Gradient projection based optimization methods for untangling and optimization of 3d meshes in implicit domains

V.A. Garanzha*, L.N. Kudriavtseva †

*Computing Center RAS, garan@ccas.ru
†Computing Center RAS, arnir@rambler.ru

Definition of complex domain as zero isosurface of scalar function resembling signed distance function is powerful and flexible mechanism of geometric modeling. Signed implicit functions can be constructed using surface triangulation, point clouds, set of planar cross-sections, "soup" consisting of disjoint edges and faces, and can be combined with analytical definitions and primitives using boolean operations. In order to construct tetrahedral meshes in implicit domains, special iterative technique for self-organization of points according to prescribed sizing function was developed in [1]. The points are distributed in such a way that sharp edges on the isosurface are approximated by Delaunay edges without special edge detection procedures. Thus meshes in volume and on the surface are constructed simultaneously.

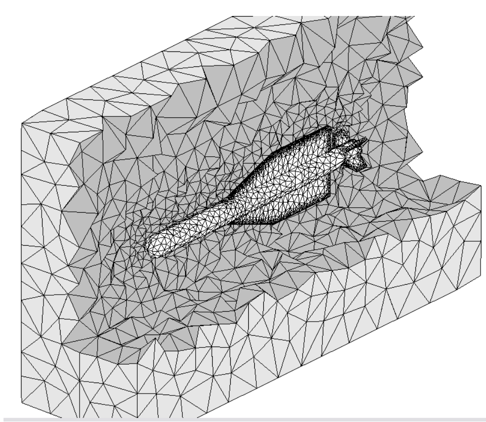

Fig. 1. Tetrahedral mesh around body which was built from primitives using boolean operations.

This algorithm has heuristic nature but it has demonstrated its applicability to relatively complex test cases. In resulting Delaunay mesh a few flat tetrahedra (slivers) can be present. For their elimination we use variational optimization method [2] allowing vertex movement along boundary.

Essentially the same method is used to construct structured hexahedral meshes. In this case one should use special algorithm for variational construction of surface meshes on implicit surfaces, and to optimize simultaneously meshes in volume and on the surface. Variational optimization is based on iterative technique which is closely related to gradient projection based optimization methods.

Consider in more detail variational problem for shape and volume optimization of mesh cells. Consider tetrahedral mesh $\mathcal{T}$ in polyhedral domain Ω_h consisting of n_s tetrahedra with n_v vertices, including n_b boundary vertices. Denote by z_i, $i = 1 \ldots n_v$ vertices of the mesh $\mathcal{T}$ and introduce on each tetrahedron $T_k \in \mathcal{T}$ local numbering of vertices y_0, y_1, y_2, y_3. For each tetrahedron we introduce the so-called target tetrahedron, which is chosen here as equilateral one. Denote by ζ_0, ζ_1, ζ_2, ζ_3 the vertices of target tetrahedron T_k^t.

Consider matrices $H = (\zeta_1 - \zeta_0 \; \zeta_2 - \zeta_0 \; \zeta_3 - \zeta_0)$ and $Q = (y_1 - y_0 \; y_2 - y_0 \; y_3 - y_0)$. It is assumed that their determinants are positive. The volume of mesh tetrahedron and target tetrahedron is defined by $\operatorname{vol} T_k^t = \frac{1}{6} \det H, \operatorname{vol} T_k = \frac{1}{6} \det Q$, while the Jacobian matrix $S = \nabla_\xi x^h$ of affine mapping $x^h(\xi) : T_k^t \to T_k$ is written as $S = QH^{-1}$.

Consider the following functional which may serve as distortion measure for mesh cells

$$F(z_1,\dots,z_{n_v}) = \sum_{k=1}^{n_s} \varphi(QH^{-1})\Big|_{T_k^t} \operatorname{vol} T_k^t = \qquad (1)$$

$$= \frac{1}{6}\sum_{k=1}^{n_s} \varphi(QH^{-1}) \det H\Big|_{T_k^t},$$

where

$$\varphi(S) = \theta\mu(S) + (1-\theta)\nu(S) \qquad (2)$$

Function φ is the sum of shape distortion measure

$$\nu(S) = \frac{1}{3}\frac{\operatorname{tr}\, S^T S}{(\det S)^{2/3}} \qquad (3)$$

and volume distortion measure

$$\mu(S) = \frac{1}{2}\left(\frac{v_0}{\det S} + \frac{\det S}{v_0}\right) \qquad (4)$$

Here $\theta = 0.8$ and parameter $v_0 = f(c)^3$ is ratio of target cell volume to volume of equilateral tetrahedron with unit edge length, while $f(c)$ is the value of prescribed size function computed in the centroid of the tetrahedron T_k.

We consider two types of boundary conditions. The first one is the boundary condition of the first kind when boundary vertex z_k is fixed. The second case is a slip boundary condition when in the process of optimization the vertex z_k can move along the implicit surface

$$u(x) = 0$$

We assume that vector $\nabla u(z_k)$ is defined. If function u does not have classical derivative at z_k, one can still define tangent cone to $\partial\Omega$ at z_k which defines the set of generalized derivative vectors. Thus one can compute vectors l_1, l_2 which can be considered as tangent vectors to $\partial\Omega$ and satisfy equality $l_i^T \nabla u(x_k) = 0$. In practice gradient is computed using finite differences.

The condition of stationarity of the functional at the vertex z_k can be written as

$$l_i^T \frac{\partial F}{\partial z_k} = 0,\ i = 1,2 \qquad (5)$$

$$u(z_k) = 0 \qquad (6)$$

These three equations correspond to three degrees of freedom in vector z_k.

Let δz_k denote displacement at z_k. Linearizing equation (6) with respect to δz_k one can obtain the following relation

$$\delta z_k^T \nabla u(z_k) + u(z_k) = 0$$

Since for boundary vertex $u(z_k) = 0$, displacement δz_k can be represented as

$$\delta z_k = \beta_1 l_1 + \beta_2 l_2, \qquad (7)$$

where β_i are arbitrary coefficients. In other words, boundary displacement is allowed only in the plane tangent to $\partial\Omega$.

Since after tangent displacement vertex can move away from exact boundary, one should introduce the boundary projection operator $\mathcal{V}$ which is identity operator for any internal vertex and projects every boundary vertex onto $\partial\Omega$ using the following simple iterative scheme.

$$p^{m+1} = p^m - \tau_1 \frac{u(p^m)\nabla u(p^m)}{|\nabla u(p^m)|^2} \qquad (8)$$

Here parameter m is local iteration count. When $u(x)$ is linear function and $\tau_1 = 1$, then formula (8) is just normal projector onto plane $u(x) = 0$. In general case iterates approximately travel along gradient curves of $u(x)$ until deviation of p^m from $\partial\Omega$ is below certain threshold.

Gradient vector R of function $F(z_1,\dots,z_{n_v})$ consists of 3d vectors $r_k = \frac{\partial F}{\partial z_k}$, while Hessian matrix H of F is built from 3×3 blocks $H_{ij} = \frac{\partial^2 F}{\partial z_i \partial z_j^T}$, where matrix H_{ij} is placed on the intersection of i-th block row and j-th block column.

The Newton-Raphson method for finding stationary point of mesh functional without slip condition can be written as follows

$$\sum_{j=1}^{n_v} H_{ij}(Z^l)\delta z_j + r_i(Z^l) = 0 \qquad (9)$$

$$z_k^{l+1} = z_k^l - \tau_l \delta z_k,\ k = 1,\dots,n_v \qquad (10)$$

Denote by L_k the 3×3 matrix with first two columns being vectors l_i computed at boundary vertex z_k, while last column is equal to zero. If vertex z_k is internal one then $L_k = I$.

In order to include slip condition into iterative technique (9), (10), let us multiply equality (9) by L_i^T from the left and take into account the fact that boundary displacement δz_j satisfies (7), namely

$$\delta z_j = L_j \begin{pmatrix} \alpha_j \\ 0 \end{pmatrix}$$

Hence in linear system (9) one can use 2d vector α_j as unknown instead of δz_j. Denote by $\delta \tilde{z}_j$ displacement vector equal to δz_j for internal vertices and equal to $(\alpha_j^T, 0)^T$ for boundary vertices. Hence $\delta z_j = L_j \delta \tilde{z}_j$. Using above notations one can write iterative scheme for finding stationary point of the function $F(Z)$.

$$\sum_{j=1}^{n_v} L_i^T H_{ij} L_j \delta \tilde{z}_j + L_i^T r_i(Z^l) = 0 \qquad (11)$$

$$z_k^{l+1} = V(z_k^l - \tau_l L_k \delta \tilde{z}_k),\ k = 1, \ldots, n_v. \qquad (12)$$

Equality (12) can be written as

$$Z^{l+1} = V(Z^l - \tau_l \delta Z),$$

Relaxation parameter τ_l can be found using search technique for 1d minization problem

$$\tau_l = \arg\min_{\tau} F(V(Z^l - \tau \delta Z)) \qquad (13)$$

We use simplest dihotomy method for finding τ_l. In principle one could use well-known Armijo technique as 1d solver, however function F is not Lipschitz continuous hence applicability of Armijo scheme is still under question.

In order to deduce from generic formulation (11) practical iterative technique essentially identical to barrier method of Ivanenko [3] one should set $H_{ij} = 0$ when $i \neq j$. Then in order to find $\delta \tilde{z}_i$ it is necessary to solve independent linear systems with 3×3 matrices at internal nodes and 2×2 matrices at the boundary vertices. It is convenient to call resulting preconditioning by diagonal scaling. Simple explicit gradient search technique can be obtained by setting $H = I$. Main advantage of this method is that time consuming second derivatives are not computed, however its convergence may suffer and 1d search stage (13) may require large number of steps and become quite costly.

In order to derive from (11) implicit method [2] one should eliminate in matrices $L_i^T H_{ij} L_j$ all off-diagonal entries. The resulting linear system $L_i^T H_{ij} L_j$ (11) will be decomposed into three independent linear systems with respect to vectors $\delta \tilde{Z}_m$ which are obtained from $\delta \tilde{z}_i$ using identity

$$(\delta \tilde{Z}_m)_i = (\delta \tilde{z}_i)_m$$

Linear systems with these matrices are solved using preconditioned conjugate gradient (CG) method. Fortunately there is no need to solve these systems with high accuracy.

Variational method can be used in the case when algebraic volume of some tetrahedra in initial mesh is not positive. Special "untangling" technique suggested in [4] was quite efficient tool for constructing admissible meshes. The idea of this technique is based on replacing the barrier by penalty function, namely the determinant of Jacobian matrix in denominator of (2) is replaced by

$$\chi(\det S) = \frac{1}{2}(\det S + \sqrt{\varepsilon^2 + \det S^2})$$

Bad mesh is untangled using special continuation technique for parameter ε from relatively large values to zero. Special untangling technique is suggested which is robust and efficient enough in order to construct admissible meshes with hundreds of thousands of vertices. Algorithm which solves quite hard problem of surface mesh untangling on implicit surfaces is also developed. It turned out that untangling procedure plays critical role in automatic hexahedral meshing. After untangling and optimization of initial medium-sized mesh, one can construct huge meshes using relatively simple local refinement and optimization scheme without mesh quality deterioration.

We have found distinct application fields for explicit methods and for preconditioned methods. Preconditioned methods are good choice for global surface flattening with minimal distortion, global untangling for complicated domains. Their main advantages are stability, robustness and good convergence, and main drawback - relatively large memory consumption. Applications of diagonal scaling and explicit methods include admissible mesh smoothing and optimization especially in the case of slip boundary conditions, local untangling. These methods are much faster for suitable applications, in some cases one can just avoid costly computation of second derivatives. Unfortunately it was found that explicit methods simply cannot solve some problems.

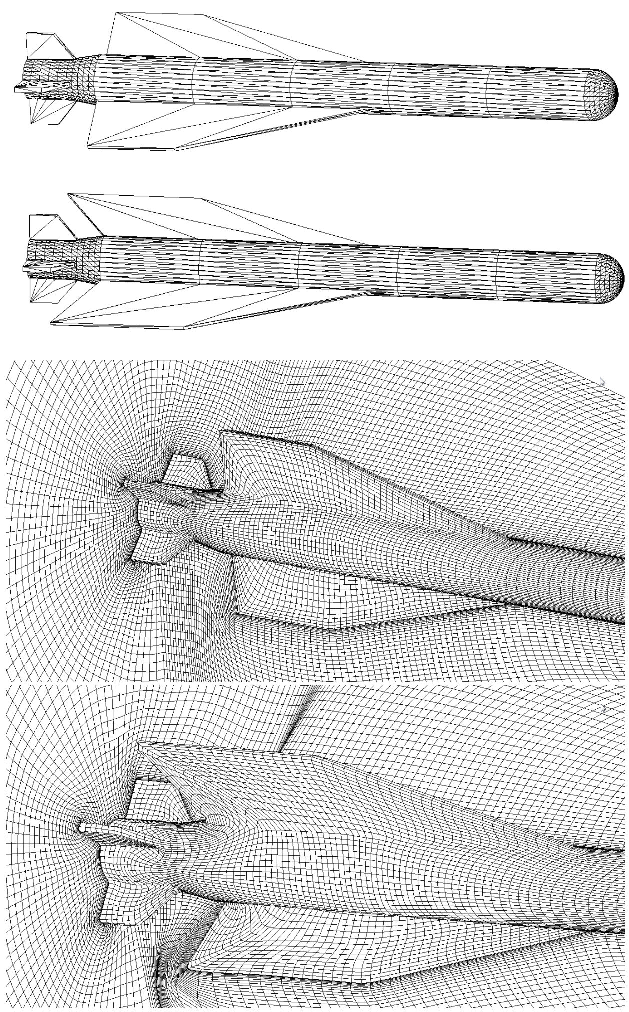

Fig. 2. Zero isosurfaces (above) and structured hexahedral meshes. Several coordinate surfaces are shown.

Fig. 2 illustrates test case where signed implicit function is constructed using surface triangulation (STL model). Hexahedral mesh which is compressed and orthogonalized near boundary is constructed in a black-box mode. Let us note that in order to adjust mesh taking into account precise CAD data one can just use function which computes distance from the surface. Such functions are available in modern geometry kernels.

Optimization problem in this case involves about 10 millions of tetrahedra. Explicit solver and diagonal scaling-based solver were not able to do the untangling. Implicit method using CG-based linear solver with second order incomplete Choleski preconditioner was found to be too memory consuming. Thus implicit solver only CG-based solver without preconditioner was able to solve this problem. After initial untangling diagonal scaling-based solver is good option for mesh optimization.

Partial support of grants P-17 of Presidium RAS and NSh-4096.2010.1 is gratefully acknowledged.

References

[1] Garanzha V.A., Kudriavtseva L.N. *Delaunay meshing using incomplete and contradictory data.* Submitted to Comput. Math. Math. Phys., 2011.

[2] Garanzha V.A. *The barrier method for constructing quasi-isometric grids* // Comput. Math. Math. Phys. 2000. V.40. N.11. P.1617-1637.

[3] Ivanenko, S. A. *Construction of nondegenerate grids* // U.S.S.R. Comput. Math. and Math. Phys. 1988. V.28. N.5. P.141-146.

[4] Garanzha V.A., Kaporin I.E. *Regularization of the barrier variational method for constructing computational grids* // Comput. Math. Math. Phys. 1999. V.39. N.9. P.1426-1440.