Vladimir A. Garanzha*, Lyudmila N. Kudryavtseva, and Aleksei I. Belokrys-Fedotov

Single and multiple springback technique for construction and control of thick prismatic mesh layers

https://doi.org/10.1515/rnam-2021-00..
Received June 29, 2020; revised October 15, 2020; accepted ..., 2020

Abstract: We suggest an algorithm for construction of semi-structured thick prismatic mesh layers which guarantees absence of inverted prismatic cells in resulting layer and allows one to control near-surface mesh orthogonality. Initial mesh is modelled as a thin layer of highly compressed prisms made of hyperelastic material glued to the triangulated surface. In order to compute robust normals at the vertices of the surface mesh we use quadratic programming algorithm based on the nearest ball concept. This pre-stressed material expands, possibly with self-penetration and extrusion to exterior of computational domain until target layer thickness is attained.

Special preconditioned relaxation procedure is proposed based on the solution of stationary springback problem. It is shown that preconditioner can handle very stiff problems related to construction of very thick one-cell-wide layers for rather fine surface meshes. Once an offset prismatic mesh is constructed self-intersections are then eliminated using iterative prism cutting procedure. Next, variational advancing front procedure is applied for refinement and precise orthogonalization of prismatic layer near boundaries. It is guaranteed that the resulting mesh is free from inverted prisms. We demonstrate that resulting mesh layer is 'almost mesh-independent' in a sense that the dependence of thickness and shape of the layer on mesh resolution and triangle quality is weak.

It is possible to apply elastic springback technique sequentially layer by layer. We compare single springback technique with multiple springback technique in terms of mesh quality, stiffness of local variational problems and mesh orthogonality/layer thickness balance.

Keywords: Prismatic mesh layer, variational method, hyperelastic deformation, mesh untangling and optimization.

MSC 2010: ???

High quality simulation of viscous flows imposes rather strict requirements on computational meshes near solid boundaries. It it very important to construct meshes which provide orthogonality near boundary and precise control over mesh element size in the direction orthogonal to boundary irrespectively of the size and shape of surface mesh elements. Variational methods make this precise control possible [3]. Prismatic mesh layers consisting of triangular prisms, hexahedra or general polygonal prisms are flexible enough to be incorporated into automatic mesh generators while providing high quality mesh near boundaries. We consider semi-structured layers with the same mesh connectivity on each sublayer. In the literature, sometimes more general case is considered where topology changes are admitted for mesh quality improvement [15]. However, we do not consider this case. Prismatic mesh layer is considered to be 'thick' when its transverse size is comparable to the characteristic size of the geometric model. One can also call prismatic layer thick when

***Corresponding author: Vladimir A. Garanzha,** Dorodnicyn Computing Center FRC CSC RAS, Moscow 119333, Russia; Moscow Institute of Physics and Technology, Dolgoprudny 141701, Moscow Region, Russia. E-mail: garan@ccas.ru
Lyudmila N. Kudryavtseva, Dorodnicyn Computing Center FRC CSC RAS, Moscow 119333, Russia; Moscow Institute of Physics and Technology, Dolgoprudny 141701, Moscow Region, Russia
Aleksei I. Belokrys-Fedotov, Dorodnicyn Computing Center FRC CSC RAS, Moscow 119333, Russia

its height is considerably larger compared to mesh element size on the surface. We present an algorithm for construction of thick prismatic layers which is the generalization of results from [8].

1 Variational principle for construction of prismatic layers

Let ξ_1, ξ_2, ξ_3 denote the Lagrangian coordinates associated with elastic material, and x_1, x_2, x_3 denote the Eulerian coordinates of a material point. Spatial mapping $x(\xi) : \mathbb{R}^3 \to \mathbb{R}^3$ defines a stationary elastic deformation. The Jacobian matrix of the mapping $x(\xi)$ is denoted by C, where $c_{ij} = \partial x_i/\partial \xi_j$.

We look for the elastic deformation $x(\xi)$ that minimizes the following weighted stored energy functional [7]:

$$F(x) = \int_{\Omega_\xi} w(\xi) W(C)\, \mathrm{d}\xi \tag{1.1}$$

where $W(C)$ is polyconvex elastic potential (internal energy) which is a weighted sum of shape distortion measure and volume distortion measure [6]:

$$W(C) = (1-\vartheta)\frac{\left(\frac{1}{3}\operatorname{tr}(C^T C)\right)^{3/2}}{\det C} + \frac{1}{2}\vartheta\left(\frac{1}{\det C} + \det C\right). \tag{1.2}$$

In most cases we set $\vartheta = 4/5$. However for one- or two- cell-wide mesh layers with fixed boundary influence of volumetric distortion term is not pronounced hence it is safe to use the value $\vartheta = 0$. Post-optimization of multilayered meshes with $\vartheta = 0$ is more likely to result in too large or too small cells hence volumetric term becomes useful. Here Ω_ξ denotes parametric domain with fixed mesh.

Since distortion measure (1.2) is minimized on the average, locally it can be quite large. In theory it can be infinite on the set of zero measure. In practice it means that with mesh refinement quality of mesh cell can locally deteriorate.

In practice, one can control the spatial distribution of distortion measure without actual contraction of the set of feasible mappings. Experience suggests that large values of distortion appear near boundaries and surfaces of material discontinuity Hence it is possible to introduce a weight function $w(\cdot)$ in the Lagrangian or Eulerian coordinates which takes large values in critical regions and is close to unity elsewhere.

In the process of minimization, elements with a larger weight tend to have a smaller value of distortion function $W(C)$. Hence, their shapes and sizes are very close to the target ones. This simple approach proved to be very efficient for mesh orthogonalization near the boundary [6]. A proper choice of the weight allows us to satisfy the no-slip boundary conditions and to approximate boundary orthogonality conditions and prescribed mesh element size in the normal direction very accurately.

Theoretical arguments suggest that in order to eliminate the local singularities of the distortion function the weight distribution should be singular. However, this singularity is only reached in the limit of mesh refinement and for any given finite mesh weight distribution is bounded. One cannot prove that resulting deformation is quasi-isometric as in [6, 7] but numerical evidences suggest independence of the global mesh distortion bounds from the mesh size.

We have also found that in the case when the locations of highly distorted fragments are not known *a priori*, one can compute the weight adaptively using special iterative scheme when weight distribution for current iteration is based on the computed distortion measure from previous iterations.

Suppose that domain Ω_ξ can be partitioned into convex polyhedra U_k. Then stored energy functional (1.1) can be approximated by the following semi-discrete functional:

$$F(x_h(\xi)) = \sum_k \int_{U_k} w(\xi) W(\nabla x_h(\xi))\, \mathrm{d}\xi \tag{1.3}$$

where $x_h(\xi)$ is continuous piecewise-smooth deformation.

In order to approximate integral over a convex cell U_k one should use certain quadrature rules. As a result semi-discrete functional (1.3) is replaced by the discrete functional:

$$F(x_h(\xi)) \approx \sum_k \mathrm{vol}(U_k) \sum_{q=1}^{N_k} \beta_q w_q W(C_q) = F^h(x_h(\xi))$$

where N_k is the number of quadrature nodes per cell U_k, C_q denotes the Jacobian matrix in qth quadrature node of U_k, while β_q are the quadrature weights and w_q are the values of weight function in the quadrature nodes.

The following majorization property should hold

$$F(x_h(\xi)) \leqslant F^h(x_h(\xi)). \tag{1.4}$$

This property can be used to prove that all intermediate deformations $x_h(\xi)$ providing finite values of discrete functional are homeomorphisms [7].

Let $G_\xi(\xi)$ and $G_x(x)$ denote the metric tensors defining linear elements and length of curves in Lagrangian and Eulerian coordinates in the domains Ω_ξ and Ω_x, respectively. Then, $x(\xi)$ is the mapping between metric manifolds M_ξ and M_x. The distortion functional (1.1) for this mapping can be written as

$$F(x) = \int_{\Omega_\xi} w(\xi) W(Q\nabla_\xi x H^{-1}) \det H \, \mathrm{d}\xi \tag{1.5}$$

where

$$H^T H = G_\xi, \quad \det H > 0, \qquad Q^T Q = G_x, \quad \det Q > 0$$

are arbitrary matrix factorizations of metric tensors G_ξ and G_x.

The corresponding discrete functional can be written as follows

$$F^h(x_h(\xi)) = \sum_k \mathrm{vol}(U_k) \sum_{q=1}^{N_k} w_q \beta_q W(Q_q C_q H_q^{-1}) \det H_q.$$

Note, that in the presence of the control metrics exact majorization inequality can be violated and one should be careful with quadrature rules in order to guarantee certain relaxed formulation for majorization, say in the form

$$F(x_h(\xi)) \leqslant K F^h(x_h(\xi)) \tag{1.6}$$

where K is a constant. This inequality should guarantee that every intermediate iteration of the mesh generation method has finite energy for mapping as a whole and not just for the finite set of quadrature nodes.

2 Single springback algorithm for construction of prismatic layers

Suppose that a thin layer of hyperelastic material is glued to the surface of the body. This material is highly compressed in the direction orthogonal to the surface. Now suppose that the surface of the layer opposite to the domain boundary is freed which results in classical springback problem for pre-stressed hyperelastic material. Static springback deformation can be found as a result of minimization of stored energy.

Elastic material is modelled by the one-cell-wide layer $\mathbb{P}$ of triangular prisms. For each prism $P \in \mathbb{P}$ the target prism P_t from the known prismatic layer in certain parametric manifold is specified. In order to construct such a target prism we consider a triangle T which belongs to the oriented surface triangulation of the polyhedral surface S. We map this triangle isometrically onto the plane $x_3 = 0$ in such a way that its normal is directed upwards and build on it rectangular triangular prism with the height $H(T)$ equal to the prescribed thickness of the layer. Consider piecewise-smooth deformation $x_h : \mathbb{P}_t \to \mathbb{P}$ as a solution of minimization problem for (1.3) with free boundary. When equilibrium solution is attained the thickness of

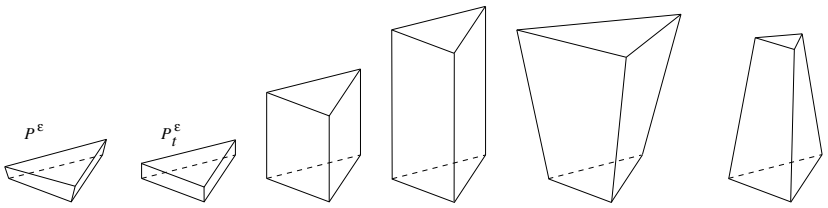


Fig. 1: (a) Initial thin prism with $h = \varepsilon$, (b)–(d) height enlargement for target prisms, (e)–(f) real prisms after springback.

elastic material would approximate the prescribed one. Note that at the springback relaxation stage material self-contact is ignored hence global overlaps are allowed.

We consider auxiliary prism P^ε constructed on the same triangular base T with the vertices p_i and $p_i+\varepsilon\nu_i$, $i = 1, \dots, 3$. Here p_i are the vertices of T, ε is a certain small constant and ν_i is the discrete unit normal to polyhedral surface S at the vertex p_i. Elastic deformation $x_h : \mathbb{P}_t \to \mathbb{P}^\varepsilon$ is quite far from isometry since $H(T)$ generally is much larger compared to ε. The tensions inside elastic material would move free surface away from the body.

Springback computation under strong compression is rather difficult. Hence we use the set of successive target states in order to relax the stiffness of the problem. The sequence of target prisms defining deformations $x_h : \mathbb{P}_t^h \to \mathbb{P}^h$ is constructed via gradual enlargement of the target height h from ε to $H(T)$.

Initial height ε is chosen in such a way that prismatic layer $\mathbb{P}^\varepsilon$ is admissible. If for a small ε the layer still contains inverted prisms then preliminary untangling problem is solved using technique from [7].

Figure 1 illustrates the springback technique: (a) thin initial prism P^ε, (b) initial target prism P_t^ε, (c)–(d) the target prism growth, while (e)–(f) show how real equilibrium prisms near convex and concave surface fragments, respectively, look like. Note that the attainable prism thickness has a variable value. For example, an attempt to build thick layer inside a sphere may lead to real thickness which is much smaller compared to target one since quite strong compression of the upper part of thick prism prevents further growth of the layer.

One can reformulate the above procedure in algebraic terms. For each prism we specify metric tensor G_ξ in Lagrangian coordinates in such a way that after minimization of (1.5) thin layer is obtained. Since target shapes are orthogonal we set $(G_\xi)_{13} = (G_\xi)_{23} = 0$. Elements $(G_\xi)_{ij}$, $i, j = 1, 2$, are fixed while element $(G_\xi)_{33}$ is gradually enlarged from ε^2 to $H^2(T)$. After each enlargement step variational problem (1.5) is solved approximately [8]. For the largest value of $(G_\xi)_{33}$ more minimization iterations are used.

3 Choice of the quadrature rules

In order to approximate integral (1.3) over the cell U_k one should use quadrature rules. In each prism elastic deformation is approximated by the bilinear mapping

$$x(\xi) = (p_0(1-\xi_1-\xi_2) + p_1\xi_1 + p_2\xi_2)(1-\xi_3) + (p_3(1-\xi_1-\xi_2) + p_4\xi_1 + p_5\xi_2)\xi_3. \tag{3.1}$$

Note that this function maps rectangular prism with half of unit square as a base onto triangular prism with 3D vertices $p_0 - p_5$. The numbering scheme for vertices is shown in Fig. 2. In order to build mapping of target prism onto current cell one has to use composition of mappings $x(\xi) \circ \eta(\xi)^{-1}$, where function $\eta(\xi)$ is similar to (3.1).

The columns of the Jacobian matrix of mapping (3.1) can be written as

$$\frac{\partial x(\xi)}{\partial \xi_1} = (p_1 - p_0)(1-\xi_3) + (p_4 - p_3)\xi_3, \qquad \frac{\partial x(\xi)}{\partial \xi_2} = (p_2 - p_0)(1-\xi_3) + (p_5 - p_3)\xi_3$$
$$\frac{\partial x(\xi)}{\partial \xi_3} = (p_3 - p_0)(1-\xi_1-\xi_2) + (p_4 - p_1)\xi_1 + (p_5 - p_2)\xi_2.$$

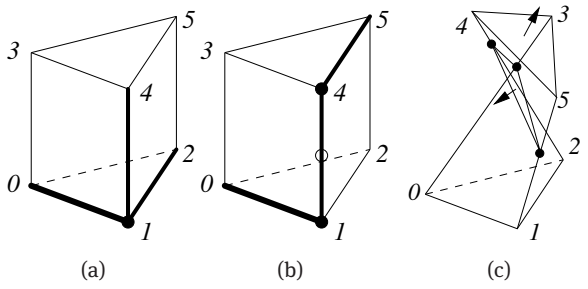


Fig. 2: (a) Quadrature node at the vertex, (b) quadrature node at the edge.

Hence the Jacobian matrix admits representation

$$\nabla_\xi x = C_1\Lambda_1(\xi) + C_2\Lambda_2(\xi) + C_3\Lambda_3(\xi), \quad \sum_{j=1}^{3}\Lambda_j = I, \quad \Lambda_j \geqslant 0$$

$$\Lambda_1 = \mathrm{diag}(1-\xi_3,\ 1-\xi_3,\ \xi_1), \qquad C_1 = (p_1 - p_0\ \ p_2 - p_0\ \ p_4 - p_1)$$
$$\Lambda_2 = \mathrm{diag}(\xi_3,\ \xi_3,\ \xi_2), \qquad C_2 = (p_4 - p_3\ \ p_5 - p_3\ \ p_5 - p_2)$$
$$\Lambda_3 = \mathrm{diag}(0,\ 0,\ 1-\xi_1-\xi_2), \qquad C_3 = (p_1 - p_0\ \ p_2 - p_0\ \ p_3 - p_0).$$

Thus the majorization principle for polyconvex distortion measures [1, 7] can be applied which means that for polyconvex function (1.2):

$$W(\nabla_\xi x) \leqslant \sum_{\nu=1}^{12} a_\nu(\xi) W(\tilde{C}_\nu)$$

where $\tilde{C}_\nu$ $\nu = 1, \ldots, 12$ means 3×3 'compound matrix' where kth column is arbitrary chosen as a kth columns of any of the basis matrices C_i, $i = 1, \ldots, 3$.

This inequality provides natural geometric quadratures for construction of discrete distortion measure for prism. To this end one has to consider all compound matrices $\tilde{C}_\nu$ generated by the basis matrices C_1, C_2, and C_3. Note that total number of quadrature nodes is not equal to $3^3 = 27$ as it was suggested in theorem from [1, 7]. Precise number is equal to $2 \times 2 \times 3 = 12$ since first two columns of the matrix Λ_3 are equal to zero. The presence of additional mapping $\eta(\xi)$ does not prevent using theorem from [1, 7] since mapping (3.1) is affine for orthogonal prisms.

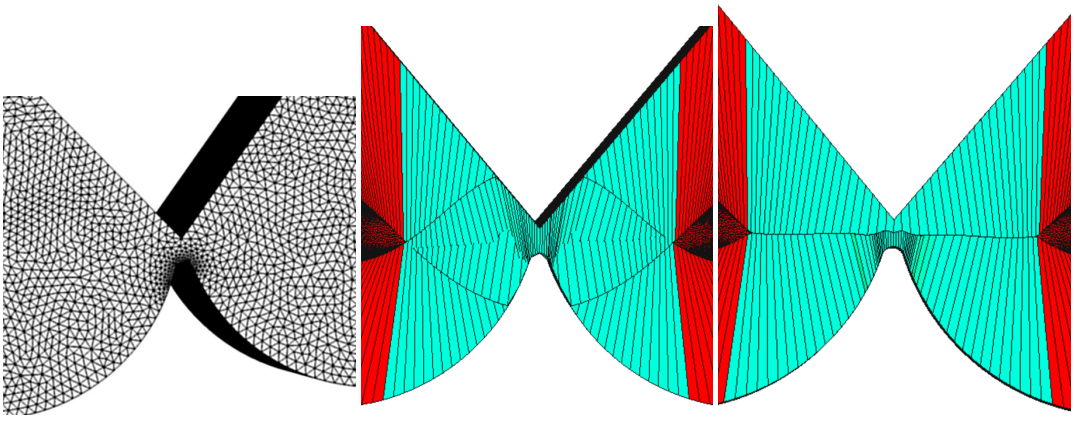
The set of quadrature nodes can be split into two groups each consisting of 6 elements.

Compound matrix where both the first and the second columns are simultaneously taken from either matrix C_1 or from matrix C_2 while the third column is arbitrary taken from the third column of C_i corresponds to vertex-based quadrature node (see Fig. 2a). When first two columns of $\tilde{C}_\nu$ are chosen from different matrices C_1 and C_2 it corresponds to vertical edge-based quadrature node (see Fig. 2b).

To summarize, one should use 6 vertex-based nodes and 6 vertical edge-based nodes in order to discretize functional (1.3).

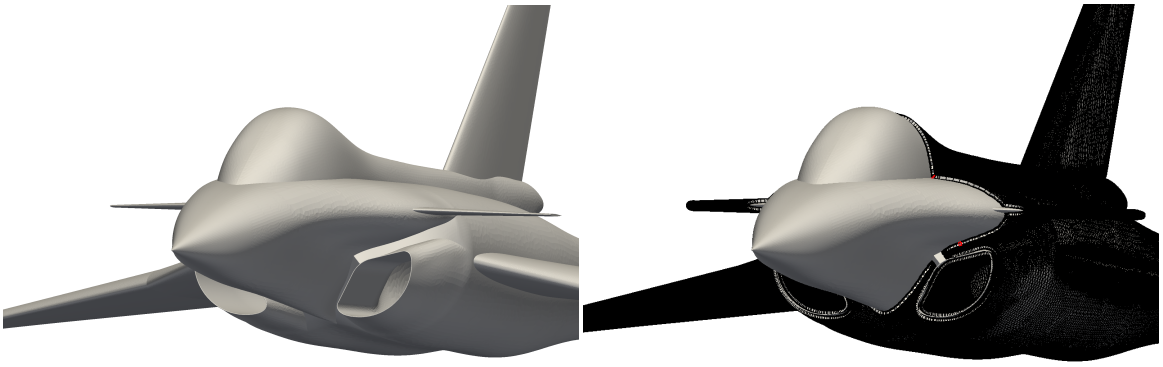
4 Elimination of self-intersections

We do not take into account self-contact of elastic material during springback. Instead we allow material to self-penetrate and freely intersect the boundary of domain. In order to eliminate self-intersection iterative cutting procedure is applied, similar to [9]. First we cut off material on each prism which goes out of the computational domain. After that we build the list of all prisms which intersect other prisms and apply to the prisms in the list thickness reduction by certain relative coefficient β slightly less then unity. This procedure is repeated until all intersections are eliminated. As a result in the self-penetration zones certain 'contact surface' is constructed. In the limiting case of opposite parallel planes at the distance D and with thicknesses



(a) (b) (c)

Fig. 3: (a) Surface mesh, (b) elimination of material in the exterior of the domain, (c) construction of the contact surface.



(a) (b)

Fig. 4: (a) 'Rafale' model, (b) intermediate layer.

of overlapping opposite layers equal to H_1 and H_2, where $\delta H = H_1 + H_2 - D > 0$, this procedure will create opposite layers with thicknesses close to

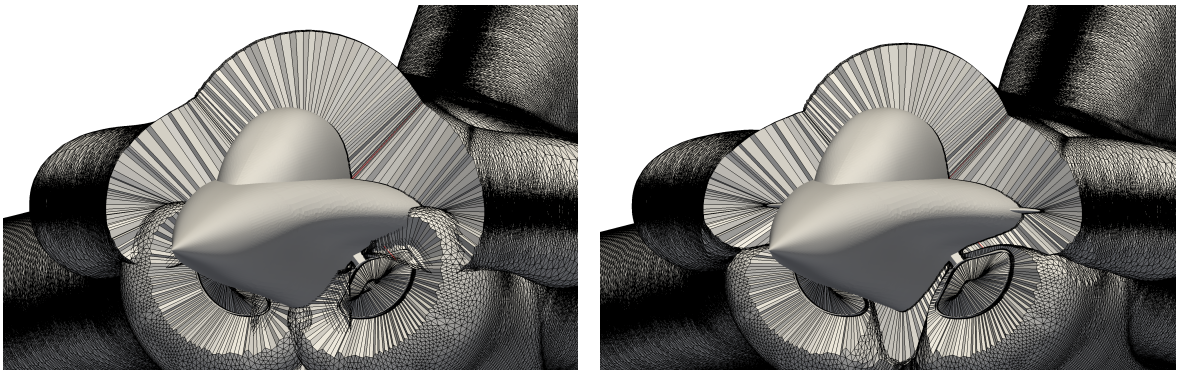
$$H_1' = \frac{H_1}{H_1 + H_2} D, \qquad H_2' = \frac{H_2}{H_1 + H_2} D$$

which means that in general the found middle surface does not correspond to the middle surface of overlap region.

Additional one-sided smoothing procedure is applied to the outer surface of the layer which can only reduce the thickness thus avoiding possibility of reappearance of overlaps. This process is illustrated in Fig. 3.

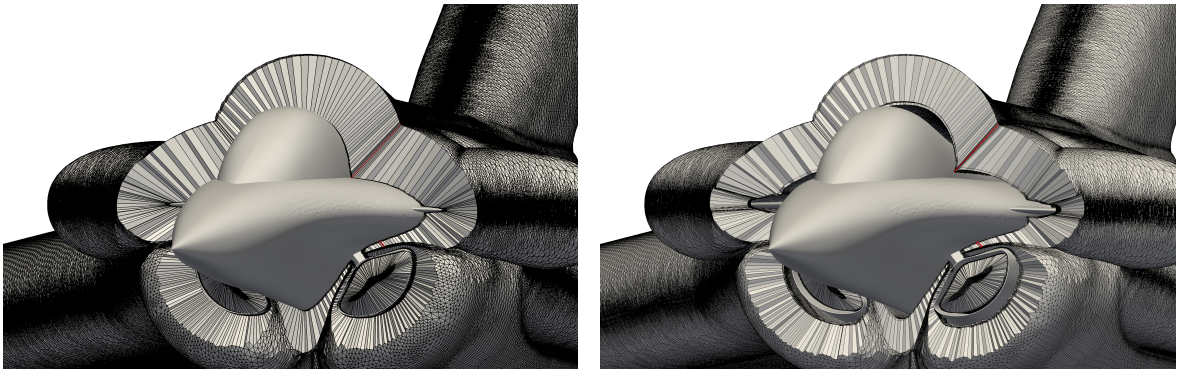
Algorithm stages are illustrated in Fig. 4–8 for Daussallt 'Rafale' model available at grabcad.com collection. Highly nonuniform surface mesh of this model contains about 600 thousand of triangles. Since initial thin layer is too thin to be visible we show intermediate relatively thin layer in Fig. 4b. Final prismatic layer is quite thick and does not contain degenerate prisms but may contain self-overlaps and boundary overlaps as shown in Fig. 5a. Elimination of self-intersection results in the mesh shown in Fig. 5b.

Additional smoothing is applied to the outer surface of layer. Its vertices are allowed to move along the transverse edges of prisms. Smoothing procedure is based on the Laplace–Beltrami smoothing iterations applied to approximate distance function computed along transverse edges of the prisms. Mean value discretization [4] is used to enforce maximum principle. Smoothed surface is shown in Fig. 6a.



(a) (b)

Fig. 5: (a) Thick layer, (b) thick layer without self-intersections.



(a) (b)

Fig. 6: (a) Layer after Laplace–Beltrami smoothing, (b) layer is split into two sublayers.

5 Layer refinement and orthogonalization

As soon as the offset surface along with the set of very long prismatic cells is constructed one has to split the layer according to prescribed mesh size distribution in the direction orthogonal to the boundary.

Figure 7a shows single prism of thick layer and imaginary mesh lines which are the images of the straight transverse edges of prism after refinement and orthogonalization. First layer is cut from the prism according to the prescribed mesh size distribution law (see Fig. 7b), then, two-cell-wide layer is optimized (see Fig. 7c). Figure 7d shows the initial guess for the next layer. Here, weight $w(\xi)$ in the lower cell is much larger than the one in the upper cell. As a result the lower cell is orthogonalized. Since upper boundary of layer is fixed this procedure eventually leads to transfer of non-orthogonality from solid boundary to outer boundary of layer. We can either dismiss certain outer fraction of layer or use the sequence of weights where difference of weights on two layers eventually diminishes and tends to unity. The ratio of weights is largest in approximately one-quarter or one-third of the total layer. One could also consider movement of vertices on the outer boundary during optimization but here we do not use such an algorithm.

Successive splitting and orthogonalization are shown in Fig. 6b and Fig. 8a. As one can see variational method is applied only to one-cell-wide or two-cell-wide layers, while total number of sublayers can be arbitrary large. Final layer orthogonal near the boundary is shown in Fig. 8b.

The above procedure guarantees the absence of degenerate elements at each stage of the prismatic layer generation algorithm.

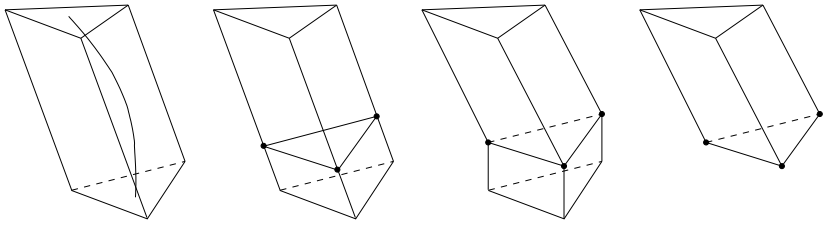


Fig. 7: Successive steps of layer splitting and orthogonalization.

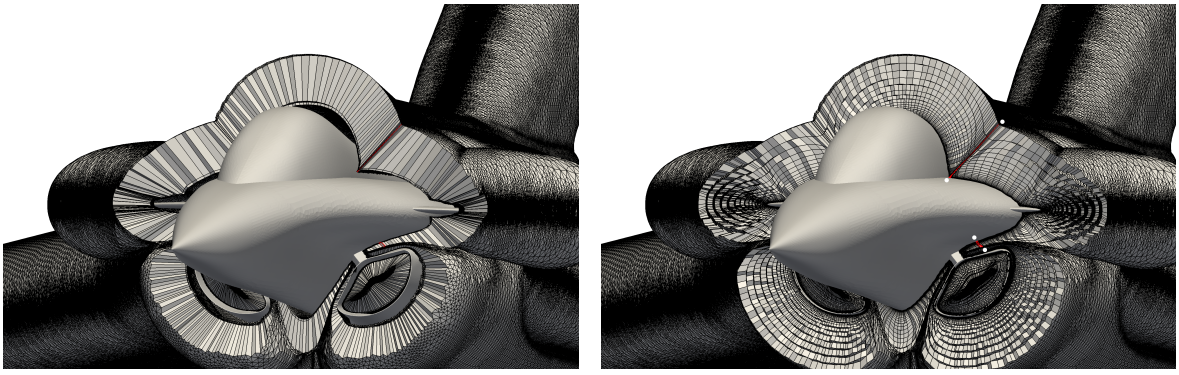


Fig. 8: (a) Orthogonalization of internal sublayer, (b) final prismatic layer.

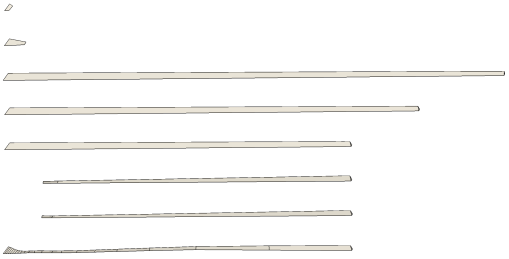


Fig. 9: Illustration of single springback algorithm for one surface triangle.

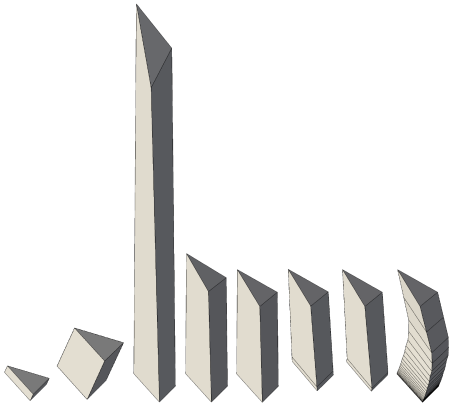


Fig. 10: Illustration of single springback algorithm for one surface triangle.

Figures 9 and 10 illustrate all steps of the above procedure for quite small surface triangle adjacent to sharp surface edge with free outer boundary (see Fig. 9), and for relatively large triangle on the regular part of surface with constrained outer boundary (see Fig. 10). The prismatic columns originating from these surface triangles in the final prismatic layer are marked by white dots in Fig. 8.

First three prisms in Figs. 9, 10 illustrate the prism grows during springback, then we show elimination of self-intersections, smoothing of outer surface, one step of splitting and orthogonalization and final sets of prisms above these triangles. As one can see, the height of prisms after springback is larger than prescribed value. We use excess ratio $E_r = 1.2$, i.e., ratio of height of target prisms for variational method to prescribed thickness of the layer. Precise value of thickness is recovered only during outer surface smoothing step, when layer thickness is first reduced by a factor E_r and then smoothing with outward-directed admissible displacements is applied thus preventing further thickness reduction. One can notice also that length of prisms near sharp edges exceeds the value that one would expect for given E_r.

In this particular example prismatic mesh contains 20 layers. After splitting very long prisms are still present. In fact for 'Rafale' model the maximum ratio of thickness to surface edge length attains $6 \cdot 10^3$ hence even splitting prismatic mesh into 100 layers will not eliminate quite long prisms. Evidently, hyperelastic springback problem for this model is quite stiff.

6 Discrete normals on sharp features of polyhedral surfaces

A number of algorithms was suggested in order to assign normal vectors to the vertices of the polyhedral surfaces.

Denote such a vertex by p, then vector n_i is the unit normal to the adjacent triangle T_i with vertices pp_ip_{i+1}. Normal vector in p is computed via

$$\nu = \frac{\sum_i w_i n_i}{\left|\sum_i w_i n_i\right|}, \; w_i > 0. \tag{6.1}$$

We use weights $w_i = \vartheta_i$, where ϑ_i is the angle of T_i at p. Note that well-known algorithms for computing vertex normals in general do not guarantee that discrete normal is really directed inward the domain, namely, that the following inequality holds

$$n_i^T \nu > 0. \tag{6.2}$$

Numerical experiments show that violations of this inequality are quite rare and are associated with sharp feature lines and conical vertices with complex neighborhood structure. Constrained algorithms for computation of discrete normals play important role for prismatic layer generation [13].

Assume that there exists a pair vertex p–adjacent triangle T with unit normal n, satisfying the following inequality

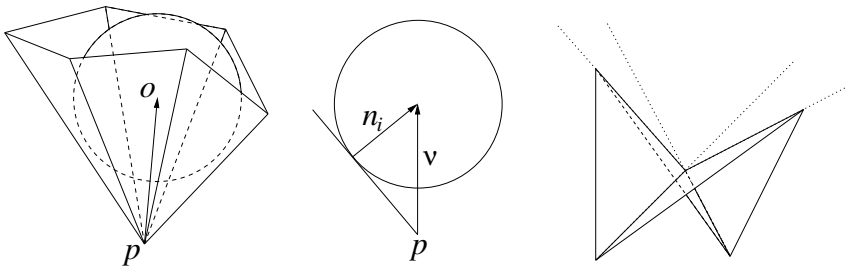
$$n^T \nu < \beta |\nu|. \tag{6.3}$$

It means that discrete normal is almost tangential to the surface. In practice we use coefficient $\beta = 1 - \cos(\pi/9)$. If such pairs are present in the mesh, then for each 'suspicious' vertex we suggest to apply the following algorithm.

Let us consider at the vertex p a convex cone K defined as the intersection of the half-spaces $(x - p)^T n_i \geqslant 0$ defined by adjacent faces. If this cone is empty then normal vector satisfying constraints (6.2) does not exist and one cannot find coordinate frame where vertex neighbourhood is presented as Lipschitz-continuous elevation function.

Let us try to place unit ball inside this cone such that the center o of the ball is as close to p as possible, as shown in Fig. 11a, while Fig. 11c shows construction of convex cone for saddle-like polyhedral surface fragment.

Denote normal vector by $\nu = o - p$. Suppose that the ball touches the plane of the adjacent face with unit normal n_i, as illustrated in Fig. 11b. Then $|\nu|^2 - |\nu - n_i n_i^T \nu|^2 = 1$, i.e., $\nu^T n_i = 1$. Hence the problem of optimal ball placement is reduced to standard quadratic programming (QP) problem.



(a) (b) (c)

Fig. 11: (a) Ball inside convex cone, (b) ball touches the face of a cone, (c) construction of convex cone for mesh vertex.

Find vector $v \in \mathbb{R}^3$ via minimization of

$$\min \frac{1}{2}|v|^2, \quad \text{s. t. } n_i^T v \geqslant 1,\ i = 1, \ldots, m \tag{6.4}$$

where m is the number of adjacent faces for vertex p.

It is geometrically evident that vector v is fully defined by 2 or 3 active faces, despite the fact that it can touch larger number of planes.

Since the number of suspicious vertices is small we use direct search method to solve QP problem instead of iterative technique. First we consider all pairs n_i, n_j, $i \neq j$, and check that the solution

$$v = \frac{n_i + n_j}{1 + n_i^T n_j}$$

satisfies all the remaining constraints (6.4). If such pair is not found, we seek a triple of distinct normals n_i, n_j, n_k being the solution of the linear system $(n_i\, n_j\, n_k)^T v = (1\,1\,1)^T$.

If no admissible solution is found then exterior penalty solution of the overdetermined QP problem is used to find the acceptable direction. Then several faces should be found which are not acceptable for this direction. The remaining faces are marked as active for variational method.

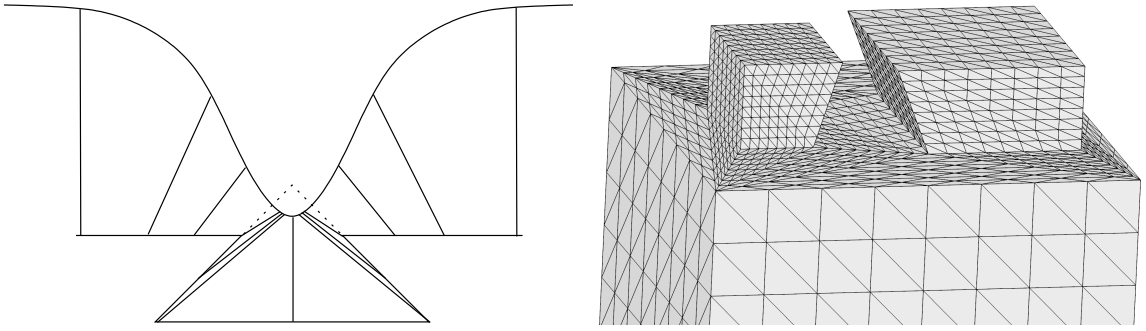
At least one of the prisms adjacent to non-Lipschitz vertex p should be degenerate and the Jacobian of mapping (3.1) should attain zero or negative values. For such a prism another approximation scheme for variational method should be used. Consider prism P adjacent to p and based on 'bad' face. We exclude from the set of 12 quadrature nodes those tetrahedra which contain at least two prism edges originating from vertex p. Remaining quadrature nodes serve to guarantee nondegeneracy of this prism as a generalized polyhedron.

7 Multiple springback technique

Numerical experiments have shown that deviation from orthogonality in resulting layers near surface is negligible, general quality of prismatic layers is quite good and it can be useful in industrial application. On the other hand variational method is time consuming if applied globally. It seems to be at least five times slower compared to the state-of-the-art industrial prismatic mesh generators.

In many cases there is no need to construct offset surfaces and one-cell-wide thick prismatic layers. Hence the above described technique while being very powerful becomes too complex and cumbersome. It has obvious drawbacks when 'locking' of transverse directions leads to sharp layer thickness decrease as shown in Fig. 12a. Another drawback is related to construction of the traces of the prismatic layer on the side surfaces ('side walls'). Single-cell-wide offset obviously is not compatible with curved side walls.

Nevertheless suggested technique still can be very useful when mesh should follow the normal mesh size distribution with very high precision irrespectively of the size and shape of the surface elements. Rather distorted cells can be created at a certain distance from the boundary but inverted mesh cells cannot appear since they result in the infinite value of the discrete stored energy.



(a) (b)

Fig. 12: (a) Cavity with undercuts and resulting decrease of the thickness of prismatic layer, (b) 3D cavity model.

Another potential application of single springback technique is simultaneous optimization of meshes on the surface of the model and on the outer surface of the layer. For thick prismatic layers such a procedure can be quite useful since in many cases it is not possible to predict the shape of triangles on the offset surface. Perfect mesh on initial surface may result in highly distorted offset surface mesh. This problem is especially pronounced when highly curved edges are present on the surface.

Obvious alternative to single springback algorithm is multiple springback algorithm. This algorithm can be formulated as follows:

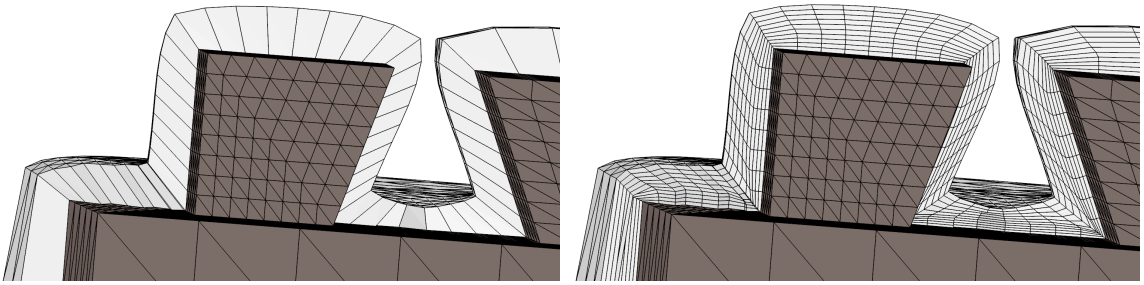
- prescribe excess ratio $E_r \geqslant 1$;
- for $k = 1, 2, \ldots$
  - precompute target thickness h_k for kth layer of the prismatic mesh and assign to this layer springback target thickness $E_r h_k$;
  - construct the kth layer using springback technique;
  - eliminate self-intersections and overlaps;
  - smooth outer surface of kth layer;
- extract mesh fragments where thickness distribution is far from the target one and apply variational optimization in order to recover constrained optimal thickness distribution.

The smoothing algorithm plays crucial role in control of behaviour of prismatic layer. To this end we use discrete biharmonic smoothing [17] with one-sided displacements. Namely, let v_i denote the vertex on the inner surface of kth layer after springback and elimination of self-intersections. Its counterpart on the outer surface is denoted by p_i. We consider surface created by a set of vertices

$$v_i + t_i(p_i - v_i), \quad t_i = \frac{1}{E_r}, \quad i = 1, \ldots, n_v$$

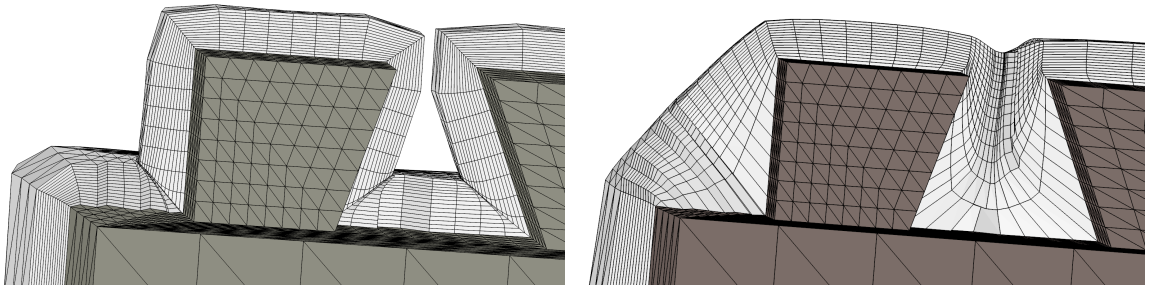
as initial guess for smoothing. The unknown variables for smoothing of the outer surface are parameters t_i, $i = 1, \ldots, n_v$, where n_v is number of vertices in the surface mesh. Layer growth is guaranteed by suppressing negative increments for t_i. In fact this algorithm constructs approximate solution to the problem of finding minimal curvature surface lying between initial surface and outer surface. Simple Laplace-Beltrami smoothing is not suitable here since it tends to create saddle surfaces as solutions thus not allowing to control the shape of layers.

We test the ability of multiple springback algorithm to construct thick prismatic layer using 'cavity with undercuts' model shown in Fig. 12b.



(a) (b)

Fig. 13: (a) Single springback layer, (b) result of splitting and orthogonalization.



(a) (b)

Fig. 14: (a) Multiple springback layer with $E_r = 2$, (b) multiple springback layer with $E_r = 20$.

Single springback algorithm results are shown in Fig. 13. As expected, the shape of the cavity does not allow to build thick prismatic layer. Fragments of prismatic meshes generated by multiple springback with excess ratio $E_r = 2$ and $E_r = 20$ are shown in Fig. 14.

Prismatic layer generated with small excess ratio is very close to orthogonal one. As one can expect, outer surface of the layer resembles isosurface of the distance function. Moreover, prism collapse is observed, which is predictable as well. In this test case local mesh quality optimization via adaptive choice of weight is not used to make relation between orthogonality and prism collapse visible. Layer generated with large excess ratio is very stable and thick for quite complicated geometries, but thickness of near-wall layers is much larger compared to the target one. One has to apply variational optimization to a mesh fragment containing all layers which would restore mesh size and orthogonality near boundary.

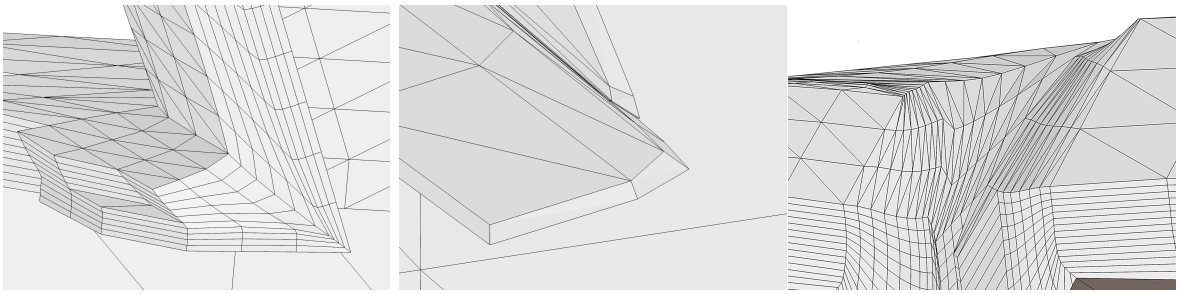
Figure 15 shows the size and shape of last prisms of the layers for different algorithms. As one can see for thick layers considerable distortion of prisms near outer boundary of layer is inevitable.

Figures 16 and 17 illustrate comparative behaviour of single springback algorithm and multiple springback algorithm depending on parameter E_r.

Figures 18 and 19 show enlarged fragments of the above prismatic layers. These fragments are candidates for post-optimization. Multiple springback algorithm tends to create larger prisms near boundary and smaller prisms near outer constrained surface.

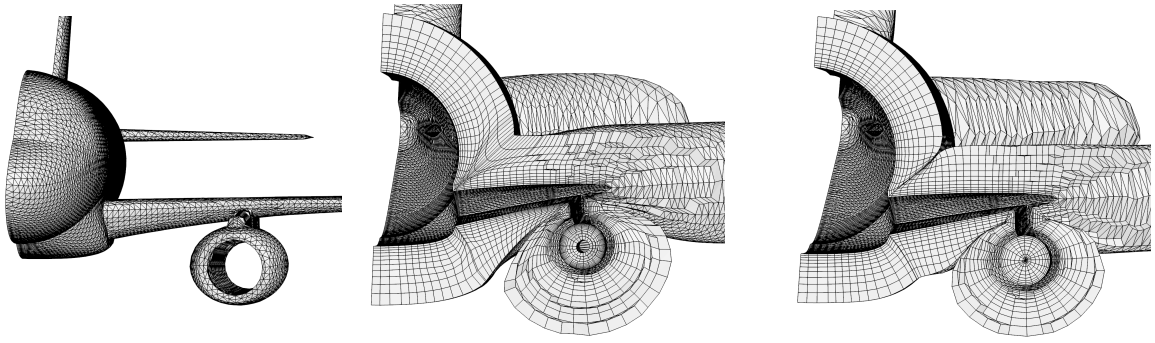
8 Discussion

We briefly discuss here the way to reduce the computational costs of the algorithm. First of all better initial guess should be constructed. It can be done by constrained smoothing the field of surface normals/transverse



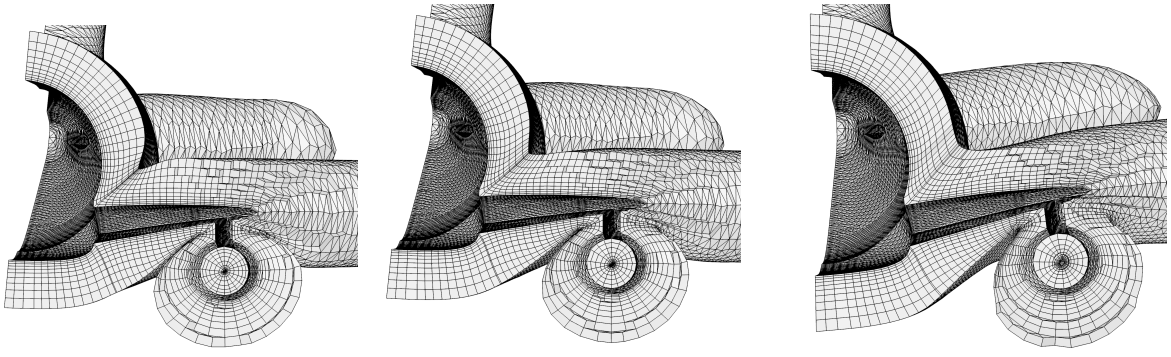
(a) (b) (c)

Fig. 15: Distorted prisms near outer surface of layer: (a) single springback layer, (b) multiple springback layer with $E_r = 2$, (c) multiple springback layer with $E_r = 20$.



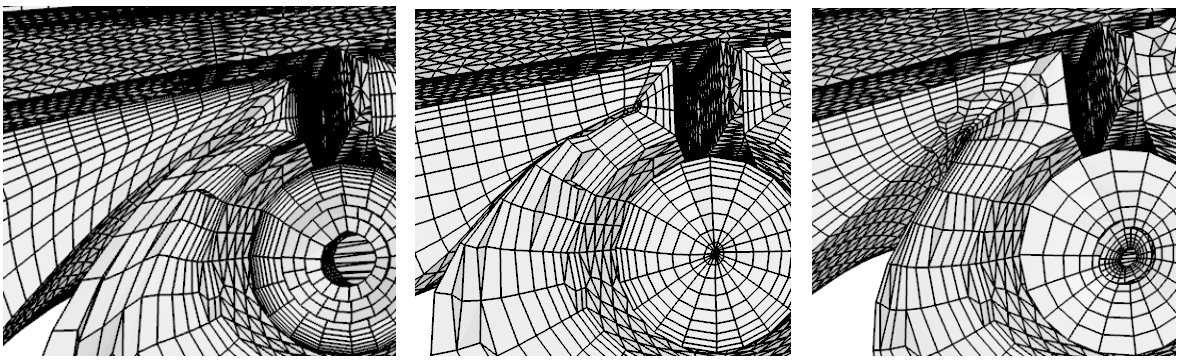
(a) (b) (c)

Fig. 16: (a) Simplified airplane model [7], (b) single springback mesh layer, (c) multiple springback layers: (a) $E_r = 1$.



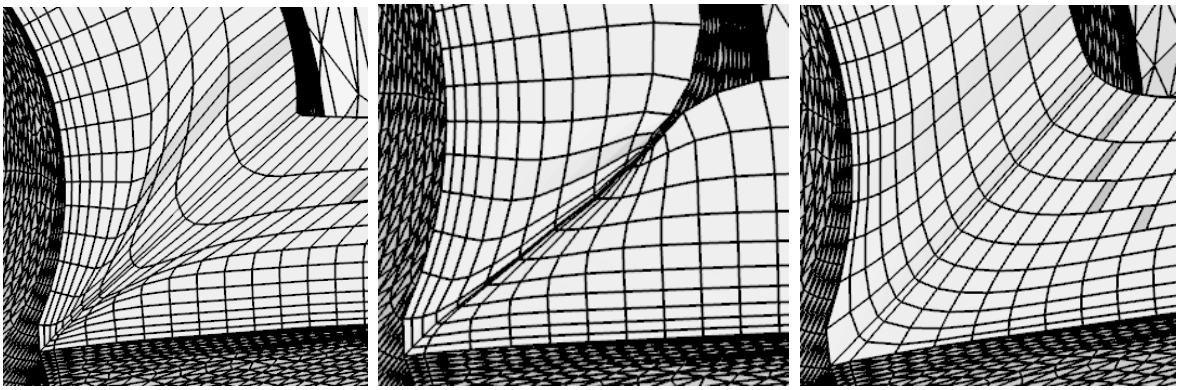
(a) (b) (c)

Fig. 17: Multiple springback layers: (a) $E_r = 2$, (b) $E_r = 3$, (c) $E_r = 5$.



(a) (b) (c)

Fig. 18: Prismatic layer behaviour near corner: (a) single springback layer, (b) multiple springback layer with $E_r = 1$, (c) multiple springback layer with $E_r = 5$.



(a) (b) (c)

Fig. 19: Prismatic layer behaviour near corner: (a) single springback layer, (b) multiple springback layer with $E_r = 1$, (c) multiple springback layer with $E_r = 5$.

directions. This smoothing should take into account sharp features on the surface of the model [16]. Generally, normals for vertices lying on sharp features are smoothed only taking into account their neighbors on the features.

This simple correction of algorithm may sharply increase the admissible thickness for majority of prismatic cells of the initial layer. In principle variational optimization can be applied locally, only around the zones where degenerate prisms are found. One should be quite careful since straightforward application of springback technique to the fragment of thin layer surrounded by thick layer results in very poor performance of iterative minimization due to the prisms with highly skewed upper lid.

We use springback technique as a part of the algorithm for hybrid polyhedral meshing which couples polygonal prisms with generalized polyhedra. The preliminary part of this algorithm is construction of exact Delaunay tetrahedral mesh as described in [2]. All boundary faces in this algorithm are Delaunay ones. We use resulting boundary mesh and construct offset surface using springback algorithm. Tetrahedral mesh becomes tangled due to boundary displacement. Untangling technique described in [5] and [7] is used recover correct tetrahedral mesh. During untangling and optimization step mesh cells from the initial Delaunay mesh provide target shapes and sizes for the mesh optimized using functional (1.1). Hence precisely the same optimization procedure is used for all types of mesh cells. Numerical evidence suggests that unlike springback problem, untangling and optimization of isotropic tetrahedral meshes is not stiff variational problem.

A sample hybrid mesh is shown in Fig. 20.

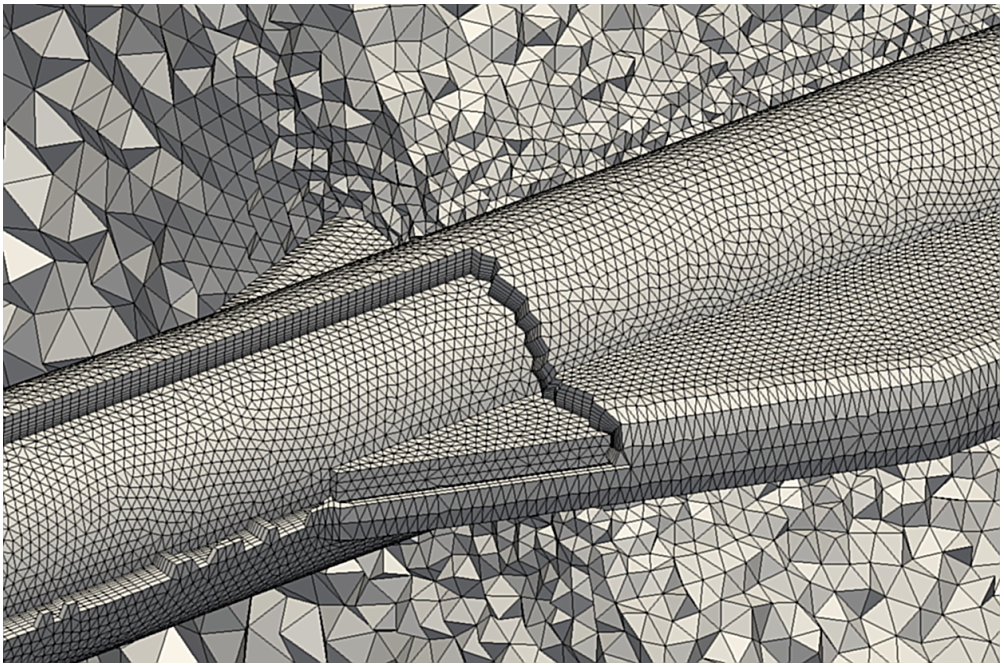


Fig. 20: Hybrid tetraprism mesh.

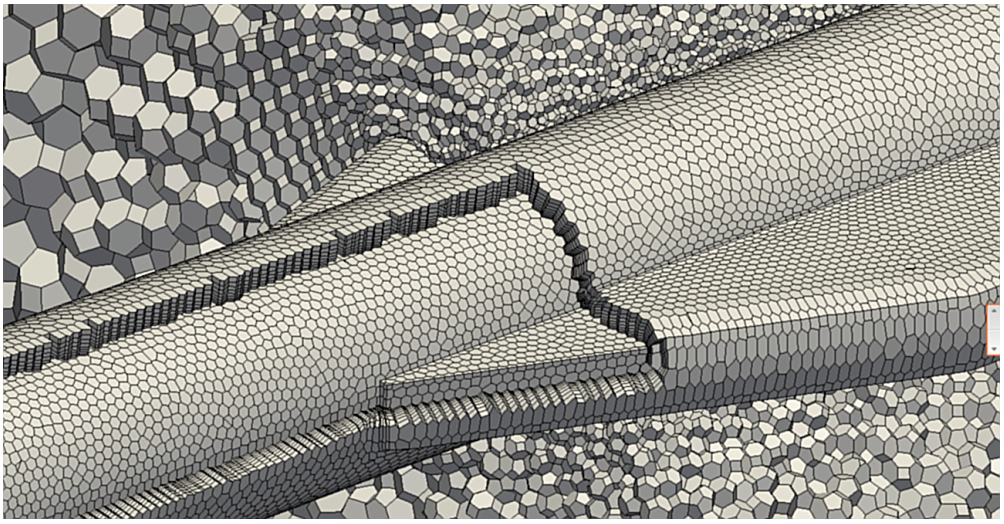


Fig. 21: Hybrid polygonal prism/generalized polyhedral mesh.

After optimization tetrahedral mesh is no longer Delaunay one, but the number of non-Delaunay cells is quite small. Since we do not use topological mesh corrections quite distorted tetrahedra potentially may appear, but mesh remains geometrically and topologically correct.

Polyhedral mesh is built via approximate dualization, similar to [10]. We compute generalized centers for tetrahedra of initial Delaunay mesh as linear combinations of circumcenters and geometric centers to keep centers strictly inside tetrahedra. Barycentric coordinates of generalized centers are stored so their positions after deformation of tetrahedral mesh can be restored. Note that Delaunay mesh generator [2] produces quite small number of tetrahedrons which do not contain their circumcenters.

Similar procedure is applied to surface triangles: we compute generalized centers, build upon them polygones and construct polygonal prisms creating transverse edges assuming constant barycentric coordinates for successive triangles. This procedure guarantees that generalized polygonal prisms in the final prismatic

layer and generalized polyhedra in the core of the computational domain join without gaps. Hybrid polyhedral mesh is shown in Fig. 21.

Numerical experiments confirm that prismatic layer mesh generator described above serves as a robust tool which allows one to build high quality hybrid meshes of different types.

Funding: This work was supported by the Ministry of Science and Higher Education of the Russian Federation, project No. 075-15-2020-799.

References

[1] L. V. Branets and V. A. Garanzha, Distortion measure for trilinear mapping. Application to 3-D grid generation. *Num. Linear Algebra Appl.* **9** (2002), 511–526.

[2] A. I. Belokrys-Fedotov, V. A. Garanzha, and L. N. Kudryavtseva, Delaunay meshing of implicit domains with boundary edge sharpening and sliver elimination. *Math. Comp. Simul.* **147** (2018), 2–26.

[3] V. Dyedov, D. Einstein, X. Jiao, A. Kuprat, J. Carson, and F. del Pin, Variational generation of prismatic boundary-layer meshes for biomedical computing. *Int. J. Numer. Methods Engrg.* **79** (2009), 907–945.

[4] M. S. Floater and K. Hormann, *Surface Parameterization: a Tutorial and Survey. Advances in Multiresolution for Geometric Modelling, Mathematics and Visualization. Part 4*. Springer, Berlin–Heidelberg–New York, 2005.

[5] V. A. Garanzha and I. E. Kaporin, Regularization of the barrier variational method for constructing computational grids. *Comput. Math. Math. Phys.* **39** (1999), 1426–1440.

[6] V. A. Garanzha, The barrier method for constructing quasi-isometric grids. *Comput. Math. Math. Phys.* **40** (2000), 1617–1637.

[7] V. A. Garanzha, L. N. Kudryavtseva, and S. V. Utyzhnikov, Untangling and optimization of spatial meshes. *J. Comput. Appl. Math.* **269** (2014), 24–41.

[8] V. A. Garanzha, and L. N. Kudryavtseva, Hyperelastic springback technique for construction of prismatic mesh layers. *Procedia Engrg.* **203** (2017), 401–413.

[9] R. V. Garimella and M. Shephard, Boundary layer mesh generation for viscous flow simulations. *Int. J. Numer. Methods Engrg.* **49** (2000), 193–218.

[10] R. Garimella, J. Kim, and M. Berndt, Polyhedral mesh generation and optimization for non-manifold domains. *Proc. of the 22nd Int. Meshing Roundtable*. 2014, 313–330.

[11] S. A. Ivanenko, Construction of nondegenerate grids. *Comput. Math. Math. Phys.* **28** (1988), 141–146.

[12] S. A. Ivanenko, A variational form of the Winslow grid generator. *J. Comput. Physics* **136** (1997), 385–398.

[13] Y. Kallinderis and S. Ward, Prismatic grid generation for three-dimensional complex geometries. *AIAA J.* **31** (1993), 1850–1856.

[14] I. E. Kaporin, High quality preconditioning of a general symmetric positive definite matrix based on its $U^TU + U^TR + R^TU$ decomposition. *Numer. Linear Algebra Appl.* **5** (1998), 483–509.

[15] A. Loseille and R. Löhner, Robust boundary layer mesh generation. *Proc. of the 21st Int. Meshing Roundtable*. 2013, 493–511.

[16] M. Tomac and D. Eller, Towards automated hybrid-prismatic mesh generation. *Procedia Engrg.* **82** (2014), 377–389.

[17] M. Wardetzky, M. Bergou, D. Harmon, D. Zorin, and E. Grinspun, Discrete quadratic curvature energies. *Computer Aided Geometric Design* **24** (2007), 499–518.