HOM File Format

Hervé Delingette
INRIA Méditerranée, Sophia Antipolis, France

June 2017

1 File Format Description

1.1 Constraints on High Order Meshes

There exists no standard file format to describe high order triangular and tetrahedral meshes of any degree. For instance common file formats such as VTK or GMSH only support triangular and tetrahedral elements of degree 2 (and 3 for GMSH). The HOM file format aims to fill this need and the SOFA plugin *SOFA-HighOrder* provides importer and exporter classes to visualize and use such meshes for scientific computing.

Suitable Element Representations. In version 1, the HOM file format is only valid for high order simplicial meshes such as high order triangulations and tetrahedralization. Furthermore, it is only suited for elements where degrees of freedom are represented as control points such as in Bezier or Lagrange elements. Elements like Hermite elements where degrees of freedom are expressed as derivatives are not represented in this format.

Suitable Mesh Representations. In the HOM format, all elements are of the same type (triangles, tetrahedra) and have the same degree.

Shape functions . A finite element is the union of its geometric representation (as a collection of control points) and shape functions. In this format, shape functions are not described explicitly but only referenced within a family of generic shape functions such as Bezier or Lagrange.

1.2 Generic representation of High Order Simplificial Meshes

In the remainder, we consider high order simplicial meshes of degree d consisting of N_v vertices, N_e edges, N_{tr} triangles, N_{tet} tetrahedra. The total number of control points is N_p which is defined as :

$$N_p = N_v + (d-1)N_e + \frac{(d-1)(d-2)N_{tr}}{2} + \frac{(d-1)(d-2)(d-3)N_{tet}}{6} \tag{1}$$

A control point lying on edge can be unambiguously indexed by :

- the index i_e in the list of edges of the edge (v_0, v_1) where it is located.

- its *Edge Index Vector* (EIV) consisting of a pair of integers $\mathbf{p} = (i, j)$ such that $|\mathbf{p}| = i + j = d$.

Similarly, a control point lying on triangle can be unambiguously indexed by :

- the index i_{tr} in the list of triangles of the triangle (v_0, v_1, v_2) where it is located.

- its *Triangle Index Vector* (TRIV) consisting of a triplet of integers $\mathbf{p} = (i, j, k)$ such that $|\mathbf{p}| = i + j + k = d$.

Finally, a control point lying on tetrahedron can be unambiguously indexed by :

- the index i_{tet} in the list of tetrahedra of the tetrahedron $(v_0, v_1, v_2, v3)$ where it is located.

- its *Tetrahedron Index Vector* (TETIV) consisting of a quadruplet of integers $\mathbf{p} = (i, j, k, l)$ such that $|\mathbf{p}| = i + j + k + l = d$.

Finally note that if $d = 1$ then $N_p = N_v$ and the control points are only the vertices of the simplicial mesh.

1.3 Format Description

The HOM file format is a text file that consists of 4 parts :

1. **the Header** describing the key parameters of the mesh such the degree of the spline, the dimension of the elements and the nature of the shape functions.

2. **the control point position** including the number of control points and their coordinates.

3. **the topology description** including the set of edges, triangles and tetrahedra.

4. **the high order control points description** describing the indexing of each control points lying on edges, triangles and tetrahedra.

The **header section** consists of 4 lines :

Line 1 includes the string "HOMF Version 1" and describes the file type and its version.

Line 2 includes 2 integers "dimEmbedding" and "dimSimplex" where *dimEmbedding* is the dimension of the embedding space and *dimSimplex* is the dimension of the simplex element. More precisely, $dimSimplex = 2$ for high order triangulations and $dimSimplex = 3$ for high order tetrahedral meshes.

Line 3 includes 1 integer "degree", the degree of the high order element. It has to be greater than 0 $degree \geq 1$, and when $degree = 1$ then the mesh is a regular (linear) simplicial mesh.

Line 4 includes 1 integer "shapeFunctionType" which characterizes the shape function with the following convention:

$shapeFunctionType$	Name of shape function
0	Bezier shape function
1	Lagrange shape function

The **control point position part** consists of $N_p + 1$ lines :

- **Line 1** includes the number of control points N_p.

- **From Line 1 to Line** $N_p + 1$ includes $dimEmbedding + 1$ float coordinates corresponding to the $dimEmbedding$ coordinates of the control point followed by its weight w. For regular (i.e. integral) splines, the weight should be set to 1.

The **topology description part** consists of the description of the edges, triangles and tetrahedra (if $dimSimplex = 3$) of the underlying simplicial mesh. This part consists of :

- **Edge section.** $N_e + 1$ lines where the first line is the number of edges N_e and the remaining lines describes the two vertices making each edge.

- **Triangle section.** $N_{tr} + 1$ lines where the first line is the number of triangles N_{tr} and the remaining lines describes the three vertices making each triangle.

- **Tetrahedron section.** $N_{tet} + 1$ lines where the first line is the number of tetrahedra N_{tet} and the remaining lines describes the four vertices making each tetrahedron. This section is only present for high order tetrahedral meshes.

The number of lines is therefore $N_e + N_{tr} + 2$ for high order triangulations and $N_e + N_{tr} + N_{tet} + 3$ for high order tetrahedrizations.
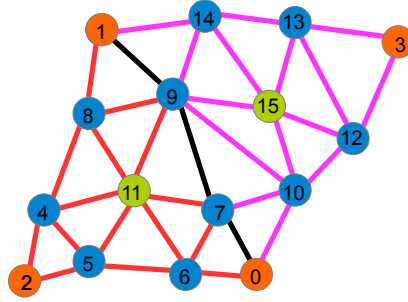
The **high order control points description part** aims at describing the indexing of all control points in the high order mesh. Note that N_v is not explicitly provided in this format but can be recovered by the knowledge of N_p, N_e, N_{tr}, N_{tet} and d from Eq. 1.

This **high order control points description part** consists of :

- **Edge Control Point section.** It includes $N_e * (d - 1)$ lines where each line consists of 4 integers : i_p, i_e, p_0, p_1. Index i_p is the index of the control point in the list of control points as provided by the control point position part. Index i_e is the index of the edge in the list of edges as provided in the topology description part. The two indices (p_0, p_1) make the edge index vector (such that $p_0 + p_1 = d$) which positions the control point on the edge.

- **Triangle Control Point section.** It includes $N_{tr} * (d - 1) * (d - 2)/2$ lines where each line consists of 5 integers : $i_p, i_{tr}, p_0, p_1, p_2$. Index i_p is the index of the control point in the list of control points as provided by the control point position part. Index i_{tr} is the index of the triangle in the list of triangles as provided in the topology description part. The three indices (p_0, p_1, p_2) make the triangle index vector (such that $p_0 + p_1 + p_2 = d$) which positions the control point on the triangle.

- **Tetrahedron Control Point section.** It includes $N_{tet} * (d - 1) * (d - 2) * (d - 3)/6$ lines where each line consists of 6 integers : $i_p, i_{tet}, p_0, p_1, p_2, p_4$. Index i_p is the index of the control point in the list of control points as provided by the control point position part. Index i_{tet} is the index of the tetrahedron in the list of tetrahedra as provided in the topology description part. The four indices (p_0, p_1, p_2, p_3) make the tetrahedron index vector (such that $p_0 + p_1 + p_2 + p_3 = d$) which positions the control point on the tetrahedron.

1.4 Example

In Fif.1 an example of a high order mesh consisting of 2 cubic Bezier triangles is exported in a HOM file format. There are $N_p = 16$ control points that are split into $N_v = 4$ vertices, $N_e * 2 = 10$ edge control points and $N_{tr} * 1 = 2$ triangle control points. The edge index vectors are either $(1, 2)$ or $(2, 1)$ and triangle index vectors are $(1, 1, 1)$.

Figure 1: (Top) Example of a high order mesh consisting in two cubic triangles sharing an edge. Orange vertices are vertices of the 2 triangles, blue vertices are edge control points and green vertices are triangle control points. (Bottom) Left panel corresponds to the HOM file while the right panel is the explanation of the corresponding line.

2 SOFA Importer and Exporter Classes

2.1 Importer Class : HOMFFileExporter

To export a high order mesh in HOM format from a SOFA scene, one should use the component HOMFFileExporter. The component can export a mesh at the beginning of the simulation, or at the end, or every n steps. The object HOMFFileExporter should be placed in the same node or below the object HighOrderTriangleSetTopologyContainer (or HighOrderTetrahedron-SetTopologyContainer) associated with the mesh. Furthermore, it has also to be placed in the same node or below the object BezierTriangleSetGeometryAlgorithms (or BezierTetrahedron-SetGeometryAlgorithms) or similar classes for different shape functions.

Important fields of this object are :

Field	Description
filename	base file name of exported file
overwrite	When false as default create a new file at each export, with suffix added to the filename
exportEveryNumberOfSteps	the number of steps between 2 exports, if 0 as default then no export is performed
formatVersion	the format version of the HOM file
exportAtBegin	if the file should be exported before any iteration
exportAtBegin	if the file should be exported when the simulation is finished

See examples in GenerateSphere.xml

2.2 Exporter Class : MeshHOMFLoader

To import a high order mesh in HOM format, use component MeshHOMFLoader. As input it only requires the field "filename" which is self explanatory. As output, it fills the fields "position", "position2D", "edges", "triangles", "tetrahedra", "weights", "inputHighOrderEdges", "inputHighOrderTriangles", "inputHighOrderEdgePositions", "inputHighOrderTrianglePositions", "inputHighOrderTetrahedronPositions". Those fields can then be connected to the classes HighOrderTetrahedronSetTopologyContainer and HighOrderTriangleSetTopologyContainer to initialize the mesh.

See TestLoaderHOMFFile2D.xml and TestLoaderHOMFFile3D.xml as examples.