

User Guide

1 Description of the Algorithms

1.1 Doo-Sabin subdivision

1. For each face with vertices v_i , $i = 1, \dots, n$, where $n \in \mathbf{N}$ is the number of all vertices defining the face, the new vertices v_i^1 , $i = 1, \dots, n$ are computed as an affine combination of the original vertices,

$$v_i^1 = \sum_{j=1}^n a_{ij} v_j, \quad \text{where} \quad a_{ij} = \begin{cases} \frac{n+5}{4n} & \text{for } i = j \\ \frac{1}{4n} \left(3 + 2 \cos \frac{2\pi(i-j)}{n} \right) & \text{otherwise.} \end{cases}$$

2. New faces are constructed by the following rules (see Fig. 1a):
 - (a) *F-faces* are generated by connecting consecutively $v_1^1, v_2^1, \dots, v_n^1$ for every face of the control mesh. The new mesh and the control mesh have equal number of the vertices.
 - (b) *E-faces* are generated using two neighbor faces sharing a common edge. The E-faces are always quadrilaterals.
 - (c) *V-faces* are generated using the E-faces of all edges that start from a common vertex. The number of the new faces is equal to the *degree* of the corresponding vertex., i. e. the number of the edges that start from this vertex.

1.2 Catmull-Clark subdivision

1. Compute the *f-point* for each face. This is the centroid of the face, i. e. the average of all vertices defining the face.
2. Compute the *e-point* for each edge. This is the average of the two endpoints of the edge and the two f-points of the faces sharing the edge.
3. Compute the *v-point* v^1 for each vertex v as follows:

$$v^1 = \frac{1}{n}Q + \frac{2}{n}R + \frac{n-3}{n}v,$$

where Q is the average of the f-points of all faces surrounding v , R is the average of the midpoints of all edges with endpoint v , and n is the degree of v .

4. The new faces are generated. The vertices of each new face form a sequence of the following type

$$v^1 \rightarrow e_1^1 \rightarrow f^1 \rightarrow e_2^1 \rightarrow v^1,$$

where the e-points e_1^1 and e_2^1 refer to the two neighboring edges with endpoint v^1 , which belong to the face with f-point f^1 (see Fig. 2a). Note that all faces after the first subdivision level are quadrilaterals.

Figure 2 shows an example of Catmull-Clark algorithm's work on a **cube** as a control mesh.

1.3 Peters-Reif subdivision

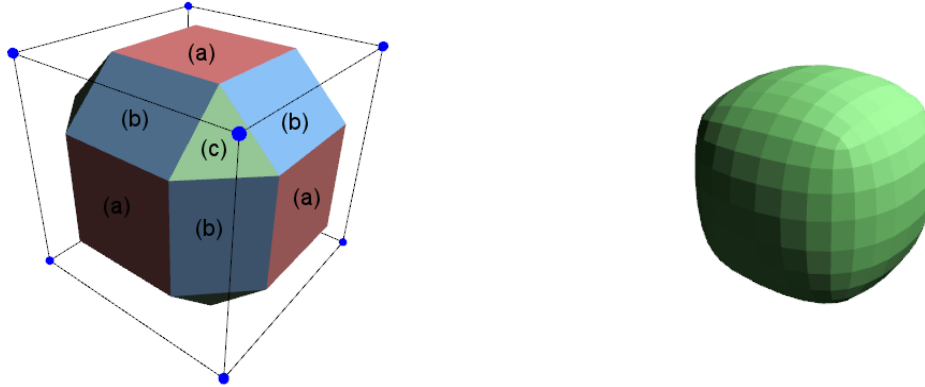
1. Compute the new vertices. These are the midpoints of all edges.
2. Form the new faces. They are of two types (see Fig. 3a):
 - (a) Face inscribed to an old face. Its vertices are the midpoints of the edges of the old face.
 - (b) Face formed by the midpoints of the edges that are incident to a fixed vertex.

2 Description of the package

This is a program package for interactive 3D visualization and comparison the behavior of Doo-Sabin, Catmull-Clark, and Peters-Reif subdivision algorithms for different closed polygonal meshes of arbitrary topology.

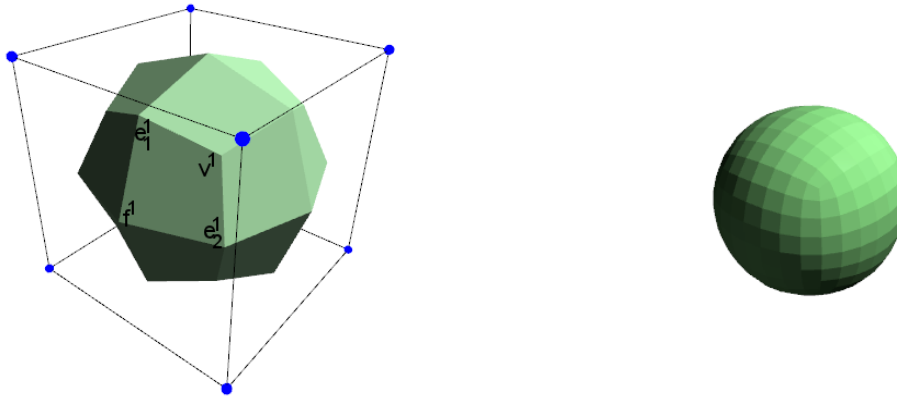
The user interface is intuitive and allows changing of the main visualization elements as lighting, shading, coloring, etc. The main features of the package are:

- The control mesh can be chosen either from a drop-down list, or can be loaded from an external file.
- The level of subdivison can be controlled. Moreover, after the initial mesh is chosen, a dialog window showing the required memory allocations appears so that the user can estimate apriori the maximum subdivision level.
- The program analyzes the loaded mesh for correctness and proceeds with a valid mesh only. Otherwise the incorrect vertices are marked in red. Note that a mesh that is not closed is recognized as invalid. An example of invalid mesh is shown in Fig. 4.
- The program visualizes the meshes in three modes: *Solid*, *Wireframe*, and *Solid with wireframe*, with options for visual adjustments of the color, points size and mesh thickness. *Wireframe* mode is more convenient to demonstrate and analyze



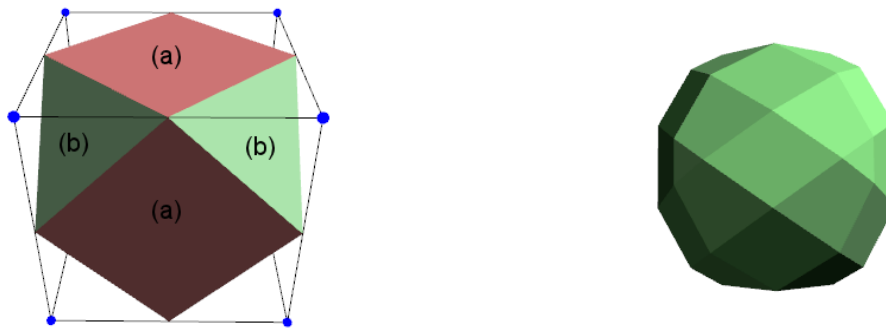
(a) The subdivision rules: (a) F-faces; (b) E-faces; (c) V-faces (b) The mesh after 3 subdivision steps

Figure 1: Doo-Sabin subdivision for a **cube** as a control mesh



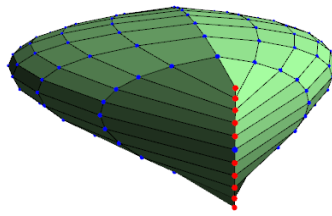
(a) The subdivision rules for generating two types of new faces (b) The mesh after 3 subdivision steps

Figure 2: Catmull-Clark subdivision for a **cube** as a control mesh

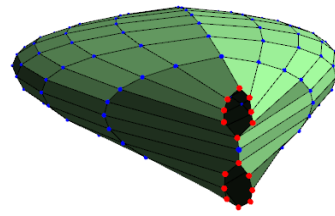


(a) The subdivision rules for generating new faces (b) The mesh after 3 subdivision steps

Figure 3: Peters-Reif subdivision for a **cube** as a control mesh

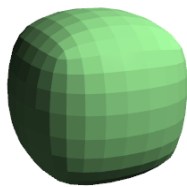


A mesh that looks closed...

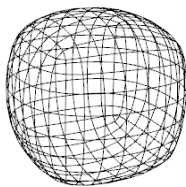


...but in fact is not: it has coincident vertices.

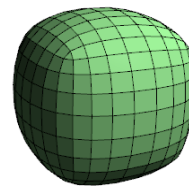
Figure 4: An invalid mesh that looks closed



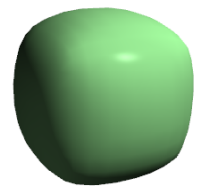
Solid and Flat



Wireframe



Solid with wireframe



Solid and Smooth

Figure 5: The different visualization modes

the subdivision process while *Solid* and *Solid with wireframe* modes are appropriate to study and compare the shape and the smoothness of the obtained surfaces. *Solid* and *Solid with wireframe* have also an additional option for a choice between two light types: *Flat* and *Smooth*, see Fig. 5.

- The surfaces can be edited interactively by rotation, translation and resizing using the mouse with option for synchronizing the transformations between different views.
- The mesh can be visualized using one, two or all three algorithms simultaneously on the screen with an option for showing the next step vertices, edges, and faces which allow better visual comparison of the results.
- The current mesh can be exported as a file. This option, together with the interactivity of the vertices, can be used to create new meshes. For example, we can apply first Doo-Sabin algorithm, then move some of the vertices, and then apply Catmull-Clark algorithm.