

# Discretization of Functionally Based Heterogeneous Objects

Elena Kartasheva

Institute for Mathematical Modeling  
Russian Academy of Science  
Moscow, Russia  
ekart@imamod.ru

Valery Adzhiev

The National Centre for Computer  
Animation, Bournemouth University  
Poole, BH12 5BB UK  
vadzhiev@bournemouth.ac.uk

Alexander Pasko

Hosei University  
Tokyo, Japan  
pasko@k.hosei.ac.jp

Oleg Fryazinov

Institute for Mathematical Modeling  
Russian Academy of Science  
Moscow, Russia  
fryazinov@imamod.ru

Vladimir Gasilov

Institute for Mathematical Modeling  
Russian Academy of Science  
Moscow, Russia  
gasilov@imamod.ru

## ABSTRACT

The presented approach to discretization of functionally defined heterogeneous objects is oriented towards applications associated with numerical simulation procedures, for example, finite element analysis (FEA). Such applications impose specific constraints upon the resulting surface and volume meshes in terms of their topology and metric characteristics, exactness of the geometry approximation, and conformity with initial attributes. The function representation of the initial object is converted into the resulting cellular representation described by a simplicial complex. We consider in detail all phases of the discretization algorithm from initial surface polygonization to final tetrahedral mesh generation and its adaptation to special FEA needs. The initial object attributes are used at all steps both for controlling geometry and topology of the resulting object and for calculating new attributes for the resulting cellular representation.

## Categories and Subject Descriptors

I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling – *Curve, surface, solid, and object modeling, Physically based modeling*; I.3.6 [Computer Graphics]: *Methodology and Techniques*; I.3.8 [Computer Graphics]: *Applications*.

## General Terms

Algorithms, Design.

## Keywords

Heterogeneous objects, attributes, function representation, cellular representation, volume modeling, constructive hypervolume, finite element analysis, mesh.

## 1. INTRODUCTION

In this paper, we deal with generation of discrete models for heterogeneous objects defined using real-valued functions. Generally, heterogeneous objects have an internal structure with non-uniform distribution of material and other attributes of an arbitrary nature (photometric, physical, statistical, etc.), and elements of different dimension. Recently we can observe a steady

interest in functionally based geometric models such as implicit surfaces, functionally defined solids and heterogeneous objects (see [28, 1, 2] for details). These models provide compact and intuitive mathematical representation for complex heterogeneous objects, support set-theoretic and other operations such as offsetting, blending, and sweeping. Rapid development of hardware, computational algorithms, and specialized software allow for manipulation of such models at interactive rates. Practical applications of functionally based models in CAD/CAM/CAE and finite element analysis (FEA) require some key procedures to be also applicable to these models. Numerical FEA methods use discrete models (surface and volume meshes) of geometric objects, although meshfree analysis and simulation methods are also emerging [34]. Algorithms for finite element mesh generation are well developed for boundary and spatial enumeration representations [8]. Meshes are also actively used now in visualization, animation, computational geometry, image processing, and other areas. However, requirements for discrete models in FEA are stricter than in other areas. Such requirements are formulated in terms of discrete model topology and metric characteristics, exactness of the geometry approximation, and conformity with initial attributes. This results from the use of meshes in FEA for approximation of systems of equations of mathematical physics. The size and shape of mesh elements and mesh structure seriously influence the stability of numerical simulation procedures and accuracy of obtained solutions. That is why meshes used for visualization usually do not satisfy FEA requirements and special refinement of them is needed. The above is the motivation for work on discrete model generation for functionally based surfaces, solids, and heterogeneous objects.

In this paper, we deal with the discretization problem within the hybrid cellular-functional model [1] of heterogeneous objects. The function representation of the initial 3D heterogeneous object is converted into the resulting cellular representation described by a simplicial complex. We consider in detail the following phases of the discretization algorithm: initial surface polygonization; iterative simplification and refinement of the surface mesh along with sharp features reconstruction; further adaptation of the surface mesh according to constraints depending on the attributes of the initial object; volume (tetrahedral) mesh generation using a modified advancing front method; its subsequent adaptation to special FEA needs. The initial object attributes are used at all steps both for controlling geometry and topology of the resulting object and for calculating new attributes for the resulting cellular representation.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SM '03, June 16-20, 2003, Seattle, Washington, USA.

Copyright 2003 ACM 1-58113-706-0/03/0006...\$5.00.

## 2. PROBLEM STATEMENT

In this section, we provide a formulation of the discretization problem in terms of initial and resulting objects along with a set of special requirements. We consider the problem of discretization of functionally based heterogeneous objects within a hybrid cellular-functional representation framework [1] in which objects are treated as hypervolumes (multidimensional point sets with multiple attributes) [28].

### 2.1 Initial heterogeneous object

Let  $D$  be an initial heterogeneous object – a hypervolume expressed by a tuple:

$$D = (G, A_1, \dots, A_k),$$

where  $G$  is a 3D point set,  $A_i$  is an attribute and  $k$  is a number of attributes. We assume that the object's geometry  $G$  is described by the function representation (FRep) [27]:

$$G_F = \{X \mid X = (x_1, x_2, x_3) \in \Omega \subseteq E^3, F(X) \geq 0\},$$

where  $\Omega$  is a modeling space and  $F: \Omega \rightarrow \mathcal{R}$  is (at least a  $C^0$  continuous) real-valued defining function. Note that the boundary of the object  $D$  is an implicit surface described as

$$B_F = \{X \mid X = (x_1, x_2, x_3) \in \Omega \subseteq E^3, F(X) = 0\}.$$

Each attribute  $A_i$  is defined by its set of values  $N_i \subseteq \mathcal{R}^{m_i}$  along with a map function  $S_i(X): \Omega \rightarrow N_i$  and can be represented by any of the attribute models introduced in [1] that differ from each other in the way of defining  $S_i(X)$ . For instance, the function representation (FRep) can be used for defining attributes representing electric or thermal field distribution as well as load distribution. Cellular-functional representation (CFRep) is especially suitable for description of material attributes. Cellular models of attributes (CRep) are used in finite element calculations for representation of functions describing simulated processes.

What is important in the context of our consideration of the discretization problem is that attributes can provide a formal description of requirements and constraints imposed by FEA. Thus, constraints on the size of elements can be expressed through the mesh density attribute  $A_r$  whose map function  $S_r(X)$  defines a proper element size at each point  $X$  of the modeling space  $\Omega$ . Such an attribute can be given by the user or can be calculated on the basis of other given attributes  $A_i$ . There is a promising way of defining  $A_r$  through FRep on the basis of "sources" [19], [1]. As to defining  $A_r$  through CRep, it is appropriate when the size distribution function is defined in a discrete manner and its values are known at the vertices of a background geometric complex. Such a complex can describe either a regular mesh specially built for defining the attribute  $A_r$  or an FE mesh used in the previous steps of adaptive numerical calculations.

### 2.2 Resulting heterogeneous object

Given the initial object  $D$ , we are going to build a resulting heterogeneous object – a hypervolume  $\tilde{D} = (\tilde{G}, \tilde{A}_1, \dots, \tilde{A}_m)$ . Here, the geometry component  $\tilde{G}$  is an approximate discrete representation of the initial geometry  $G$ , and attributes  $(\tilde{A}_1, \dots, \tilde{A}_m)$

describe the object's properties. Formally,  $\tilde{G}$  can be expressed as a particular case of a model based the cellular representation (CRep) [1]:

$$\tilde{G}_c = \{X \mid X \in \Omega \subseteq E^3, X \in |K^3|\},$$

where  $K^3 = \{C_i^r; r = 0, 1, 2, 3; i = 1, \dots, I_r\}$  is a three-dimensional polyhedral complex consisting of cells  $C_i^r$ . In conventional terms, such a discrete model is called a mesh. As we have already stated, the resulting mesh  $\tilde{G}_c$  depends not only on the initial geometry  $G$  but also on the attributes  $A_1, \dots, A_k$  (in particular, on the mesh density attribute  $A_r$ ):  $\tilde{G}_c = \tilde{G}_c(G_F, A_1, \dots, A_k)$ .

Note that the attributes  $\tilde{A}_1, \dots, \tilde{A}_m$  can differ from the initial ones  $A_1, \dots, A_k$  in terms of their number, set of values, and description. Some of the initial attributes may not have direct counterparts in the discrete model (e.g., there may be no need to retain the mesh density attribute). Other attributes associated with the hypervolume  $\tilde{D}$  can have the same meanings and similar values as their initial counterparts but may be described by another representational scheme. For instance, the material property being described by FRep in  $D$  can be represented by CFRep in  $\tilde{D}$  allowing each cell to have its own material index. Completely new attributes can also appear in  $\tilde{D}$ . Thus, such an attribute can describe normals to the initial implicit surface at the nodes of a mesh or can represent values of 3D cell volumes evaluated in advance and useful for speeding up FE-based calculations. So, in general we have the resulting attributes defined as:

$$\tilde{A}_j = \tilde{A}_j(G, \tilde{G}, A_1, \dots, A_k), \quad j = 1, \dots, m.$$

### 2.3 Problem formulation

Now we can formulate the discretization problem as the problem of conversion of the initial functionally based heterogeneous object  $D$  into the object  $\tilde{D}$  with discrete (mesh-based) geometry:

$$D = (G_F, A_1, \dots, A_k) \rightarrow \tilde{D} = (\tilde{G}_c, \tilde{A}_1, \dots, \tilde{A}_m),$$

where  $G_F$  is an FRep based model, and  $\tilde{G}_c$  is a CRep based model. The boundary of the object  $D$  is an implicit surface  $B_F$ . As to the boundary  $\tilde{B}_c$  of  $\tilde{D}$ , within the entire discrete model  $\tilde{G}_c$  it can be described through a polyhedral complex, which is a boundary subcomplex  $L^2$  of the complex  $K^3$ , such as  $L^2 \subset K^3$  and  $L^2 = \{C_j^r; r = 0, 1, 2; j = 1, \dots, J_r\}$ .

Then  $\tilde{B}_c = \{X \mid X \in \Omega \subseteq E^3, X \in |L^2|\}$ .

### 2.4 Special requirements

Our approach is oriented towards some demanding applications such as FEA. So, we take into account the following requirements and constraints upon the resulting discrete structures (surface and volume meshes):

1. The topology of the surface mesh  $\tilde{B}$  has to conform to the topology of the boundary of the initial solid  $G$ .

2. The surface mesh  $\tilde{B}$  has to include all the sharp features of the surface B; this means that there should be conformity between initial object ridges and peaks, and edges and nodes described by the complex  $L^2$ .
3. The boundary mesh  $\tilde{B}$  must provide an adequate approximation of the underlying implicit surface B. To this end, it is necessary to bound the distance between the mesh and the initial surface and to limit the deviation of normal vectors to the mesh from those to the implicit surface.
4. There can be specific constraints upon the shape of cells included in the complexes  $K^3$  and  $L^2$ , which can be given, for instance, as  $Q(C_i) < \epsilon$ , where Q yields a shape quality measure based on the metric characteristics of tetrahedrons or triangles.
5. Some constraints concerned with proportions of adjacent cell sizes in the complexes  $K^3$  and  $L^2$  can also be imposed. They can be defined as  $M(C_i)/M(C_j) < \epsilon$ , where M is a certain measure of a tetrahedron (triangle) such as the length of its maximal side, or its volume (square), or circumscribed sphere (circle) radius. This constraint (along with the previous one) is useful for ensuring a reliable tetrahedrization and convergence of numerical simulation procedures.
6. Initial object attributes reflecting some features of FE modeling can also result in specific constraints upon surface and volume meshes. Let us describe the following cases:
  - a) Given the mesh density attribute  $A_r$  along with the corresponding map function  $S_r(X)$ , one can set the following constraint for each element  $C_i$  of  $K^3$  and  $L^2$ :  $M(S_r(X), C_i) < \epsilon$ , where M is a certain metric;
  - b) Given the attribute  $A_v$  along with its map function  $S_v(X)$ , one can set the following constraint concerned with the limitation of this function's variations within each  $C_i$  of  $K^3$  and  $L^2$ . In practice, one usually compares values of  $S_v(X)$  at some reference points (e.g., in triangle nodes) of  $C_i$ . The function  $S_v(X)$  can describe some estimated characteristics of the process being modeled or just error estimations. Attributes representing material or medium properties can also impose constraints of this kind.
  - c) Let  $S_s(X)$  be a piecewise continuous map function defined for an attribute  $A_s$ . Then, while decomposing the object G, one should take into account that function features, namely its singular points, have to coincide with mesh nodes, and lines/surfaces of discontinuity have to Cellular-functional representation (CFRep) is especially suitable for description of material attributes. Cellular models of attributes (CRep) are used in finite element calculations for representation of functions describing simulated processes. be exactly described by 1D/2D subcomplexes of the complex  $K^3$  representing the discrete model  $\tilde{G}_c$ . For example, such requirements are necessary when a computational domain is decomposed into sub-areas corresponding to different materials or different properties of medium. Singular

points and lines can also conform to disposition of sources or to load application areas.

7. If no special constraints are formulated, the number of elements (cells) in complexes  $K^3$  and  $L^2$  should be minimal. For example, an ideal surface mesh of a cube consists of twelve triangles: two triangles per each face.

### 3. RELATED WORKS

In this section, we give a brief review of works in the areas that are relevant in the context of our consideration of the discretization problem: heterogeneous object modeling, implicit surface polygonization, and finite element mesh generation and refinement.

#### 3.1 Heterogeneous objects modeling

Particular attention in solid modeling is paid to modeling heterogeneous objects with multiple materials and non-uniform internal material distribution. Boundary representation, functionally based, voxel and cellular models are used to represent such objects.

A non-manifold BRep scheme is used in [15] to subdivide an object into components made of unique materials. In the object model proposed in [16], a fiber bundle is used for general description of all characteristics and attributes of an object. Constructive operations for modeling functionally graded materials associated with a BRep geometric model are discussed in [35].

Voxel arrays in volume modeling and graphics can be considered as discrete attribute models with the default geometry represented by a bounding box. Constructive Volume Geometry (CVG) [4] utilizes voxel arrays and continuous scalar fields for representing both geometry and photometric attributes (opacity, color, etc.). Issues of functionally based modeling of volumetric distribution of attributes are also addressed in [11, 21, 26].

Modeling heterogeneous objects as multidimensional point sets with multiple attributes is discussed in [28]. The proposed constructive hypervolume model is based on function representation (FRep) [27] and supports uniform constructive modeling of point set geometry and attributes using vectors of real-valued functions of several variables. Multiple materials are also represented in [2] by vectors of real-valued functions. Distance fields are used to model varying material properties satisfying different types of constraints predefined on the initial object geometry.

The approach of [28] was extended in [1] to dimensionally heterogeneous objects with multiple attributes by combining the functionally based and cellular representations into a single hybrid model. In this paper, we describe one of the important operations in the hybrid model, namely the conversion between functionally based and cellular heterogeneous objects.

#### 3.2 Polygonization

Existing methods of the polygonal approximation (polygonization) of implicit surfaces include two major groups. The first group consisting of continuation algorithms is characterized by introducing a seed local triangulation of the implicit surface with the consecutive addition of new triangles to

the mesh by moving along the surface [36, 3, 10] with the triangle size adapted to the local surface curvature [12].

The second group includes methods for generating polygons as the result of the intersection of the implicit surface with cells of a regular grid (see, for example, [29, 20, 22]) or an adaptive grid [31]. The algorithms of this group differ in the type of grid and surface sample points approximation.

The main disadvantages of the mentioned approaches are smoothing or cutting sharp features of the surfaces. Algorithms of sharp features extraction are presented in [14, 23, 24]. The optimization [24] is based on the special vertex relocation strategy and triangles subdivision and allows for extraction of sharp edges and peaks taking into account the surface curvature. However, in the process of optimization, the shapes and relative sizes of neighboring triangles are not controlled which can result in generation of degenerate triangles. Moreover, these mesh optimization algorithms can produce an excessive number of triangles in the regions of not very high curvature, which is also undesirable for further calculations. Thus, to satisfy the discussed earlier requirements of FEA, a special mesh refinement while still preserving sharp features is needed.

### 3.3 Finite element mesh generation and refinement

Issues of surface mesh optimization for FEA are considered in detail in [8, 7]. The described techniques are based on the consecutive application of different mesh simplification, mesh subdivision, and mesh adaptation procedures. Details of such operations are discussed also in [13, 9, 33] and other works. Note that in these works the surface models are not defined in terms of analytical functions but rather by means of triangulation (resulted, for example, from measurements, CAD, biomedical engineering). During the mesh refinement, the exact definition of underlying surfaces is unknown. When optimizing polygonized implicit surfaces, we can use both approximate and precise functional surface models, which provide for more precise calculations of surface characteristics and corrections of the node positions in respect to the underlying surface for remeshing.

Issues of finite element mesh generation are discussed in detail in [8]. Unstructured mesh generation methods are also surveyed in [25, 32]. Tetrahedrization is one of the widely used methods of 3D discretization. The main approaches to automatic tetrahedral (triangular) mesh generation include spatial decomposition based methods, Delaunay type methods, and advancing-front techniques. Algorithms based on spatial decomposition are relatively easy to implement, but they do not allow for detection of boundary sharp features and cannot distinguish boundary entities which are rather close but not directly connected. The boundary connectivity constraint is not taken into account in Delaunay tetrahedrization. So, local mesh modifications are necessary to fit the boundary.

More accurate boundary representation is supported by the advancing-front method. This method starts from a domain boundary discretization and marches into the region to be processed by adding one element at a time. However, since the method is based on local operations, convergence problems may be encountered. The convergence problem is common for all methods as there is no theoretical result which can guarantee that a polyhedron with the given boundary triangulation can be

subdivided into tetrahedrons without adding internal points. In spatial decomposition methods, the convergence problems appear when refining small details and sharp features. For the Delaunay type methods, the convergence of the boundary fitting procedures has not been proven. We will use the advancing-front technique as it is applicable to arbitrary solids and allows us to control shapes and sizes of tetrahedrons during the mesh generation process. In addition, we consider a modification of this technique for increasing the effectiveness of the tetrahedrization procedure for FRep solids. 3D mesh optimization procedures are described in [5, 6, 30, 17, 18].

## 4. GENERAL ALGORITHM OF DISCRETIZATION

In this section we provide a systematic description of our discretization algorithm.

### 4.1 General description

In principle, there are two main strategies to discretize a heterogeneous object with generation of a volume mesh. The first strategy involves decomposing an initial 3D object into 3D elements (tetrahedrons, blocks, prisms, some combined volume mesh) that are optimised under requirements and constraints described in 2.4. Here, the boundary mesh  $\tilde{B}$  appears as a side effect of the 3D initial object decomposition. Another approach implies that first we decompose the surface  $B$  of the initial object thus yielding the surface mesh  $\tilde{B}$ . Then, this surface mesh is subjected to optimisation and refinement to make sure that it satisfies all the possible requirements following which one can build a volume mesh conformable to the refined surface mesh.

In our work, we follow the second approach, because most of the constraints and requirements given in 2.4 deal with the boundary mesh  $\tilde{B}$  whose quality is crucial in the context of FEA. As some of the constraints contradict each other, it is important to ensure that all the accessible iterative optimisation procedures are performed to provide the best possible result. In addition, it is known [8] that some effective methods of volume mesh generation are actually based on boundary descriptions of computational domains.

So, we decompose the discretization problem into two relatively independent sub-tasks:

- generation of a mesh of the object surface along with its refinement:  $\tilde{B}_c = \tilde{B}_c(B_F, A_1, \dots, A_k)$ ;
- generation of a conformable volume mesh:  $\tilde{G}_c = \tilde{G}_c(\tilde{B}_c, G_F, A_1, \dots, A_k)$ . We use tetrahedral meshes as we consider them the most universal: they are successfully used both in visualization and in FEA, and they often serve as a base for building meshes consisting of more complex patterns.

### 4.2 Surface discretization

Here, we describe how the first task of generating the quality boundary mesh approximating the initial object surface can be solved in step-by-step manner.

#### 4.2.1 Polygonal approximation of the object surface

We form a simplicial complex  $L_0^2 = L_0^2(B_F)$  representing (in accordance with 2.3) an approximate CRep based surface model  $\tilde{B}_{c0}$ . Requirement 1 from 2.4 should be satisfied, and the subsequent steps on the surface mesh reconstruction are such that they preserve the surface topology. We use a polygonization algorithm described in [29], which solves a problem of topological ambiguities on the faces with four edge-surface intersection points peculiar to the second group of polygonization methods described in 3.2. We assume that the user can control the preservation of topology equivalence between the initial implicit surface  $B_F$  and the resulting polygonal (triangular) surface  $\tilde{B}_{c0}$  by providing proper initial data necessary for polygonization, namely the surface bounding box and the grid resolution.

#### 4.2.2 Sharp features reconstruction

This step deals with optimization of the surface mesh  $\tilde{B}_{c0}$  thus yielding a new CRep based model  $\tilde{B}_{c1}$ , described by the complex  $L_1^2 = L_1^2(L_0^2, B_F)$ . In this model, sharp edges and corners present in the initial surface  $B_F$  are described by the conformable 0D and 1D elements in the complex  $L_1^2$ . Accordingly, the requirements 1 and 2 from 2.4 are satisfied for the cellular model  $\tilde{B}_{c1}$ . To extract sharp features from an implicit surface coarse triangulation, we use the algorithm described in [24]. This algorithm is based on combining the application of the following mesh optimization procedures:

- curvature-weighted vertices resampling;
- dual/primal mesh optimization that involves projecting the triangle centroids onto the implicit surface and moving each vertex of the initial surface triangulation to a new position minimizing the sum of the squared distances from the vertex to the planes which are tangent to the implicit surface at the projections of adjacent triangles centroids.
- one-to-four subdivision of mesh triangles where the mesh normals have large deviations from implicit surface normals.

Our experience with application of this optimization algorithm shows that in most cases it does produce surface mesh  $\tilde{B}_{c1}$  allowing for a quality representation of sharp features of the initial implicit surface. However, the resulting mesh can have badly shaped or degenerate triangles near sharp edges and corners and may consist of an excessive number of triangles produced by the adaptive subdivision procedure in the regions of low curvature. Consequently, we propose further refinement described in the following subsections.

#### 4.2.3 Surface mesh refinement and simplification

The objective of this phase is to get rid of badly shaped triangles, to make the mesh finer in the regions of high surface curvature, and, on the contrary, to make the mesh coarser in the areas where the curvature is low. The requirement for preserving sharp

features should be satisfied. As an ideal result, the surfaces of the tetrahedron and cube should be represented by just four and twelve triangles respectively.

Thus we aim at building CRep based model  $\tilde{B}_{c2}$  described by a complex  $L_2^2 = L_2^2(L_1^2, B_F)$ , which is obtained as the result of optimization of the complex  $L_1^2$ . The optimization procedure consists in iterative application of edge swapping, edge splitting, edge collapsing and vertex relocation operations. Edge splitting allows for enriching the mesh, edge collapsing provides mesh simplification, and edge swapping and vertex relocation operations are used for mesh refinement. We describe in section 5 the main characteristics of these operations as well as the criteria for their application. In the end, the model  $\tilde{B}_{c2}$  described by the complex  $L_2^2$  must ensure a quality polygonal approximation of the implicit surface  $B_F$  under the requirements 1 – 4 from the subsection 2.4.

#### 4.2.4 Surface mesh adaptation

At this step, we aim at adapting the mesh to the needs of FEA in the context of some particular application, thus producing CRep based surface model  $\tilde{B}_{c4}$ . This means that the requirements 4 – 6 from section 2.4 should be satisfied. First, singular lines and points of attribute functions should be taken into account. We project those lines and points onto the discretized surface  $\tilde{B}_{c2}$  obtained at the previous step, and then make a partition of elements of the complex  $L_2^2$ , thus yielding a new complex

$L_3^2 = L_3^2(L_2^2, B_F, A_1, \dots, A_k)$  in which the singular lines and points are described by some 1D and 0D subcomplexes. After this procedure, it is useful to execute the previous step once again to preserve not only geometric sharp features but also the singular lines and points of the attribute functions. Then, we make the adaptive mesh subdivision and refinement to satisfy the constraints 4 – 6 through using once again edge splitting, edge swapping, and vertex relocation operations as well as the triangle subdivision procedure. Note that the criteria for application of those procedures does depend on both the attribute functions  $S_i(X)$  and the function  $F(X)$  describing geometry (more details are given in Section 5). Finally, we get the complex  $L_4^2 = L_4^2(L_3^2, B_F, A_1, \dots, A_k)$  describing the surface model  $\tilde{B}_{c4}$  that hopefully meets all the constraints and requirements important for FE meshes.

### 4.3 3D object discretization

The second task of building a quality volume mesh based on the surface mesh generated in the previous stage is described in this section.

#### 4.3.1 Tetrahedral mesh generation

Given the CRep based model of the surface mesh  $\tilde{B}_c = \tilde{B}_{c4}$ , described by the complex  $L_c^2 = L_4^2$  we can now build a CRep based model of the 3D object  $\tilde{G}_{c0} = \tilde{G}_{c0}(\tilde{B}_{c4}, G_F, A_1, \dots, A_k)$ . To

produce such a volume mesh, we use the advancing front method briefly characterized in 3.3. More details about our modification of that algorithm are given in subsection 5.2. The model  $\tilde{G}_{c0}$  is described by a 3D complex  $K^3_\theta$ , which is built on the basis of the boundary complex  $L_C$  and on the initial object attributes. So, we have  $K^3_0 = K^3_0(L^2_c, A_1, \dots, A_k)$ . While generating such a volume mesh, the constraints 4 - 6 from 2.4 are taken into account. However, because of the convergence problems discussed in 3.3, not all of those restrictions can immediately be satisfied. So, additional post-remeshing may be needed to get a quality tetrahedral mesh.

### 4.3.2 Volume mesh adaptation

The objective of this phase is to re-build the volume mesh described by the complex  $K^3_\theta$  that allows for improving the tetrahedra shapes under all the requirements for the mesh related to attributes. To succeed in solving this problem, we use face swapping, tetrahedra subdivision and vertex relocation operations ([6], [30], [17], [18]). Some features of the implementation of these operations are considered in 5.2 Note that once again we pay attention to preserving sharp features, and singular points and lines of the boundary surface. As a result, we get a new complex  $K^3_1 = K^3_1(K^3_1, L^2_c, A_1, \dots, A_k)$  defining a CRep based model  $\tilde{G}_{c1}$  that should satisfy all the requirements critical for FE meshes.

### 4.3.3 Attribute transformations

The final step in discretization of heterogeneous objects is concerned with converting initial attributes for getting new ones:  $A_j = A_j(K^3_1, L^2_c, A_1, \dots, A_k)$ ,  $j=1, 2, \dots, m$ . As we discussed in 2.4, some attributes present in the initial model can disappear, others may be described by another representation scheme, and new attributes can appear. The attribute conversion procedures can heavily depend on application specifics. Eventually, we get the sought-for discrete model  $\tilde{D} = (\tilde{G}_c = \tilde{G}_{c1}, \tilde{A}_1, \dots, \tilde{A}_m)$  of the initial heterogeneous object  $D = (G_F, A_1, \dots, A_k)$  with CRep based geometry, and CRep and CFRep based attributes.

## 5. DETAILS OF BASIC PROCEDURES

In this section we present some operations and procedures in more significant detail.

### 5.1 Surface mesh optimization

Here we consider in detail the basic mesh refinement operations introduced in 4.2.3. These operations are iteratively applied to the current mesh in order to provide an accurate piecewise linear approximation of the underlying implicit surface and to adapt the element sizes to FEA requirements given in 2.4. In particular, it is necessary to preserve sharp edges and corners of the surface. For this purpose, we introduce so-called sharp function  $Sh$  for the mesh edges and vertices. Using the planarity estimations [8] for each edge  $e$ , we set:

$$Sh(e)=1, \text{ if } \frac{1+(\bar{n}^T(t1), \bar{n}(t2))}{2} \leq \varphi \text{ and } \frac{1+(\bar{N}^T(t1), \bar{N}(t2))}{2} \leq \varphi$$

$Sh(e)=0$ , otherwise.

Here  $\bar{n}(t1), \bar{n}(t2)$  are unit normals of the adjacent triangles  $t1, t2$  correspondingly;  $\bar{N}(t1), \bar{N}(t2)$  are the implicit surface normals at the central points of  $t1, t2$ ;  $\varphi$  is a threshold that measures the sharpness of a feature.

The implicit surface normals are calculated as the normalized gradients of the function  $F(X)$  describing the implicit surface. Note that a nonzero value of  $Sh(e)$  indicates that the edge  $e$  probably lies on a sharp feature of the surface. For the vertices, the sharp function is defined as follows:

$$Sh(P) = \sum_i Sh(e_i),$$

where  $e_i$  is an edge incident to the vertex  $P$ . If  $Sh(P)=2$  is true, then vertex  $P$  lies on a sharp edge, but the case  $Sh(P)>2$  corresponds to a corner point.

To single out those spikes that do not lie on sharp edges, we use the heuristic estimation proposed in [14].

$$\min_i n(\bar{m}^T, \bar{n}(ti)) \leq \tau \ \& \ \min_i n(\bar{M}^T, \bar{N}(t1)) \leq \tau.$$

Here  $\bar{n}(ti)$  are unit normals of triangles  $ti$  adjacent to point  $P$ ,  $\bar{N}(ti)$  are the implicit surface normals at the central points of  $ti$ ,  $\bar{m} = [\bar{n}_{t0} \times \bar{n}_{t1}]$  is the normal vector to the plane spanned by two normals  $\bar{n}_{t0}, \bar{n}_{t1}$  which enclose the largest angle, and similarly  $\bar{M} = [\bar{N}(t0) \times \bar{N}(t1)]$ ,  $\tau$  is a threshold.

For the vertices which are found in such a way, we set  $Sh(P)=3$ .

Let us consider all the basic mesh refinement operations in more detail.

#### 5.1.1 Mesh vertices relocation

This operation consists in application of two procedures: mesh smoothing and vertex moving.

*Mesh smoothing* improves regularity of elements' size. If  $\bar{V}_i$  is a non-sharp vertex (i.e.,  $Sh(V_i)=0$ ), then its relocation is described as:

$$(\bar{V}_i)_{new} = (\bar{V}_i)_{old} + \lambda \bar{U}_i / \|\bar{U}_i\|.$$

Here  $\lambda$  is a small positive number ( $\lambda < l_{min}$ , where  $l_{min}$  is the smallest length of all edges incident to the point) that limits displacement of vertices, and taking it into account prevents the appearance of such mesh defects as folds and self-intersection.

The moving direction  $\bar{U}_i$  is defined by the formula [23]:

$$\bar{U}_i = \bar{R}_i - (\bar{R}_i^T \bar{N}_i) \bar{N}_i,$$

where  $\bar{N}_i$  is the surface normal at the point  $(\bar{V}_i)_{old}$ ,

$$\bar{R}_i = \frac{1}{\sum_j w_j} \sum_j w_j \bar{P}_j - (\bar{V}_i)_{old}, \quad w_i = \frac{1}{\|\bar{P}_j - (\bar{V}_i)_{old}\|},$$

$\bar{P}_j$  is a vertex having a common edge with  $\bar{V}_i$ .

If the vertex  $\bar{V}_i$  is a corner or belongs to more than two sharp edges, then its position does not change. However, if the vertex  $\bar{V}_i$  has exactly two incident sharp edges, for example,

$$e_1 = (\bar{V}_i, \bar{P}_1), e_2 = (\bar{V}_i, \bar{P}_2) \text{ and } \frac{1 + (\bar{N}^T(e_1), \bar{N}(e_2))}{2} \geq \sigma, \text{ where}$$

$\bar{N}(e_1), \bar{N}(e_2)$  are the surface normals at the centers of edges  $e_1, e_2$ ,  $\sigma$  is a threshold, then a new point  $(\bar{V}_i)_{new}$  is placed at the center of the arc connecting vertices  $\bar{V}_i, \bar{P}_1, \bar{P}_2$ .

One should pay attention that during the mesh smoothing process some vertices can be detached from the initial surface; therefore, their positions have to be corrected. The *vertex moving* procedure moves vertices towards the underlying implicit surface. It is organized as an optimal search process in accordance with the formula:

$$(\bar{V}_i)_{new} = (\bar{V}_i)_{old} + r^* \bar{Z}_i / \|\bar{Z}_i\|,$$

$$\text{where } r^* = \arg \min_{r \in [0, \lambda]} F^2((\bar{V}_i)_{old} + r \bar{Z}_i / \|\bar{Z}_i\|)$$

$$\text{with } \bar{Z}_i = -2F((\bar{V}_i)_{old}) \nabla F((\bar{V}_i)_{old}).$$

### 5.1.2 Edge swapping

This operation is used to improve the mesh elements' shape quality. Let us consider an illustrative example in Fig. 1. We eliminate the diagonal  $(P_1, P_2)$  and instead insert the diagonal  $(P_3, P_4)$  if  $\max(\beta_1, \beta_2) < \max(\alpha_1, \alpha_2)$ .

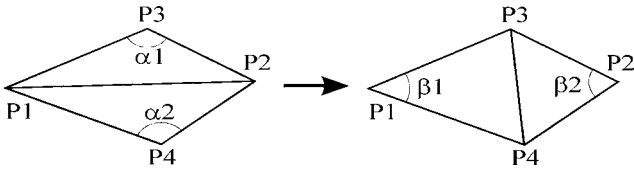


Figure 1: Edge swapping

This operation is applied only if adjacent triangles sharing the edge  $e$  are almost coplanar, so that

$$\frac{1 + (\bar{n}^T(t_1), \bar{n}(t_2))}{2} \geq \sigma_n, \text{ and } \frac{1 + (\bar{N}^T(t_1), \bar{N}(t_2))}{2} \geq \sigma_N,$$

where the constants  $\sigma_N$  and  $\sigma_n$  are user-specified thresholds.

### 5.1.3 Edge splitting

This operation is performed for a number of reasons: to improve geometry approximation, to fit attribute functions or to subdivide badly shaped triangles. To describe criteria for choosing those edges which are suitable for splitting, we introduce the following denotations. Let  $W(X)$  be a scalar characteristic function, and  $\bar{D}(X)$  be a vector one. Then, in the context of the optimization procedure of geometry approximation, these functions depend on the function  $F(X)$  describing geometry of the initial object and on the gradient of  $F(X)$ :

$$W(X) = F^2(X)$$

$$\bar{D}(X) = \nabla F(X) / \|\nabla F(X)\|.$$

In the process of the finite element mesh adaptation, these characteristic functions are defined with respect to the scalar and vector attribute functions:

$$W(X) = W(S_1(X), \dots, S_k(X))$$

$$\bar{D}(X) = \bar{D}(S_1(X), \dots, S_k(X)).$$

In order to control the quality of triangles, we use the finite element shape quality measure [8]:

$$q(T) = \alpha \frac{l_{max}}{r_T},$$

where  $r_k$  denotes the inradius of the triangle  $T$ ,  $l_{max}$  is the largest edge length, and  $\alpha$  is a normalization coefficient so that  $q(T)=1$  for an equilateral triangle.

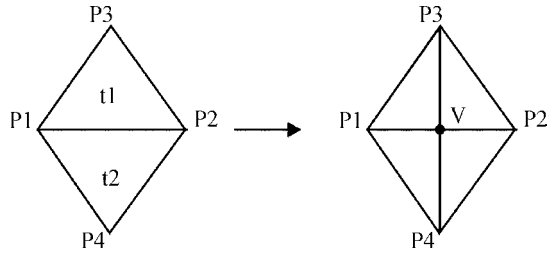


Figure 2: Edge splitting

Our edge splitting operation is illustrated by Fig. 2. The process of mesh subdivision is iterative, and edges are processed in the order of decreasing their lengths. We use the following estimations to measure value variations of the characteristic functions:

$$\tau_1 = \text{length}(e) / S_r(C)$$

$$\tau_2 = \max(q(T_1), q(T_2))$$

$$\varepsilon_1 = \max(|W(C) - W(P_1)|, |W(C) - W(P_2)|)$$

$$\varepsilon_2 = \max(|W(C_1) - W(P_1)|, |W(C_1) - W(P_2)|, |W(C_1) - W(P_3)|)$$

$$\varepsilon_3 = \max(|W(C_2) - W(P_1)|, |W(C_2) - W(P_2)|, |W(C_2) - W(P_4)|)$$

$$\nu_1 = 1 - \min_{i,j \in \{1,2,3,4\}} (\bar{D}^T(P_i), \bar{D}(P_j))$$

$$\nu_2 = \frac{(\bar{N}^T(P_1), \bar{P}_1 \bar{P}_2) + (\bar{N}^T(P_2), \bar{P}_1 \bar{P}_2)}{\|\bar{P}_1, \bar{P}_2\|^2}$$

$$\nu_3 = \max((1 - |(\bar{N}^T(C_1), \bar{n}(T_1))|), (1 - |(\bar{N}^T(C_2), \bar{n}(T_2))|))$$

where  $C$  is a center of an edge, and  $C_1, C_2$  are centroids of adjacent triangles,  $S_r$  is a mesh density attribute function,  $\bar{N}$  denotes surface normals and  $\bar{n}$  are normals of triangles.

The edge  $e$  is subdivided if one of the listed estimations or several of them exceed the user-specified thresholds. The split vertex  $V$  is placed at the position

$$V = P_1 + \frac{S_r(P_1)}{S_r(P_1) + S_r(P_2)} \overline{P_1 P_2},$$

where  $S_r(X)$  is a mesh density attribute function. Then,  $V$  is moved towards the implicit surface according to the procedure described in 5.1.1.

Different characteristic functions and estimations can be used at the different phases of mesh optimization and adaptation process. After a few iterations, it is recommended to execute edge swapping and mesh smoothing operations which can improve the mesh quality.

For quick mesh subdivision in the cases where an average mesh element size is rather greater than the desired finite element size, we use one-to-four and one-to-three *triangle subdivision* coupled with mesh smoothing and edge swapping.

### 5.1.4 Edge collapsing

This operation is used to simplify the surface mesh in low curvature regions. We choose an edge  $e$  and replace it with a vertex  $V$  (see Fig. 3a). In this process, two vertices  $P_1$ ,  $P_2$  are substituted with a new vertex  $V$ , and the triangles  $T_1$ ,  $T_2$  are collapsed to edges.

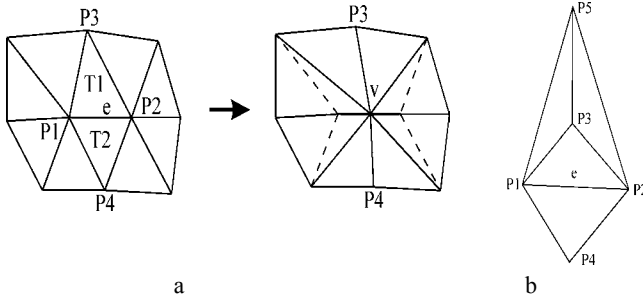


Figure 3: Edge collapsing

We check for collapse candidacy edges in the order of increasing length. The process is as follows. First, we check the topological validity of the edge collapsing operation for each edge. Edge  $e$  can be substituted with a new vertex if the following condition is satisfied:

$$Star(P_1) \cap Star(P_2) = \{P_3, P_4\},$$

where  $Star(P_i)$  is the set of the mesh vertices sharing a common edge with the vertex  $P_i$ . This restriction allows for avoiding mesh degradation in such cases as shown in Fig.3b.

Secondly, if  $length(e) \leq \sigma_{min}$ , then the edge  $e$  is collapsed without need for any additional analysis, and the midpoint of  $e$  is chosen as the position of a new vertex  $V$ . Otherwise, we continue checking  $e$  for a candidacy edge. For this purpose, we calculate so-called weight coefficients  $w(P_1)$ ,  $w(P_2)$  for the edge vertices. We set  $w(P_i)=1$ , if  $Sh(P_i)=0$ .

The values  $Sh(P_i)=1$  or  $Sh(P_i) > 2$  indicate that the vertex  $P_i$  is a spike or a corner, so we set  $w(P_i)=0$  in this case. From  $(Sh(P_i)=2 \& Sh(e)=0)$  it follows that there exist other sharp edges incident to  $P_i$ , so we set  $w(P_i)=0$ . But if  $e$  is one of the two sharp edges incident to  $P_i$  (i.e.  $Sh(pi)=2 \& Sh(e)=1$ ), then we should calculate the angle  $\alpha$  between these edges. If  $cos(\alpha) < 0$  and  $(1 - |cos(\alpha)|) \leq \sigma_\alpha$ , then we set  $w(P_i)=0.5$ , otherwise  $w(P_i)=0$ .

Let us formulate the conditions under which one can consider execution of the edge collapsing operation. The possibility of edge collapsing and the position of a new vertex  $V$  depends on the edge vertices weights as follows:

if  $w(P_1)=w(P_2)=0$ , then the edge  $e$  cannot be collapsed;

if  $w(P_1)=w(P_2)>0$ , then the edge  $e$  can be substituted with the vertex  $V$  placed at the midpoint of  $e$ ;

if  $w(P_1) > w(P_2) \geq 0$ , then the edge  $e$  can be substituted with the vertex  $V = P_2$ ;

if  $w(P_2) > w(P_1) \geq 0$ , then the edge  $e$  can be substituted with the vertex  $V = P_1$ .

Then we move the new vertex  $V$  onto the implicit surface using the procedure described in 5.1.1. Finally, we evaluate the quality degradation during the intended collapse operation using the following measures:

$$\theta_1 = \max_{T_i=inc(V)} ((1 - |\overline{N}^T(C_i), \overline{n}(T_i)|))$$

$$\theta_2 = \max_{P_j \in Star(V)} \frac{2(\overline{N}^T(V), \overline{VP_j})}{\|\overline{VP_j}\|^2}$$

Here,  $\overline{N}$  denotes normals of the initial surface,  $\overline{n}$  is for normals of triangles,  $C_i$  is a centroid of the triangle  $T_i$ , and  $T_i=inc(V)$  is for triangles incident to  $V$ .

If  $\theta_1 \leq \sigma_t$  and  $\theta_2 \leq \sigma_n$ , then the corresponding edge  $e$  is collapsed and substituted with the vertex  $V$ . Constants  $\sigma_{min}$ ,  $\sigma_\alpha$ ,  $\sigma_t$ , and  $\sigma_n$  are user-specified thresholds. It is recommended to couple the edge collapsing operation with mesh smoothing and edge swapping.

## 5.2 Volume mesh generation and optimization

### 5.2.1 The modification of the advancing front method

First, let us characterize some critical steps of the advancing front algorithm (briefly outlined in 3.3) as applied to tetrahedrization. A surface mesh described by a 2D complex forms the initial front. Then, at each step of the algorithm an *active face* is selected to serve as a base for building a new tetrahedron whose size is calculated taking into account the given mesh density attribute. Provided the new tetrahedron does not cross the front, it is added to the front thus forming a new one. To avoid emergence of thin elements, a new tetrahedron vertex is placed in the closest node of the mesh satisfying the given distance threshold. There are differing strategies [8] for selection of both the next active face and new tetrahedron node that prevent the procedure from generating an infinite series of tetrahedrons yet yield a volume mesh of reasonable quality (e.g., without thin elements, etc.).

We propose modification of the advancing front method that takes advantage of the fact that we deal with the problem within the cellular-functional framework. In particular, availability of an exact functional description of the object to be subdivided can greatly simplify the procedure of the evaluating point membership relation which is important in the context of determining whether a tetrahedron crosses the front.

Now, let us outline the modified algorithm. We start from the CRep based model of the surface mesh  $\tilde{B}_c$  represented by the simplicial complex  $L_c^2$  and the function  $S_r(X)$  describing the mesh density attribute  $A_r$ . First, a regular tetrahedral mesh covering the object is generated. This mesh is described by a simplicial complex  $M^3_0$ . Then, elements of  $M^3_0$  are subdivided to conform to the mesh density attribute  $A_r$ . For this purpose, 3D mesh optimization operations (see 5.2.2) are applied. So, the modified mesh is represented by the simplicial complex  $M^3_1 = M^3_1(M^3_0, A_r)$ .

At the next step, we single out the subcomplex  $M^3_2$  lying completely inside the subdivided object  $G_F$ . So  $M^3_2 \subseteq M^3_1$ ,  $|M^3_2| \subseteq G$ . To ensure the convergence of the algorithm, the distance between the boundary surface of the object and the mesh described by the complex  $M^3_2$  must be a few times greater than the tetrahedron size described by the function  $S_r(X)$ . Therefore,  $M^3_2$  includes only those elements of  $M^3_1$  whose nodes  $V_i$  satisfy the following condition:

$$F(V_i) - k * S_r(V_i) \geq 0, \text{ where } k \geq 2$$

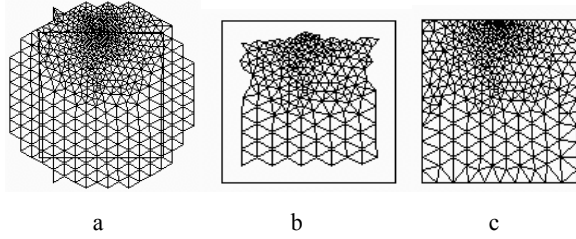


Figure 4: Modified advancing front method: a) the initial rectangular object with the pattern mesh subdivided according to a mesh density attribute b) the sub-mesh lying completely inside the subdivided object c) the final mesh

Then we define a two-dimensional complex  $Q^2_2$  that represents the boundary of  $M^3_2$ . This complex together with the complex  $L_c^2$  describing the discretization of the boundary surface form the front  $Q^2_3 = L_c^2 \cup Q^2_2$ . Using this front, we subdivide the remaining sub-area with help of the advancing front tetrahedrization method and thus get the 3D complex  $M^3_3 = M^3_3(Q^2_3, S_r(X))$ . The overall complex  $K^3_0$  defining a CRep based model  $\tilde{G}_{c1}$  is calculated as  $K^3_0 = M^3_2 \cup M^3_3$ . Then, we apply a few iterations of 3D mesh optimization to the complex  $K^3_0$  to ensure that we get a volume mesh of good quality and regularity, and that all the FE related constraints are satisfied. As a result we get a new complex  $K^3_1 = K^3_1(K^3_0, L_c, A_1, \dots, A_k)$ .

Fig. 4 shows a 2D illustration of the described algorithm. The initial rectangular object with the pattern mesh subdivided according to a mesh density attribute  $S_r(X)$  is shown in Fig 4a. Fig. 4b illustrates the sub-mesh lying completely inside the subdivided object. The area between the initial boundary and that sub-mesh is subdivided using the advancing front method. The final mesh is shown in Fig 4c.

Our modified frontal method is more effective in cases when a considerable part of the object can be covered by pattern mesh elements that remain unchangeable. In these cases we reduce the number of elements generated by the frontal tetrahedrization and keep well shaped elements of the pattern mesh. Moreover our modified algorithm allows for combination of meshes of different

types, for example, the initial mesh covering the object can consist of hexahedral elements.

### 5.2.2 Volume mesh optimization operations

The 3D mesh optimization algorithm consists in the iterative application of face swapping, tetrahedral splitting and vertex repositioning techniques.

- *The face swapping operation* is the 3D extension of the edge swapping technique described in 5.1.2. Two neighboring tetrahedra having a common face are transformed with the exchange of the common diagonal.
- *The tetrahedra splitting operation* includes one-to-two subdivision by bisecting the longest edge. The selection of a tetrahedron to be split is made on the basis of criteria similar to those described in 5.1.2 for the edge splitting operation for a surface mesh.
- *The vertex relocation technique* in 3D consists in the movement of a node towards the barycenter of the polyhedron formed by the surrounding tetrahedral mesh. Since the surrounding polyhedron can be non-convex, the positioning of the node directly at the barycenter can result in overlapping tetrahedra. This is the reason for using an adaptive procedure with a variable step of the movement to the barycenter.

More general three dimensional versions of swap and split operators remesh a polyhedron formed by neighbouring tetrahedra sharing a common edge or a common vertex [8].

## 6. EXAMPLES

In this section we present several examples illustrating our algorithm for discretization of functionally based heterogeneous objects. The examples were prepared using our original software tools initially intended for data preprocessing in computational physics and allowing for the users' interactive work in a step-by-step manner. We used HyperFun modeling language [28] to define all the models, and specialized software tools with the built-in HyperFun interpreter have been used to implement the examples on a Pentium III 800 MHz computer.

### 6.1 Tetrahedrization of FRep solids

Here we illustrate the main steps of the algorithm using the example of an FRep object with sharp features shown in Figure 5. The initial surface mesh produced by polygonization is shown in Fig 5a. The surface mesh generated in the process of sharp features reconstruction is shown in Fig. 5b. For the purpose of FEA, this mesh has badly shaped and degenerate triangles near sharp edges and corners, and includes an excessive number of triangles produced by the adaptive subdivision procedure in the regions of low curvature. Subsequent mesh optimization and simplification produces the minimal surface triangulation presented in Fig. 5c. This triangulation then serves as an initial front while applying our advancing front algorithm. Fig. 5d shows the resulting tetrahedrization.

Tetrahedrization of another FRep object having more complex topology is presented in Fig. 6. It is a typical CSG-like object with sharp edges and both flat and curvilinear faces. The result of polygonization and sharp features extraction is shown in Fig. 6a.

The mesh generated is not completely suitable for subsequent FEA: one can observe badly shaped and degenerate triangles near sharp edges and corners in the surface mesh after the sharp features extraction. The surface mesh adaptation to FEA requirements after applying mesh optimization procedures is shown in Fig. 6b. An enlarged view of an internal structure of the tetrahedral mesh is shown in Fig. 6c. As to computational times, mesh decimation process took 4.2 sec, FE mesh adaptation took 2.9 sec, and tetrahedrization took 331 sec.

## 6.2 Discretization of a heterogeneous object with various attributes

This example illustrates the influence of attributes on the mesh elements' sizes. Fig. 7 presents four discretization variants conforming to different attributes. We start from Fig. 7b where no attribute has an effect. Note that the minimal surface triangulation for this object was created using our mesh optimization and simplification algorithm similar to the previous example. Then, this surface mesh was subdivided to conform to the attributes. Figs. 7c-7e illustrate application of various types of attribute functions.

Fig. 7c shows the result of subdivision based on an attribute influencing the mesh density and defined by a "point source" placed at a cube corner. Figure 7d simulates mesh adaptation near the vertical well, where the element sizes are adjusted to the reservoir pressure gradient. An example of a mesh fitted to an annular heating device is shown in Fig. 7e. Mesh adaptation process in all these examples took no longer than 7 sec.

## 6.3 Modeling of a mixing tank impeller

An application example shown in Fig. 8 is concerned with real computer-aided modeling of a mixing tank impeller which is used as the agitator of the flow components in a chemical reactor.

The impeller whose central body and blades are made of different materials (marked by different colors) is shown in Fig. 8a. Fig. 8b shows the surface mesh with recovered sharp features. One can observe that there are many badly shaped triangles in the mesh, so the following mesh optimization is necessary. Fig. 8c illustrates the result of such optimization. The surface mesh taking into account the material attributes is shown in Fig. 8d. The mesh was split on the blades' faces according to the specified attribute. This surface mesh then serves as initial data for the advancing front tetrahedrization. The resulting 3D mesh was used for FE thermal and stress-strain analysis. As to calculation times, mesh decimation took 3.1 sec, FE mesh adaptation took 10 sec, mesh adaptation according to the attribute took 27 sec, and tetrahedrization took 634 sec.

## 7. CONCLUSION

This work aims at making functionally defined solids and heterogeneous objects with implicit surfaces available for practical applications requiring finite element analysis and simulation. We discussed a discretization procedure resulting in surface and volume meshes for heterogeneous objects with geometry and attributes defined using real-valued functions. This procedure is considered as an implementation of the functional to cellular models conversion operation in the cellular-functional modeling framework for heterogeneous objects [1].

In contrast to previous works on implicit surface polygonization and volume mesh generation, the main motivation of this work is generation of meshes suitable for finite element analysis with constraints imposed by the heterogeneous object attributes. Let us summarize the main contributions of the paper:

- 1) The discretization problem is stated for heterogeneous objects represented as functionally defined 3D solids (with implicit surfaces) and attributes (scalar fields).
- 2) The paper aggregates different techniques into a systematic step-by-step procedure of solving the above problem. Some of the known techniques have been adapted and extended to work with the functionally defined solids as explained below.
- 3) The proposed surface mesh optimization preserves sharp features of implicit surfaces, satisfies the requirements of FEA, and uses the defining function to do this (see the mesh transformation from Fig. 5b to 5c).
- 4) The advancing front method of 3D tetrahedrization is extended taking the defining function of the solid into account.

We provided examples of tetrahedrization of objects with complex topology and sharp features, and illustrated mesh adaptation to attributes of various types. A practical example of FEM generation for a mixing tank impeller was given. All examples have been prepared using software tools developed by the authors.

In the cellular-functional model [1], heterogeneous objects are represented as hypervolumes or multidimensional point sets with multiple attributes. A multidimensional point sets can include elements of different dimensions, which can be higher than three. In this paper, we dealt only with 3d objects with attributes. The extension of the proposed algorithms to the dimensionally heterogeneous and time-dependent objects is a subject of future work.

## 8. ACKNOWLEDGMENTS

The authors would like to thank Drs. Y. Ohtake and A. Belyaev for fruitful discussions and for the possibility to work with their implementation of the dynamic meshes algorithm. Jody Vilbrandt has greatly helped with making the text more reader friendly.

## 9. REFERENCES

- [1] Adzhiev, V., Kartasheva, E., Kunii, T., Pasko, A., Schmitt, B. Cellular-functional modeling of heterogeneous objects, Proc. ACM Solid Modeling and Applications 2002 Symposium, Saarbrücken, Germany, Ed. Kunwoo Lee, N. Patrikalakis, ACM Press, 2002, 192-203
- [2] Biswas, A., Shapiro, V., Tsukanov, I. Heterogeneous material modeling with distance fields, Technical Report SAL 2002-4, University of Wisconsin-Madison, June, 2002.
- [3] Bloomenthal, J. An Implicit Surface Polygonizer, Graphics Gems IV, Academic Press, 1994.
- [4] Chen, M., Tucker, J. Constructive volume geometry, Computer Graphics Forum, 19(4), 2000, 281-293.
- [5] Frank, K., Lang, U. Gradient and curvature approximation in data-dependent surface simplification, Computing and Visualization in Science, 2(4), 2000, 221-228

- [6] Freitag, L., Ollivier-Gooch, C. Tetrahedral mesh improvement using swapping and smoothing, *Int. J. Numer. Meth. Eng.*, 40, 1997, 3937-4002.
- [7] Frey, P.J., Borouchaki, H., Geometric surface mesh optimization, *Computing and Visualization in Science*, 1(3), 1998, 113-121
- [8] Frey, P.J., George, P.-L. Mesh Generation: Application to Finite Elements - HERMES Science Europe, OXFORD & PARIS, 2000, 814p.
- [9] Garland M., Heckbert, P.S. Surface simplification using error metrics, *Computer Graphics (Proceedings of SIGGRAPH'97)*, 209-216.
- [10] Hartmann, E. A marching method for the triangulation of surfaces, *The Visual Computer*, 14(3), 1998, 95-108.
- [11] Jackson, T. R., Liu, H., Patrikalakis, N. M., Sachs, E. M., and Cima, M. J. Modeling and designing functionally graded material components for fabrication with local composition Control, *Materials and Design*, 20(2/3), 1999, 63-75.
- [12] Karkanis, T., Stewart, A. J. Curvature-dependent triangulation of implicit surfaces, *IEEE Computer Graphics and Applications*, 21(2), 2001, 60-69.
- [13] Kobbelt, L.  $\sqrt{3}$ -Subdivision, *Computer Graphics (Proceedings of SIGGRAPH'2000)*, 103-112.
- [14] Kobbelt, L., Botsch, M., Schwanecke, U., Seidel, H.-P. Feature sensitive surface extraction from volume data, *Computer Graphics (Proceedings of SIGGRAPH 2001)*, 57-66.
- [15] Kumar, V., Dutta, D. An approach to modeling multi-material objects, *Fourth Symposium on Solid Modeling and Applications*, ACM SIGGRAPH, 1997, 336-345.
- [16] Kumar, V., Burns, D., Dutta, D., Hoffmann, C. A framework for object modeling, *Computer-Aided Design*, 31(9), 1999, 541-556.
- [17] Liu, A., Joe, B. On the shape of tetrahedra from bisection, *Math. Comp.*, 63, 1994, 141-154.
- [18] Liu, A., Joe, B. Quality local refinement of tetrahedral meshes based on bisection, *SIAM J. Sci. Comput.*, 16, 1995, 1269-1291.
- [19] Lohner, R. Automatic unstructured grid generators, *Finite Elements in Analysis and Design*, 25, 1997, 114-134
- [20] Lorensen, W., Cline, H. Marching Cubes: a high resolution 3D surface construction algorithm, *Computer Graphics*, 21(4), 1987, 163-169.
- [21] Martin, W., Cohen, E. Representation and extraction of volumetric attributes using trivariate splines: a mathematical framework, *Sixth ACM Symposium on Solid Modeling and Applications*, D. Anderson, K. Lee (Eds.), ACM Press, 2001, 234-240.
- [22] Nielson, G., Hamann, B. The asymptotic decider: resolving the ambiguity in Marching Cubes, *Proceedings of Visualization '91*, IEEE Computer Society Press, October, 1991, 29-38.
- [23] Ohtake, Y., Belyaev, A., Pasko, A. Dynamic meshes for accurate polygonization of implicit surfaces with sharp features, *Shape Modeling International 2001*, IEEE Computer Society, 2001, 74-81.
- [24] Ohtake Y., Belyaev A., Dual/primal mesh optimization for polygonized implicit surfaces, *ACM Solid Modeling*, K. Lee and N. Patrikalakis (Eds.), ACM Press, 2002, 171-178.
- [25] Owen, S. J., A survey of unstructured mesh generation technology, *Proceedings 7th International Meshing Roundtable*, Dearborn, MI, October 1998
- [26] Park, S. M., Crawford, R., Beaman, J. Volumetric multi-texturing for functionally gradient material representation, *Sixth ACM Symposium on Solid Modeling and Applications*, D. Anderson, K. Lee (Eds.), ACM Press, 2001, 216-224.
- [27] Pasko, A., Adzhiev, V., Sourin, A., Savchenko, V. Function representation in geometric modelling: concepts, implementation and applications, *The Visual Computer*, 11(8), 1995, 429-446.
- [28] Pasko, A., Adzhiev, V., Schmitt, B., Schlick, C. Constructive hypervolume modelling, *Graphical Models, a special issue on volume modeling*, 63(6), Academic Press, 2001, 413-442.
- [29] Pasko, A., Pilyugin, V., Pokrovskiy, V. Geometric modeling in the analysis of trivariate functions, *Communications of Joint Institute of Nuclear Research*, P10-86-310, Dubna, USSR, 1986 (in Russian). English translation: *Computers and Graphics*, 12(3/4), 1988, 457-465.
- [30] Rivara, M., Levin, C. A 3D refinement algorithm suitable for adaptive and multi-grid techniques, *J. Comp. and Appl. Math.*, 8, 1992, 281-290.
- [31] Schmidt, M. Cutting cubes – visualizing implicit surfaces by adaptive polygonization, *The Visual Computer*, 10(2), 1993, 101-115.
- [32] Sethian, J. A., *Level Set Methods and Fast Marching Methods*, Cambridge, 1999.
- [33] Sheffer, A., Model simplification for meshing using face clustering, *CAD*, 33(13), 2001, 925-934
- [35] Shin, K., Dutta, D. Constructive representation of heterogeneous objects, *Journal of Computing and Information Science in Engineering*, 1(3), 2001, 205-217.
- [34] V. Shapiro, V., Tsukanov, I. Meshfree simulation of deforming domains, *Computer Aided Design*, 31(7), 1999, 459-471.
- [36] Wyvill, G., McPheeters, C., Wyvill, B. Data structure for soft objects, *The Visual Computer*, 2:4, 1986, 227-23.

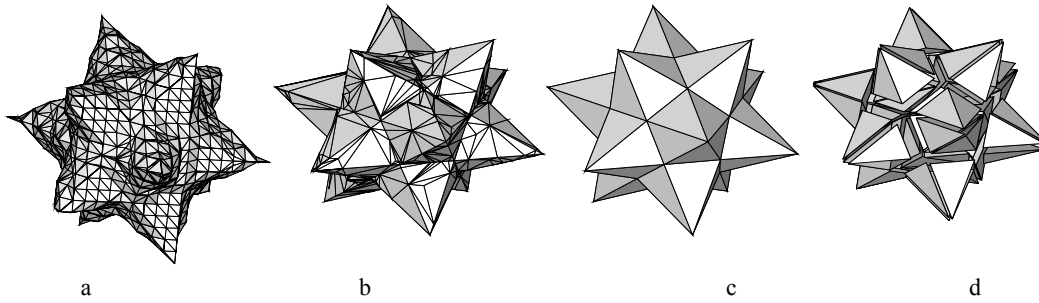


Figure 5: An example of discretization of a functionally based object with sharp features: a) polygonization of the initial object surface; b) a surface mesh after sharp features reconstruction; c) the optimal surface mesh; d) the final tetrahedral tessellation .

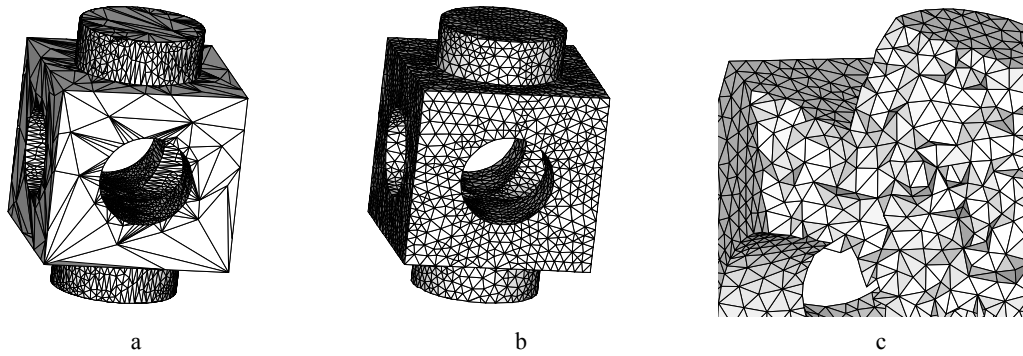


Figure 6: An example of discretization of a complex FRep object: a) surface mesh with reconstructed sharp features (5992 triangles); b) surface mesh after FE adaptation (7794 triangles); c) cut of tetrahedral mesh (enlarged view) (43059 tetrahedra)

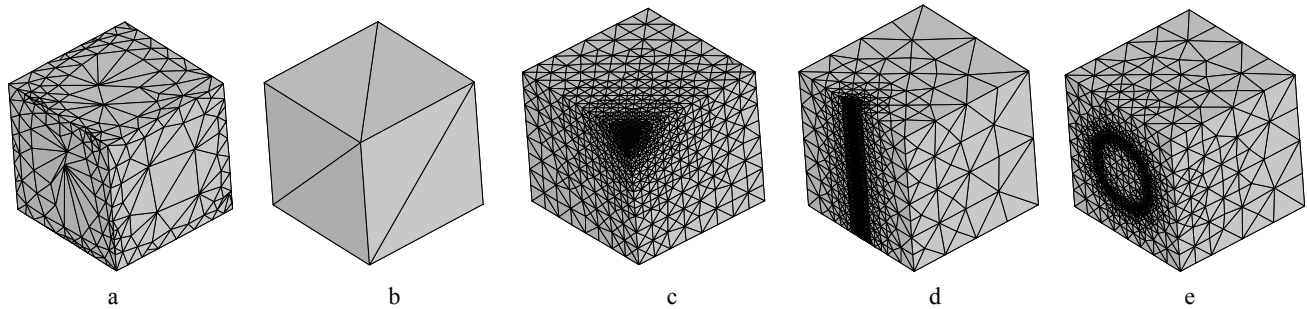


Figure 7. An influence of attributes on the mesh elements' sizes: a) surface mesh with reconstructed sharp features (2776 triangles); b) the minimal surface triangulation (12 triangles); c-e) mesh adaptation to attributes of various types (2052, 7012, 6152 triangles, respectively)

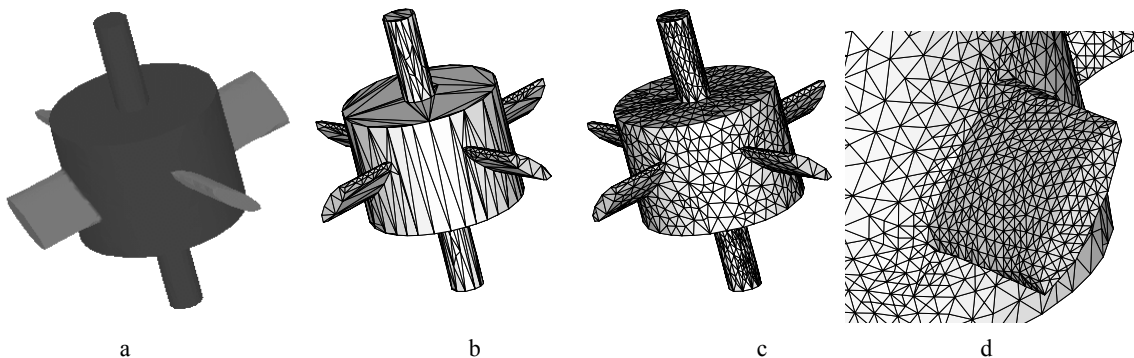


Figure 8: Modeling of an impeller: a) impeller consisting of various materials); b) surface mesh with reconstructed sharp features (846 triangles); c) surface mesh after FE adaptation (2828 triangles) ; d) enlarged fragment of a surface mesh conforming the material attribute