Mesh primitive counting formulas for subdivision surfaces

Martin Čavarga

Department of Algebra and Geometry, Comenius University Bratislava
Mlynská dolina 5692, 841 04, Bratislava, Slovak Republic
email: martin.cavarga@fmph.uniba.sk

Abstract. Subdivision surfaces are commonly used to simulate additional detail for mesh objects on the scene. For certain problems, such as Lagrangian shrink-wrapping, determining the number of vertices of the subdivision surface is crucial for estimating vertex density. Furthermore, gauging the number of mesh primitives relative to the subdivision level becomes useful for memory preallocation during the surface creation process. We propose a general method for estimating these values using solutions to simple systems recurrence equations.

Keywords: Subdivision, mesh, level-of-detail, recurrence.

1 Introduction

Subdivision surfaces, created by refining a base cage surface, are used in geometric design [7] and real-time rendering [8]. This paper presents counting formulas for estimating mesh primitive (vertices, edges, faces) counts based on the connectivity of the base mesh and the recursive properties of subdivision (see Fig. 1 (a)). These formulas help in computing vertex density [6] and in efficient memory allocation for subdivision surface construction.

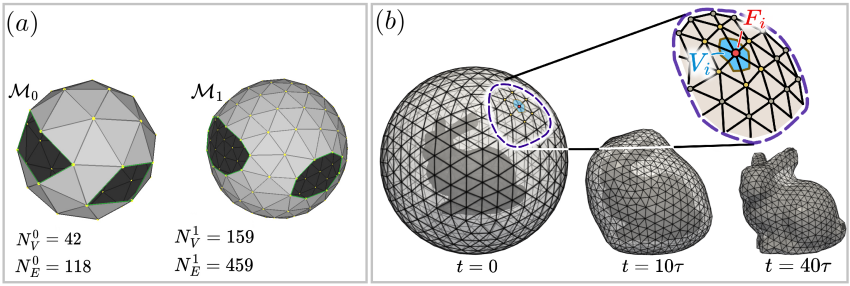

Fig. 1: (a) The amount of mesh vertices N_V^s and edges N_E^s with respect to subdivision level $s = 0, 1...$ depends on the initial connectivity $\mathcal{M}_0$. (b) Three snapshots of the evolution of an icosphere with subdivision level $s = 3$ under an advection-diffusion shrink-wrapping model introduced in [6]. The detail shows a chosen mesh vertex F_i (red) with its corresponding barycentric Laplacian co-volume V_i (blue).

2 Motivation

According to Section 2.4 of [6] a semi-implicit formulation of a parabolic advection-diffusion evolution applied to manifold mesh surfaces, requires us to ensure that the time step size $\tau > 0$ is close to the value of mean area $\mu(V)$ of a barycentric Laplacian co-volume V surrounding each mesh vertex (see Fig. 1). The most straightforward approach is to compute a 2-dimensional scaling factor

$$\phi = \sqrt{\frac{\tau}{\mu_r(V)}}\ , \quad \text{where} \quad \mu_r(V) = \frac{4\pi r^2}{N_V^s}. \tag{1}$$

This assumes a spherical evolving surface with uniform vertex distribution. To achieve stability and return to the original scale, the mesh is scaled using ϕ and then reverted with $1/\phi$.

For a surface formed by $s > 0$ subdivision steps, a counting formula evaluates N_V^s from the recursive nature of the subdivision operation. A typical example of such surface is an *icosphere* which is a form of *spherical geodesic grid* used as a discrete computational domain for applications such as climate modeling [13] and global data visualization [16]. Starting from initial vertex count of an icosahedron $N_V^0 = 12$ we have

$$N_V^1 = 42,\ N_V^2 = 162,\ N_V^3 = 642,\ N_V^4 = 2562,\ N_V^5 = 10242, \ldots\ .$$

Other, more general surfaces with different initial valences for each vertex, would need individual evaluation, which is clearly impractical.

3 Related work

In acoustic simulations Alarcão et al. [2] subdivided an icosahedron's radiation pattern for ray direction determination, with formulas for counting vertices and faces:

$$N_V^s = 5\left(2^{2s} - 2^s + 2\sum_{m=1}^{2^s} m\right) + 2\ , \quad N_F^s = 20 \cdot 4^s. \tag{2}$$

[12] discusses the OLAM geodesic grid construction, beginning with an icosahedron inscribed in the earth. Each triangle subdivides into N^2 smaller triangles, introducing $30(N^2 - 1)$ new edges and $10(N^2 - 1)$ vertices. This approach differs from the icosahedron counting formula (2) as it does not consider recursion and solely focuses on 4:1 triangle subdivisions.

Both techniques, however, only handle a single type of triangular base surface under a 4:1 triangle subdivision.

4 Manifold mesh subdivision theory

Polygonal meshes are perhaps the most widely adopted representation in the realm of 3D data storage and display. We evaluated the formal definitions of meshes in the prominent literature from the field, such as Botsch et al. [3], and Hoppe et al. [10], and formulated the following definition:

Definition 4.1. Let $\mathcal{K}$ be an abstract simplicial complex containing at most 2-simplices (triangles). Let $V = \{\mathbf{v}_1, ..., \mathbf{v}_{N_V}\} \subset \mathbb{E}^3$ a finite set of points referred to as the *vertex set*. Then $(\mathcal{K}, V)$ is called a *triangle mesh*. A *polygon* is a union of 2-simplices (triangles) each of which is edge-adjacent to another triangle[1]. Let $\mathcal{M} \supseteq \mathcal{K}$ possibly contain polygons in addition to triangles in $\mathcal{K}$. Pair $(\mathcal{M}, V)$ is then called a *polygonal mesh*.

A pure point-set surface image $\overline{\mathcal{M}} \subset \mathbb{E}^3$ of the mesh is known as the *geometric realization* of $(\mathcal{M}, V)$. This notion is explained in more detail in Section 2 of [10].

Meshes approximating smooth surfaces require distinguishing between general simplicial realizations and those approximating smooth surfaces. Hence, we distinguish between manifold and non-manifold meshes.

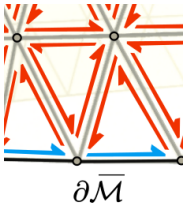

Fig. 2: Each triangle has 3 half-edges, and each interior edge has two opposing half-edges, and there is a single half-edge for each boundary edge.

Definition 4.2. Let X be a topological 2-manifold, and $F : X \to \mathbb{E}^3$ its immersion. A polygonal mesh $(\mathcal{M}, V)$ is then said to be a *manifold mesh* if $\overline{\mathcal{M}} = F[X]$. If the geometric realization $\overline{\mathcal{M}}$ does not have a boundary $\partial\overline{\mathcal{M}}$, we say that $(\mathcal{M}, V)$ is *watertight*.

A 2-manifold mesh is sometimes referred to as a *surface mesh* for which there exists an efficient data structure [15] which uses ordered 1-simplices - *half-edges* (see Fig.2).

Now define a map $\mathcal{M} \to \mathcal{M}^*$ referred to as a *tessellation-changing operation* on $\mathcal{M}$, such that $\mathcal{M}^* \cap \mathcal{M} \neq \varnothing$ where $\mathcal{M}^*$ is also an extension of the resulting simplicial complex $\mathcal{K}^*$ containing modified polygons forming a polygonal mesh $(\mathcal{M}^*, V^*)$. An example of such operation is evidently subdivision:

Definition 4.3. Let $T = \{i_0, i_1, i_2\} \in \mathcal{M}$ be a triangle in a surface mesh $\mathcal{M}$. A tessellation-changing operation $\Sigma : \mathcal{M} \to \mathcal{M}^*$ which introduces new 0-simplices (vertices) $\{i^*_{01}\}$, $\{i^*_{12}\}$, and $\{i^*_{20}\}$ per each edge

[1] For example, triangles $T_0 = \{i, j, k\}$ and $T_1 = \{j, i, l\}$ are *edge-adjacent* sharing edge $\{i, j\} \in \mathcal{K}$.

$\{i_0, i_1\}, \{i_1, i_2\}, \{i_2, i_0\} \in \mathcal{M}$, and replaces T with four triangles:

$$\begin{aligned} T_0 &= \{i_0, i_{01}^*, i_{20}^*\}, \quad T_1 &&= \{i_{20}^*, i_{01}^*, i_{12}^*\}, \\ T_2 &= \{i_{01}^*, i_1, i_{12}^*\}, \quad T_3 &&= \{i_{20}^*, i_{12}^*, i_2\}, \end{aligned} \tag{3}$$

is called a 4:1 *triangle subdivision* on $\mathcal{M}$. If the change in connectivity information (for triangles and edges) also propagates to all three possible edge-neighboring triangles $T^{(01)}$, $T^{(12)}$, and $T^{(20)}$ of T, we say that Σ is *compatible with* $T^{(01)}$, $T^{(12)}$, or $T^{(20)}$. If Σ is targeting all triangles of $\mathcal{M}$, and compatible with respect to all neighbors of all triangles, we say that Σ is *globally-compatible*.

A subdivision of T compatible with respect to $T^{(e)},\ e \in \{01, 12, 20\}$ provided that no neighbors of $T^{(e)}$ are also subdivided yields two edge-adjacent triangles $T^{(e)} \mapsto T_0^{(e)}, T_1^{(e)}$ sharing vertex $\{i_e^*\} \in \mathcal{M}^*$. For the purposes of this paper, however, we only consider globally-compatible subdivisions, that is: if T subdivides into T_0, T_1, T_2, and T_3, so do its edge neighbors $T^{(e)},\ e \in \{01, 12, 20\}$ if they exist.

An *approximating*[2] variant of such subdivision is a scheme proposed by Loop [11]. An interpolating variant would be, for example, a simple spherical projection scheme

$$\mathbf{v}_e^* \leftarrow \mathrm{proj}_{\mathbb{S}^2}(\mathbf{v}_e^*) = \mathbf{v}_e^* / \|\mathbf{v}_e^*\|. \tag{4}$$

for constructing an icosphere.

This is, of course, not the only way to subdivide triangle faces in $\mathcal{M}$. If we also add an interior vertex $\{i_{012}^*\} = \{i^*\} \in \mathcal{M}^*$ subdividing T into three quadrilaterals

$$Q_0 = \{i_0, i_{01}^*, i^*, i_{20}^*\},\ Q_1 = \{i_{01}^*, i_1, i_{12}^*, i^*\},\ Q_2 = \{i^*, i_{12}^*, i_2, i_{20}^*\}, \tag{5}$$

we formulate the *combinatorial Catmull-Clark subdivision* variant for triangles with its approximating scheme described in [4] and [5]. This scheme can be extended to subdivide an arbitrary mesh ($m \geq 3$)-gon with the resulting quads $Q_0, ..., Q_{m-1}$ sharing the inserted interior vertex $\{i^*\}$. For example, if $m = 4$, we get

$$\{i_0, i_{01}^*, i^*, i_{30}^*\},\ \{i_{01}^*, i_1, i_{12}^*, i^*\},\ \{i^*, i_{12}^*, i_2, i_{13}^*\},\ \{i_{30}^*, i^*, i_{13}^*, i_3\}. \tag{6}$$

The subdivision operation can, of course, be repeated $s > 0$ times where we write $\Sigma^s = \overbrace{\Sigma \circ ... \circ \Sigma}^{s\text{-times}}$. Infinite application of subdivision then leads to a *limit surface*.

[2]Such that $(\{i\}, \mathbf{v}_i) \neq \Sigma(\{i\}, \mathbf{v}_i)$ because of the movement of positions $\mathbf{v}_i$ of the original vertices $\{i\} \in \mathcal{M}$ under Σ.

5 Counting formulas

Recall that in Section 2, we wanted to evaluate the number of mesh vertices N_V as a function of the subdivision level. This number is closely related to both the count of edges N_E and faces N_F through the *Euler polyhedron formula* [9]. In this section, we utilize all the tools at our disposal to prove such "counting formulas" for different types of subdivision.

Theorem 5.1. *Let $\mathcal{M}_s = \Sigma^s(\mathcal{M})$, $s \in \mathbb{N}_0^+$ be a watertight triangle surface mesh, and let $\Sigma : \mathcal{M}_{s-1} \mapsto \mathcal{M}_s$, $s > 0$ be a globally-compatible $4:1$ subdivision inserting a single vertex for each edge $e \in \mathcal{M}_s$. Let N_V^s, N_E^s, and N_F^s denote the number of vertices, edges, and faces of $\mathcal{M}_s$ respectively. Then given starting counts N_V^0, N_E^0, and N_F^0 we have:*

$$N_E^s = 4^s N_E^0, \;\; N_F^s = 4^s N_F^0, \tag{7}$$

$$N_V^s = \frac{1}{3}\big(N_E^0(4^s - 1) + 3N_V^0\big). \tag{8}$$

Proof. First, we consider that Σ subdivides each face into 4 faces, that is $N_F^s = 4N_F^{s-1}$ which yields $N_F^s = 4^s N_F^0$ for any $s \in \mathbb{N}$. However, since we insert a new vertex for each existing edge, the number of added vertices in step s will be equal to edge count N_E^{s-1}. This gives rise to a system of recurrence equations:

$$\begin{aligned} N_V^s &= N_V^{s-1} + N_E^{s-1}, \\ N_E^s &= 4N_E^{s-1}. \end{aligned} \tag{9}$$

Before solving this system, we need to verify that under Σ the number of edges in $\mathcal{M}_{s-1}$ increases to 4 times the count in previous step (the second equation for N_E).

Since for a triangle mesh without boundary, the total number of half-edges is:

$$N_H = 2N_E = 3N_F, \tag{10}$$

and subdivision from Definition 4.3 updates the number of edge by doubling the amount of existing edges, and adding 3 new interior edges per each triangle, we have

$$N_E^s = 2N_E^{s-1} + 3N_F^{s-1} = 4N_E^{s-1},$$

using (10).

After solving (9) using the s-th power of the matrix of the system, we get $N_E^s = 4^s N_E^0$ and (8). □

The fact that the proof of the above theorem depends only on identity (10) yields:

Corollary 5.1. *The statement of Theorem 5.1 is independent from the genus of the surface mesh.*

Introducing boundary violates idenity (10) which must be replaced by

$$N_H = 2N_{IE} + N_{BE} = 3N_F, \tag{11}$$

where $N_E = N_{IE} + N_{BE}$ with interior and boundary edge counts N_{IE}, N_{BE} respectively.

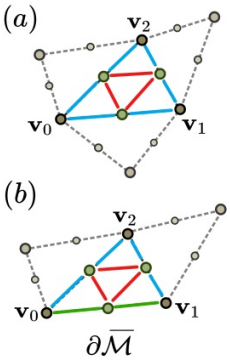

Fig. 3: (a) subdividing interior (Theorem 5.1) (b) and boundary triangles (Theorem 5.2).

Theorem 5.2. *Let $\mathcal{M}_s = \Sigma^s(\mathcal{M})$, $s \in \mathbb{N}_0^+$ be a possibly non-watertight triangle surface mesh, and let $\Sigma : \mathcal{M}_{s-1} \mapsto \mathcal{M}_s$, $s > 0$ be a globally-compatible $4:1$ subdivision inserting a single vertex for each edge $e \in \mathcal{M}_s$. Let N_V^s denote the number of vertices, N_{IE}^s the number of interior edges, and N_{BE}^s the number of boundary edges of $\mathcal{M}_s$. Then:*

$$\begin{aligned} N_V^s &= \frac{1}{6}(4^s - 4 + 3 \times 2^s)N_{BE}^0 + \frac{1}{3}(4^s - 1)N_{IE}^0 + N_V^0, \\ N_{IE}^s &= 2^{s-1}\big((2^s - 1)N_{BE}^0 + 2^{s+1}N_{IE}^0\big), \\ N_{BE}^s &= 2^s N_{BE}^0. \end{aligned} \tag{12}$$

Proof. Recurrence relation (9) needs to be adjusted, so that it handles edge vertex insertion differently for interior, and for boundary edges. Σ applied to boundary edges simply doubles their amount N_{BE}^s. For counting interior edges N_{IE}^s requires us to use the generic identity $N_E^s = 2N_E^{s-1} + 3N_F^{s-1}$ combined with (11) which yields the second equation in:

$$\begin{aligned} N_V^s &= N_V^{s-1} + N_{IE}^{s-1} + N_{BE}^{s-1}, \\ N_{IE}^s &= 4N_{IE}^{s-1} + N_{BE}^{s-1}, \\ N_{BE}^s &= 2N_{BE}^{s-1}. \end{aligned} \tag{13}$$

(12) is then the solution of system (13). □

Theorem 5.3. *Let $\mathcal{M}_s = \Sigma^s(\mathcal{M})$, $s \in \mathbb{N}_0^+$ be a watertight quad surface mesh, and let $\Sigma : \mathcal{M}_{s-1} \mapsto \mathcal{M}_s$, $s > 0$ be a globally-compatible $4:1$ subdivision inserting a single vertex for each edge $e \in \mathcal{M}_s$, and a vertex*

for each face $Q \in \mathcal{M}_s$, according to connectivity scheme (6). Then

$$\begin{aligned} N_E^s &= 2^s(N_E^0 + 2(2^s - 1)), \\ N_F^s &= 4^s N_F^0, \\ N_V^s &= (2^s - 1)^2 N_F^0 + (2^s - 1)N_E^0 + N_V^0. \end{aligned} \tag{14}$$

Proof. Similarly to the derivation of previous recurrence formulas, we conclude that the amount of pre-existing edges doubles during subdivision, and we add four additional interior edges connecting from the newly inserted edge vertices $\{i_e^*\}, e \in \{01, 12, 23, 30\}$ to the new interior vertex $\{i^*\}$. Analogously, the newly inserted vertices $\{i_{01}^*\}, \{i_{12}^*\}, \{i_{23}^*\}, \{i_{30}^*\}$, and $\{i^*\}$ contribute to the updated vertex count:

$$\begin{aligned} N_V^s &= N_V^{s-1} + N_E^{s-1} + N_F^{s-1}, \\ N_E^s &= 2N_E^{s-1} + 4N_F^{s-1}, \\ N_F^s &= 4N_F^{s-1}. \end{aligned} \tag{15}$$

Solving (15) then yields (14). □

6 Tests and performance improvement

We tested the simplest triangle case in Theorem 5.1 for an icosphere, and additional watertight input meshes. We also verified the validity of counting formulas (8) for tori with higher genus (see Fig. 4 (b)). An icosphere with two holes (see Fig. 1 (a) and Fig. 4 (a)) was used to verify Theorem 5.2. Moreover, the utility of theorems in Section 5 was tested via time measurement speedup for Loop subdivision on dataset in 4 (c) while using preallocated memory with the a priori known mesh vertex, edge, and face counts (see Table 1).

Armadillo	Blub	Bunny	Max Planck	3Holes	Rocker Arm
0.85%	3.61%	2.08%	2.08%	2.06%	2.08%

Table 1: Speedup percentages for various test meshes.

s	1	2	3	4	5	6
Speedup [%]	-94.43	-31.23	-7.23	9.13	15.15	12.83

Table 2: Recursive vs preallocated icosphere construction speedup with respect to subdivision level s.

Evidently, the computation of new vertex positions in Loop subdivision [11] limits the potential gain from preallocation. For this reason we

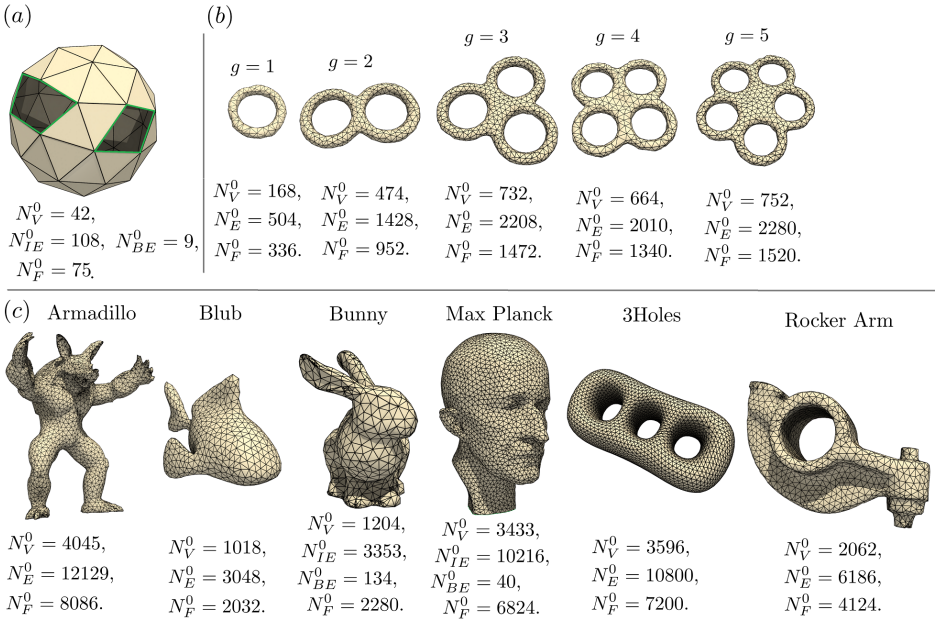

Fig. 4: Three different tests carried out on mesh datasets: (a) icosphere with boundary, (b) testing counting formulas for arbitrary genus, and measuring preallocation performance on standard datasets (c).

performed another set of tests for an icosphere subdivision scheme (4). We tried to mitigate the expensive vertex position computation by performing simple barycentric interpolation within base triangles of an icosahedron followed by more complex connectivity construction. As can be seen in Table 2, we sacrifice a lot of computation time to the construction of connectivity up to subdivision level $s = 3$ after which we start to save as much as 15 % of the time.

Note that for the test mesh collection (c), we perfomed *isotropic remeshing* [1] to obtain better vertex distribution. This step was done in MeshLab™by the Visual Computing Lab team from ISTI, Pisa, and the final 3D visualizations were rendered in ParaView™by the Kitware team.

7 Conclusion

Originating from the motivation for vertex counting formula, specifically for 4:1 subdivision aimed at stabilizing the semi-implicit formulation of Lagrangian evolution (refer to Section 2), this paper derives and justifies the utility of counting formulas for mesh vertices, edges, and faces under

recursive face subdivision operations (see Section 4). Theorems in Section 5 formulate the "counting formulas" for triangle 4:1 subdivision for meshes with arbitrary genus, and extend the statement even for meshes with boundary loops (Theorem 5.2), and for Catmull-Clark scheme on quad meshes (Theorem 5.3).

We validate the theoretical results with tests in Section 6 including the measurement for performance improvement under memory preallocation which yields up to 3.6% speedup for Loop subdivision and 15% acceleration in parametric icosphere construction.

8 Acknowledgments

The credit for test meshes goes to the Stanford 3D Model Repository (Armadillo, Bunny), Keenan Crane (Blub), Leif Kobbelt and the team at RWTH Aachen (Max Planck), to INRIA (Rocker Arm), and to the Libgl team (3Holes). Additionally, considerable gratitude goes to the development team of the PMP library led by Daniel Seiger and Mario Botsch.

Special thanks goes to doc. RNDr. Zbyněk Šír, PhD. for providing his installation of Wolfram Mathematica™ for obtaining solution (14) of system (15) moments before the presentation of this paper.

References

[1] Alliez, Pierre, Eric Colin De Verdire, Olivier Devillers, and Martin Isenburg. Isotropic surface remeshing. *Shape Modeling International.*, pp. 49-58. IEEE, 2003.

[2] Alarcão, Diogo, J. Luis Bento Coelho, Roberto Tenenbaum: On the Use of Hybrid Methods for Fast Acoustical Simulations in Enclosures, *Proceedings of Forum Acusticum '02*, 2002.

[3] Botsch, Mario, Leif Kobbelt, Mark Pauly, Pierre Alliez, and Bruno Lévy. *Polygon Mesh Processing*, CRC press, 2010.

[4] Doo, D. and Sabin, M. Behaviour of recursive division surfaces near extraordinary points, *Computer-Aided Design*, vol 10 (6), p. 356-360, 1978.

[5] Catmull, Edwin, and Clark, James. Recursively generated B-spline surfaces on arbitrary topological meshes, *Seminal graphics: pioneering efforts that shaped the field*, p. 183-188, 1998.

[6] Čavarga, Martin: Advection-Driven Shrink-Wrapping of Triangulated Surfaces, *Proceedings of the 26th Central European Seminar on Computer Graphics : CESCG 2022*, pp. 95-104, 2022.

[7] DeRose, Tony and Kass, Michael and Truong, Tien: Subdivision Surfaces in Character Animation, *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pp 85-94, 1988.

[8] Dupuy, Jonathan and Deliot, Thomas: Bisection Based Triangulation of Catmull Clark Subdivision, *SIGGRAPH '22 Advances in Real-Time Rendering*, 2022.
[9] Henle, Michael. *A combinatorial introduction to topology.* Dover Publications. ch. 5, s. 26, 1994.
[10] Hoppe, H., DeRose, T., Duchamp, T., McDonald, J., & Stuetzle, W.: Mesh optimization, *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pp 19-26, 1993.
[11] Loop, Charles. *Smooth subdivision surfaces based on triangles*, 1989.
[12] Osthoff, Carla, Roberto P. Souto, Fabrício Vilasbôas, Pablo Grunmann, Pedro L. Silva Dias: Improving Atmospheric Model Performance on a Multi-Core Cluster System, *Atmospheric Model Applications*, ch 1, pp 1-26, 2012.
[13] Randall, David A., Todd D. Ringler, Ross P. Heikes, Phil Jones, and John Baumgardner. Climate modeling with spherical geodesic grids. *Computing in Science & Engineering 4*, no. 5: pp. 32-41, 2002.
[14] Sieger, Daniel and Botsch, Mario. *The Polygon Mesh Processing Library* `http://www.pmp-library.org`, 2019.
[15] Sieger, D., and Botsch, M. Design, implementation, and evaluation of the surface mesh data structure, *Proceedings of the 20th International Meshing Roundtable*, pp. 533–550, 2012.
[16] Xie, Jinrong, Hongfeng Yu, and Kwan-Liu Ma. Visualizing large 3D geodesic grid data with massively distributed GPUs. *2014 IEEE 4th Symposium on Large Data Analysis and Visualization* (LDAV), pp. 3-10. IEEE, 2014.