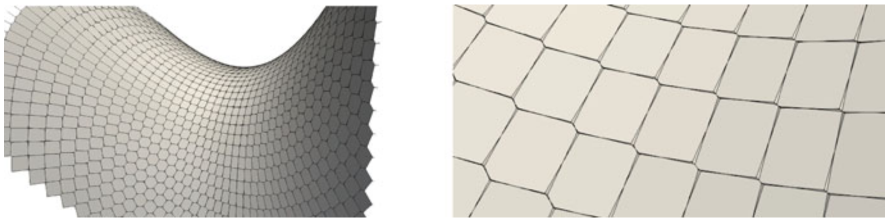


Fig. 23 The far view of a Voronoi surface looks nice, whereas a close view reveals faults

As it can be guessed, the far view of the Voronoi surface looks quite nice and only a close examination allows the identification of faults (Fig. 23).

One can formulate the following basic principles which may describe generalization of Voronoi meshing algorithm for 3d domains with piecewise-smooth boundary. The treatment of sharp edges in 3d is fairly straightforward: one can create auxiliary cylindrical tubes around each sharp edges which would be responsible for creation of almost Delaunay polygonal prisms. One should apply special treatment and create balls around conical vertices and around transitions points on the sharp edges where the switch from one type of prisms to another should be applied. One can build an auxiliary network of the curvature lines. The curvature line network, the set of curved tubes around feature lines coupled through balls around feature vertices constitute the input for the meshing algorithm itself. Unfortunately, since feature lines are not supposed to be aligned with curvature lines, for a moment it is not clear how to construct fault-free Voronoi meshes approximating feature lines.

Alternatively, one can consider a generalized polyhedral mesh and admit curved faces on and near the boundary which makes elimination of faults a routine operation. It means that our cells near boundary are Voronoi cells in certain isotropic curvilinear coordinates which looks like a reasonable solution for the problem of hybrid Voronoi meshing and requires numerical experiments with flow solvers to evaluate performance on such meshes.

Acknowledgement Supported by the Russian Foundation for Basic Research (RFBR) under grant 18-01-00726 A.

References

1. Garimella, R., Kim, J., Berndt M.: Polyhedral mesh generation and optimization for non-manifold domains. In: Proceedings of the 22nd International Meshing Roundtable, pp. 313–330. Springer, Cham (2014)
2. Lee, S.Y.: Polyhedral mesh generation and a treatise on concave geometrical edges. Proc. Eng. **124**, 174–186 (2015)
3. Amenta, N., Bern, M., Kamvysselis, M.: A new Voronoi-based surface reconstruction algorithm. In: Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques, pp. 415–421. ACM, New York (1998)

4. Liu, Y., Wang, W., Levy, B., Sun, F., Yan, D.M., Lu, L., Yang C.: On centroidal Voronoi tessellation – energy smoothness and fast computation. ACM Trans. Graph. **28**(4), 101 (2009)
5. Budninskiy, M., Liu, B., De Goes, F., Tong, Y., Alliez, P., Desbrun M.: Optimal voronoi tessellations with hessian-based anisotropy. ACM Trans. Graph. **35**(6), Article No. 242 (2016)
6. Tournois, J., Alliez, P., Devillers, O.: 2D centroidal voronoi tessellations with constraints. Numer. Math. Theory Methods Appl. **3**(2), 212–222 (2010)
7. Levy, B., Liu, Y.: Lp centroidal voronoi tessellation and its applications. ACM Trans. Graph. **29**(4), Art. 119 (2010)
8. Garanzha, V.A., Kudryavtseva, L.N.: Generation of three-dimensional Delaunay meshes from weakly structured and inconsistent data. Comput. Math. Math. Phys. **52**(3), 427–447 (2012)
9. Persson, P.-O., Strang, G.: A simple mesh generator in MATLAB. SIAM Rev. **46**(2), 329–345 (2004)
10. Ju, T., Losasso, F., Schaefer, S., Warren, J.: Dual contouring of hermite data. ACM Trans. Graph. **21**(3), 339–346 (2002)
11. Ohtake, Y., Belyaev, A.: Dual/primal mesh optimization for polygonized implicit surfaces. In: Proceedings of the 7th ACM Symposium on Solid Modeling and Applications (SMA02́), pp. 171–178. ACM, New York (2002)
12. Cohen-Steiner, D., Alliez, P., Desbrun, M.: Variational shape approximation. ACM Trans. Graph. **23**(3), 905–914 (2004)
13. Garanzha, V.A.: Discrete extrinsic curvatures and approximation of surfaces by polar polyhedra. Comput. Math. Math. Phys. **50**(1), 65–92 (2010)
14. Gunther, F., Jiang, C., Pottmann, H.: Smooth polyhedral surfaces. arXiv:1703.05318 [math.MG] (2017)
15. Liu, Y., Pottmann, H., Wallner, J., Yang, Y.L., Wang, W.: Geometric modeling with conical meshes and developable surfaces. ACM Trans. Graph. **25**(3), 681–689 (2006)