Pacific Graphics (2018)
H. Fu, A. Ghosh, and J. Kopf (Editors)

Short Paper

Gauss-Seidel progressive iterative approximation (GS-PIA) for Loop surface interpolation

Zhihao Wang [1], Yajuan Li [1], Weiyin Ma[2] and Chongyang Deng[1] †

[1] School of Sciences, Hangzhou Dianzhi University, Hangzhou, China
[2] Department of Mechanical Engineering, City University of Hong Kong, Hong Kong, China.

Abstract

We propose a Gauss-Seidel progressive iterative approximation (GS-PIA) method for Loop subdivision surface interpolation by combining classical Gauss-Seidel iterative method for linear system and progressive iterative approximation (PIA) for data interpolation. We prove that GS-PIA is convergent by applying matrix theory. GS-PIA algorithm retains the good features of the classical PIA method, such as the resemblance with the given mesh and the advantages of both a local method and a global method. Compared with some existed interpolation methods of subdivision surfaces, GS-PIA algorithm has advantages in three aspects. First, it has a faster convergence rate compared with the PIA and WPIA algorithms. Second, compared with WPIA algorithm, GS-PIA algorithm need not to choose weights. Third, GS-PIA need not to modify the mesh topology compared with other methods with fairness measures. Numerical examples for Loop subdivision surfaces interpolation illustrated in this paper show the efficiency and effectiveness of GS-PIA algorithm.

CCS Concepts

•***Computing methodologies*** $\rightarrow$ ***Parametric curve and surface models;***

1 Introduction

Subdivision defines a smooth curve or surface as the limit of sequence of successive refinements [ZS00]. Subdivision schemes can be divided into interpolatory and approximating schemes. To interpolate given meshes, we can use many interpolatory subdivision schemes [DLG90, ZSS96, KL10, Lev99, DM13]. However, the limit surface of interpolatory subdivision schemes may exhibit distortions [ZSS96] and may need bigger local supporter regions to generate surface with higher continuity. So sometimes it can not meet the quality requirements of practical applications.

In order to obtain interpolation surfaces with high quality, scholars have successively proposed various methods for interpolation surfaces with high quality [ZS00] based on approximating subdivision schemes. Hoppe et al. [HDD*94] and Nasri [NAS87] forced the subdivision surfaces to interpolate a particular set of control points by modifying the rules of subdivision. In addition, Bruner [Bru88] also defined a set of scalar shape handles, which effectively improved the quality of interpolation surface. For Catmull-Clark subdivion surface interpolation, based on minimizing of a certain fairness, Halstead et al. [HKD93] presented an efficient method. However, both Nasri's and Halstead et al.'s methods need to solve a linear equations. When the given mesh has too many vertices or the linear equations are singular or ill-conditioned [ZSS96, HKD93], that is troublesome. In the literature, one may also find methods for least squares fitting of a subdivision surface to a dense input mesh, in which the number of control vertices of the resulting subdivision surface is substantially less than that of the initial input dense mesh. Ma and Zhao [MZ00, MZ02] presented a method for least squares fitting of Catmull-Clark subdivision surfaces based on parametric evaluation of subdivision surfaces by Stam [Sta98]. Ma et al. [MMTP04] also presented another method for Loop subdivision surface fitting by simultaneously applying both the subdivision and limit position masks to a topology- and feature-preserving simplified mesh of the input dense mesh. While both of the above least squares fitting solutions produce stable results, they all lead to the solution of a linear least squares system that may be expensive to solve, especially if the resulting subdivision surface is to be defined by a large control mesh.

To avoid solving linear equations, Litke et al. [LLS01] proposed a fast and local algorithm for fitting a Catmull-Clark subdivision surface. Zheng and Cai [ZC06] proposed the two-phase subdivision scheme for Catmull-Clark subdivision to interpolate meshes with arbitrary topology. By introducing extra freedom of the variables in the first step subdivision to force the coefficient matrix of the linear system strictly diagonally dominant, it can be quickly solved by more efficient iterative methods. In addition, using the excess degrees of freedom, Zheng et al. defined some shape parameters to control surface. However, the added vertices, edges and faces in the first subdivision step lead to more surface patches that

† Corresponding author: dcy@hdu.edu.cn

DOI: 10.2312/pg.20181284

may bring inconvenience in surface modeling. A similar algorithm is also applied to the Doo-Sabin subdivision scheme [ZC05]. By modifying the geometric rules of the first step of Catmull-Clark subdivision, Deng and Yang [DY10] used a simple and efficient method to derive interpolation surface. However, the result surface also have more surface patches than the origin mesh implies. Later, Deng [Den10] and Deng and Wang [DW10] also extended this method to $\sqrt{3}$ and Loop subdivisions, respectively.

Similar to the progressive and iterative approximation (PIA) method of B-spline curves and surfaces [LWD04], the iterative method which avoids solving linear system has also been applied in subdivision surface interpolation. Chen et al. [CLT*08] and Cheng et al. [CFL*09] proposed the progressive interpolation approximation (PIA) algorithm for Catmull-Clark and Loop subdivision, respectively. Recently, Deng and Ma [DM12] proposed the weighted progressive interpolation approximation (WPIA) method for Loop subdivision surfaces that can effectively control the convergent rate by selecting weights based on experiments or theoretical analysis.

Combining the PIA algorithm and Gauss-Seidel method, in this paper we propose the Gauss-Seidel progressive and iterative approximation (GS-PIA) algorithm for Loop [Loo87] subdivision surfaces interpolation. The convergence of GS-PIA is proofed according to matrix theory. Compared with some existed interpolation method of subdivision surfaces, GS-PIA algorithm has advantages in three aspects. First, it has a faster convergence rate compared with the PIA and WPIA algorithms. Second, compared with WPIA, GS-PIA need not to choose weights. Third, GS-PIA need not to modify the mesh topology compared with other methods [ZC06, DY10] with fairness measures.

2 Loop subdivision surfaces

Loop subdivision scheme defined over triangular mesh generates new vertex points and new edge points in refinement. Loop subdivision surfaces is the generalization of box-spline surfaces.

2.1 The rules for Loop subdivision

Given a triangular mesh with a set of vertices $\{v_i\}$. The process for each refinement iteration includes:

I. The new vertex points: for each vertex point v_i, compute the new vertex point v_i' by a linear combination of the points within the neighborhood of v_i (see Figure 1(a))

$$v_i' = (1 - n_i\beta_{n_i})v_i + \beta_{n_i}\sum v_{i,j},$$

where n_i is the valence of v_i, and $\beta_{n_i} = \frac{1}{n_i}(\frac{5}{8} - (\frac{3}{8} + \frac{1}{4}\cos\frac{2\pi}{n_i}))$.

II. The new edge points: if the triangles (v_0, v_1, v_2) and (v_0, v_1, v_3) share the same edge v_0v_1, then the new edge point is computed as (see Figure 1(b))

$$v_e = \frac{3}{8}(v_0 + v_1) + \frac{1}{8}(v_2 + v_3).$$

III. Connect each new vertex point to surrounding new edge points (see Figure 1(c)).

IV. Connect each new edge point to the new edge points of adjacent edges (see Figure 1(c)).

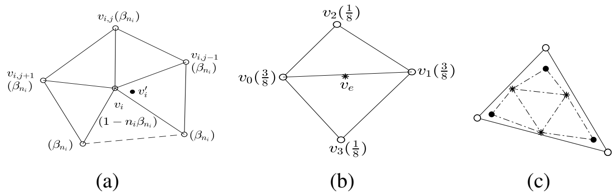

Figure 1: *Loop subdivision rules (old vertex ○): (a) new vertex point ●; (b) new edge point ∗; (c) refined mesh construction.*

By repeating the above process (I-IV), it yields a sequence of refined meshes which eventually converges to a limit surface, known as the Loop surface.

2.2 Formula of the limit point for Loop subdivision surfaces

For each vertex v_i in mesh M, assuming its valence is n_i, its limit point is

$$v_{i,\infty} = \alpha_i v_i + Q_i. \tag{1}$$

where

$$\alpha_i = \frac{3}{11 - 8 \times (\frac{3}{8} + (\frac{3}{8} + \frac{1}{4}cos\frac{2\pi}{n_i})^2)}, Q_i = \frac{(1-\alpha_i)\sum v_{i,j}}{n_i}, \tag{2}$$

the points $v_{i,j}$ are adjacent vertices of v_i .

3 The GS-PIA algorithm for Loop subdivision surfaces interpolation

Given an initial mesh M with a set of vertices $\boldsymbol{V} = \{v_i\}_{i=0}^m$, we want to design a algorithm (Algorithm 1) for constructing Loop surface that efficiently interpolates the given mesh.

3.1 The GS-PIA algorithm

Let $\boldsymbol{V^0} = \{v_i^0 = v_i\}_{i=0}^m$ be the original vertices of the mesh $M^0 = M$, and compute the limit point $v_{0,\infty}^0$ of v_0^0 on M^0 by Eq. (1). Denote that M^k is the mesh that we get from the original mesh after k steps of modification, where their vertices are $\boldsymbol{V^k} = \{v_i^k\}_{i=0}^m$. And let $\boldsymbol{V_i^k} = \{v_0^{k+1}, \cdots, v_{i-1}^{k+1}, v_i^k, \cdots, v_m^k\}$ correspond to the mesh M_i^k, where $0 < i \leq m$.

Firstly, denote the adjustment vector as

$$d_0^0 = \frac{v_0^0 - v_{0,\infty}^0}{\alpha_0}.$$

Let

$$v_0^1 = v_0^0 + d_0^0.$$

Then let $\boldsymbol{V_i^0} = \{v_0^1, \cdots, v_{i-1}^1, v_i^0, \cdots, v_m^0\}_{i=0}^m$ be the vertices of the mesh M_i^0 , and compute the limit point $v_{i,\infty}^0$ of v_i^0 on M_i^0 as

$$v_{i,\infty}^0 = \alpha_i v_i^0 + Q_i^0,$$

where Q_i^0 is the value of the Eqs. (2) for the mesh M_i^0.

At the same time, we also have

$$d_i^0 = \frac{v_i^0 - v_{i,\infty}^0}{\alpha_i}, v_i^1 = v_i^0 + d_i^0.$$

When $i = m$ holds, we set the vertices of the mesh M^1 to $\boldsymbol{V^1} = \{v_j^1\}_{j=0}^m$, where

$$v_j^1 = v_j^0 + d_j^0, j = 0, 1, \cdots, m.$$

At this point, we complete the first iteration.

As the k increases, the corresponding adjustment vectors are written as

$$d_i^k = \frac{v_i^0 - v_{i,\infty}^k}{\alpha_i}, i = 0, 1, 2, \cdots,$$

where $v_{i,\infty}^k = \alpha_i v_i^k + Q_i^k$, and Q_i^k is the value of the Eqs. (2) for the mesh M_i^k.

For the Loop subdivision scheme from Eqs. (1), we have

$$v_i^{k+1} = v_i^k + d_i^k = \frac{\alpha_i v_i^k + v_i^0 - v_{i,\infty}^k}{\alpha_i} = \frac{v_i^0 - Q_i^k}{\alpha_i}. \quad (3)$$

In this way, we get a mesh sequence $\{M^k\}(k = 1, 2, \cdots)$. In the following Sect. 3.2, we will prove that as k tends to infinity, M^k converges to M^∞ whose Loop surface interpolates the all original vertices of given mesh M.

Algorithm 1 The GS-PIA algorithm for Loop subdivision surface

---

Input: Given the vertices $\{v_i\}_{i=0}^m$ of the initial mesh M.

Output: The new vertices $\{v_i^\infty\}_{i=0}^m$ of the limit mesh M^∞.

Initializa: $v_i^0 = v_i, i = 0, 1, \cdots, m.$

for $k = 0, 1, 2, \cdots$ **do**

  for $i = 0$ to m **do**

    $v_{i,\infty}^k = \alpha_i v_i^k + Q_i^k$, $d_i^k = \frac{v_i^k - v_{i,\infty}^k}{\alpha_i}$, $v_i^{k+1} = v_i^k + d_i^k$.

  end for

end for

3.2 Convergence of the GS-PIA algorithm

Firstly, we define a matrix $\boldsymbol{B}$, where

$$\boldsymbol{B} = \begin{pmatrix} \alpha_{n_1} & \cdots & \frac{1-\alpha_1}{n_1} & \cdots \\ \vdots & \ddots & & \\ \frac{1-\alpha_i}{n_i} & & \alpha_i & \cdots \\ \vdots & \cdots & & \alpha_m \end{pmatrix}. \quad (4)$$

Then we can get

$$\boldsymbol{V}_\infty^k = \boldsymbol{B}\boldsymbol{V^k}. \quad (5)$$

By Eq. (3), the GS-PIA format can be written in matrix form as

$$\begin{aligned} \boldsymbol{V}^{k+1} &= \boldsymbol{D}^{-1}(\boldsymbol{V^0} - \boldsymbol{L}\boldsymbol{V}^{k+1} - \boldsymbol{U}\boldsymbol{V}^k), \\ &= -(\boldsymbol{L}+\boldsymbol{D})^{-1}\boldsymbol{U}\boldsymbol{V^k} + (\boldsymbol{L}+\boldsymbol{D})^{-1}\boldsymbol{V^0}, \end{aligned} \quad (6)$$

where $\boldsymbol{B}$ is decomposed into diagonal matrix $\boldsymbol{D}$, lower and upper triangular matrices $\boldsymbol{L}$ and $\boldsymbol{U}$, i.e. $\boldsymbol{B} = \boldsymbol{D} + \boldsymbol{L} + \boldsymbol{U}$.

By subtracting $\boldsymbol{B}_{\boldsymbol{L}}^{-1}\boldsymbol{V^0}$ both sides of Eq. (6), we have

$$\boldsymbol{V}^{k+1} - \boldsymbol{B}^{-1}\boldsymbol{V^0} = (-(\boldsymbol{L}+\boldsymbol{D})^{-1}\boldsymbol{U})^{k+1}(\boldsymbol{V}^k - \boldsymbol{B}^{-1}\boldsymbol{V^0}). \quad (7)$$

Then denote error vector

$$\delta^k = \boldsymbol{V}^k - \boldsymbol{B}^{-1}\boldsymbol{V}^0$$

We can get

$$\delta^{k+1} = (-(\boldsymbol{L}+\boldsymbol{D})^{-1}\boldsymbol{U})^{k+1}\delta^k.$$

Denote $\boldsymbol{G_B} = -(\boldsymbol{L}+\boldsymbol{D})^{-1}\boldsymbol{U}$ and $\rho(\boldsymbol{G_B})$ be the spectral radius of the matrix $\boldsymbol{G_B}$. If we can get $\rho(\boldsymbol{G_B}) < 1$, then the GS-PIA algorithm is convergent for Loop subdivision.

Lemma 1 If the coefficient matrix $\boldsymbol{A}$ of the linear system of equations $\boldsymbol{Ax} = \boldsymbol{b}$ is Hermite positive definite, the Gauss-Seidel iteration of the matrix $\boldsymbol{A}$ is convergent, namely $\rho(\boldsymbol{G_A}) < 1$ [GVL96].

Theorem 1 The Gauss-Seidel iterative interpolation process for Loop subdivision surface interpolation is convergent.

Proof Let $b_i = \frac{1-\alpha_i}{n_i}(i = 1, \cdots, m)$ and $\boldsymbol{B} = \boldsymbol{MN}$, where

$$\boldsymbol{M} = \begin{pmatrix} b_1 & & \\ & \ddots & \\ & & b_m \end{pmatrix}, \boldsymbol{N} = \begin{pmatrix} \frac{\alpha_1}{b_1} & \cdots & 1 & \cdots \\ \vdots & \ddots & & \\ 1 & & \frac{\alpha_i}{b_i} & \cdots \\ \vdots & & & \frac{\alpha_m}{b_m} \end{pmatrix}. \quad (8)$$

Then we have

$$\begin{aligned} \boldsymbol{G_B} &= -(\boldsymbol{L}+\boldsymbol{D})^{-1}\boldsymbol{U} = -(\boldsymbol{M}\boldsymbol{L_N} + \boldsymbol{M}\boldsymbol{D_N})^{-1}\boldsymbol{M}\boldsymbol{U_N} \\ &= -(\boldsymbol{L_N} + \boldsymbol{D_N})^{-1}\boldsymbol{U_N} = \boldsymbol{G_N}. \end{aligned}$$

It is known from [CFL*09] the matrix $\boldsymbol{N}$ is positive definite Hermite matrix. Therefore using Lemma 1, we have $\rho(\boldsymbol{G_B}) = \rho(\boldsymbol{G_N}) < 1$. Then the iterative process of GS-PIA is convergent for the Loop subdivision surfaces. □

Table 1: *The statistics of the Loop surface interpolation.*

Examples	Vertex	Valences	Spectral radius			$\frac{E_{10}}{E_0}$		
			PIA	W-PIA	GS-PIA	PIA	W-PIA	GS-PIA
1	477	$4, \cdots, 13$	0.769	0.624	0.312	0.028	0.002	1.08×10^{-5}
2	1618	$3, \cdots, 11$	0.783	0.643	0.267	0.017	0.001	2.12×10^{-6}
3	8604	$3, \cdots, 24$	0.815	0.688	0.341	0.018	0.002	1.10×10^{-5}

4 Examples

In this section, we give some specific examples to illustrate the interpolating results by GS-PIA method. Define the error after iteration k steps as $e_i^k = \|v_i^0 - v_{i,\infty}^k\|$, $E_k = max_i\{e_i^k\}$.

First of all, for the Loop subdivision scheme, we compare PI-A [CFL*09] algorithm and W-PIA [DM12] algorithm with GS-PIA algorithm. Note that in next example, we all use the optimal weights for W-PIA. Three examples are presented in Figure 2. We can see the surface generated by the GS-PIA algorithm faithfully resembles

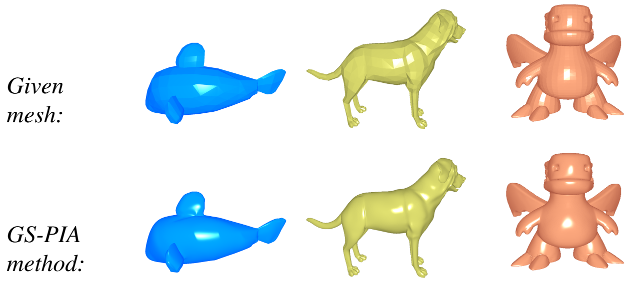

Figure 2: *Loop surfaces interpolation.*

the shape of the original mesh. Besides the basic information of the original mesh in Figure 2, Table 1 describes the spectral radius of the iterative matrix corresponding to the three methods and the ratio $\frac{E_{10}}{E_0}$ after iteration 10 times for loop subdivision surface. We can see that the iteration matrix of the GS-PIA algorithm has a smaller spectral radius and therefore the convergent rate is much more effective. More importantly, GS-PIA is simpler than W-PIA because it does not need to select a weight.

5 Conclusions

We have presented a GS-PIA interpolation method for Loop subdivision surfaces. GS-PIA is a simple, stable and efficient surface interpolation method that can be applied to meshes with arbitrary topology. As an iterative method, the convergence of the GS-PIA is proved. In addition, the GS-PIA algorithm does not modify the initial mesh topology and has a faster convergence rate. In the future, we will continue to investigate the convergence of GS-PIA algorithm for the interpolation of other subdivision surfaces.

6 Acknowledgments

This work has been supported by the NSFC (61502128, 61872121, 61370166), Natural Science Foundation of Zhejiang Province (LQ17A010009) and City University of Hong Kong (S-RG 7004452).

References

[Bru88] BRUNET P.: Including shape handles in recursive subdivision surfaces. *Computer Aided Geomertic Design. 5*, 1 (June 1988), 41–50. 1

[CFL*09] CHENG F. H., FAN F. T., LAI S. H., HUANG C. L., WANG J. X., YONG J. H.: Loop subdivision surface based progressive interpolation. *Journal of Computer Science and Technology 24*, 1 (2009), 39–46. 2, 3

[CLT*08] CHEN Z., LUO X., TAN L., YE B., CHEN J.: Progressive Interpolation based on Catmull-Clark Subdivision Surfaces. *Computer Graphics Forum* (2008). 2

[Den10] DENG C. Y.: Interpolating closed triangular meshes by approximation $\sqrt{3}$ subdivision scheme. *Journal of Computer-Aided Design & Computer Graphics 22*, 2 (2010), 312–317. 2

[DLG90] DYN N., LEVINE D., GREGORY J. A.: A butterfly subdivision scheme for surface interpolation with tension control. *ACM Trans. Graph. 9*, 2 (Apr. 1990), 160–169. 1

[DM12] DENG C., MA W.: Weighted progressive interpolation of loop subdivision surfaces. *Computer-Aided Design 44*, 5 (2012), 424–431. 2, 3

[DM13] DENG C., MA W.: A unified interpolatory subdivision scheme for quadrilateral meshes. *ACM Trans. Graph. 32*, 3 (2013), 23:1–11. 1

[DW10] DENG C. Y., WANG G. Z.: Interpolating triangular meshes by loop subdivision scheme. *Science China(Information Sciences) 53*, 9 (2010), 1765–1773. 2

[DY10] DENG C. Y., YANG X. N.: A simple method for interpolating meshes of arbitrary topology by catmull-clark surfaces. *Vis. Comput. 26*, 2 (Jan. 2010), 137–146. 2

[GVL96] GOLUB G. H., VAN LOAN C. F.: *Matrix Computations (3rd Ed.)*. Johns Hopkins University Press, Baltimore, MD, USA, 1996. 3

[HDD*94] HOPPE H., DEROSE T., DUCHAMP T., HALSTEAD M., JIN H., MCDONALD J., SCHWEITZER J., STUETZLE W.: Piecewise smooth surface reconstruction. In *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 1994), SIGGRAPH '94, ACM, pp. 295–302. 1

[HKD93] HALSTEAD M., KASS M., DEROSE T.: Efficient, fair interpolation using catmull-clark surfaces. In *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 1993), SIGGRAPH '93, ACM, pp. 35–44. 1

[KL10] KOBBELT, LEIF: Interpolatory subdivision on open quadrilateral nets with arbitrary topology. *Computer Graphics Forum 15*, 3 (2010), 409–420. 1

[Lev99] LEVIN A.: Interpolating nets of curves by smooth subdivision surfaces. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 1999), SIGGRAPH '99, ACM Press/Addison-Wesley Publishing Co., pp. 57–64. 1

[LLS01] LITKE N., LEVIN A., SCHRÖDER P.: Fitting subdivision surfaces. In *Proceedings of the Conference on Visualization '01* (Washington, DC, USA, 2001), VIS '01, IEEE Computer Society, pp. 319–324. 1

[Loo87] LOOP C.: Smooth subdivision surfaces based on triangles. *Masters Thesis University of Utah Department of Mathematics* (1987). 2

[LWD04] LIN H., WANG G., DONG C.: Constructing iterative non-uniform b-spline curve and surface to fit data points. *Science in China 47*, 3 (2004), 315–331. 2

[MMTP04] MA W., MA X., TSO S. K., PAN Z.: A direct approach for subdivision surface fitting from a dense triangle mesh. *Computer-Aided Design 36*, 6 (2004), 525–536. 1

[MZ00] MA W., ZHAO N.: Catmull-clark surface fitting for reverse engineering applications. In *Proceedings of Geometric Modeling and Processing 2000* (2000), IEEE Computer Society, pp. 274–284. 1

[MZ02] MA W., ZHAO N.: Smooth multiple b-spline surface fitting with catmull-clark subdivision surfaces for extraordinary corner patches. *The Visual Computer 18*, 7 (2002), 415–436. 1

[NAS87] NASRI A. H.: Polyhedral subdivision methods for free-form surfaces. *ACM Trans. Graph. 6*, 1 (1987), 29–73. 1

[Sta98] STAM J.: Exact evaluation of catmull-clark subdivision surfaces at arbitrary parameter values. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 1998), SIGGRAPH '98, ACM, pp. 395–404. 1

[ZC05] ZHENG J., CAI Y.: Making doo-sabin surface interpolation always work over irregular meshes. *Visual Computer 21*, 4 (2005), 242–251. 2

[ZC06] ZHENG J., CAI Y.: Interpolation over arbitrary topology meshes using a two-phase subdivision scheme. *IEEE Transactions on Visualization and Computer Graphics 12*, 3 (May 2006), 301–310. 1, 2

[ZS00] ZORIN D., SCHRÖDER P.: Subdivision for modeling and animation. *Course Notes of SIGGRAPH* (2000). 1

[ZSS96] ZORIN D., SCHRÖDER P., SWELDENS W.: Interpolating subdivision for meshes with arbitrary topology. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 1996), SIGGRAPH '96, ACM, pp. 189–192. 1