

# Non-Linear Subdivision using Local Spherical Coordinates

Nicolas Aspert,<sup>1</sup> Touradj Ebrahimi and Pierre Vanderghyest

*Signal Processing Institute  
Swiss Federal Institute of Technology (EPFL)  
1015 Lausanne  
Switzerland*

---

## Abstract

In this paper, we present an original non-linear subdivision scheme suitable for univariate data, plane curves and discrete triangulated surfaces, while keeping the complexity acceptable. The proposed technique is compared to linear subdivision methods having an identical support. Numerical criteria are proposed to verify basic properties, such as convergence of the scheme and the regularity of the limit function.

*Key words:* non-linear subdivision, discrete surface, univariate data

---

## 1 Introduction

Subdivision is a powerful tool, widely used in many different areas, ranging from computer-aided design to data compression. Recent works have shown the close relationship between subdivision operators and multiresolution analysis. In particular, subdivision is often regarded as a basic block used to build multiresolution-like representations for discrete surfaces. However, adapting the univariate subdivision schemes to discrete surfaces is not straightforward, even when dealing with linear schemes. As shown for instance in [27], even a simple univariate linear scheme requires non-negligible adaptation to handle correctly irregular meshes. In this paper, we propose a univariate non-linear

---

<sup>1</sup> Corresponding author :

E-Mail: [Nicolas.Aspert@epfl.ch](mailto:Nicolas.Aspert@epfl.ch)

Tel: +41 21 693 3632 Fax: +41 21 693 7600

subdivision scheme that can be adapted rather easily to discrete surfaces, and that does not require specific adaptation in the case of irregular meshes.

The paper is organized as follows: section 2 presents the existing univariate and surface subdivision techniques. Section 3 describes the details of the proposed method for univariate data and plane curves. The extension for triangular meshes is presented in section 4, and the results obtained are described and analyzed in section 5.

## 2 Review of subdivision

In this section, we review some key concepts of of subdivision for both univariate data and triangular meshes.

Univariate data is the simplest kind of data to study, nevertheless is the basis for generalizations of subdivision algorithms to higher-dimensional data. Subdivision aims at refining univariate data iteratively, i.e. produce a smooth limit curve from a discrete initial dataset (often termed *control polygon*). Discrete univariate data can be described by a mapping between  $k \in \mathbb{Z}$  and sample points  $f_k \in \mathbb{R}^d$ . By extension, we assimilate the data as the piecewise linear function linking consecutive samples  $f_k$  and  $f_{k+1}$ . In order to distinguish the different levels, the  $k$ -th sample at level  $j$  will be denoted by  $f_k^j$ . Usually, only  $n$ -ary refinement operations are considered, i.e. the data at level  $j$  is a mapping between  $kn^{-j}$  and  $f_k^j$ . In this study, we will only consider binary subdivision, which is so far the only one adapted to surfaces, although recent works [19,11] open the way toward ternary surface subdivision.

Linear subdivision techniques have been widely studied and used (see for instance [2] or [12] for complete studies). The idea is to used a linear combination of the samples of level  $j$  to generate samples of level  $j + 1$ , i.e.

$$f_k^{j+1} = \sum_{p \in \mathbb{Z}} a_{k-2p}^j f_p^j, \quad (1)$$

where  $\{a_k\}_{k \in \mathbb{Z}}$  is a set of real numbers, also called *mask*. For obvious complexity reasons, only finite support masks are considered (which leads to a finite sum in (1)). Historically, the refinement operations that were considered only involved stationary and uniform masks, i.e. coefficients not depending on  $j$  and remaining invariant through even integer translations. In such a case, the refinement relation is fully described by two relations (one for odd sample and one for even samples).

For instance, the ‘‘corner-cutting’’ rule proposed by Georges de Rham [8,9]

and re-proposed later by Chaikin [3] can be written as

$$\begin{aligned} f_{2k}^{j+1} &= (1 - \alpha)f_k^j + \alpha f_{k+1}^j, \\ f_{2k+1}^{j+1} &= \beta f_k^j + (1 - \beta)f_{k+1}^j, \end{aligned} \tag{2}$$

with  $\alpha > 0$ ,  $\beta > 0$  and  $\alpha + \beta < 1$ . De Rham’s initial choice was  $\alpha = \beta = \frac{1}{3}$ , which only produces  $\mathcal{C}^0$  limit curves, whereas Chaikin used  $\alpha = \beta = \frac{1}{4}$ , which gives  $\mathcal{C}^1$  limit curves (in fact Chaikin’s rule produces quadratic B-spline curves). Another category of interest is interpolating subdivision, which implies that the samples at level  $j$  also belong to level  $j + 1$ . A famous example is the *four-point scheme* proposed by N. Dyn, J. Gregory and D. Levin in [13]

$$\begin{aligned} f_{2k}^{j+1} &= f_k^j, \\ f_{2k+1}^{j+1} &= -w f_{k-1}^j + \left(\frac{1}{2} + w\right) f_k^j + \left(\frac{1}{2} + w\right) f_{k+1}^j - w f_{k+2}^j, \end{aligned} \tag{3}$$

where  $w$  is a real. The choice  $w = \frac{1}{16}$  is of interest since for this value the scheme coincides with the cubic Lagrange-based scheme proposed in [10], and satisfies the necessary condition to achieve  $\mathcal{C}^2$  limit functions, although it only leads to  $\mathcal{C}^{2-\varepsilon}$  functions, as shown for instance in [7].

Univariate subdivision techniques have been extended to higher-dimensional data, e.g. using tensor-product schemes, which were used as basis for *surface subdivision*. The most generic representation of discrete surfaces is achieved through *triangular meshes*. In this study, we will focus on discrete surfaces represented by triangular meshes, i.e. by a couple  $(\mathcal{V}, \mathcal{T})$  where  $\mathcal{V}$  is a set of  $N$  vertices  $\mathcal{V} = \{v_i \in \mathbb{R}^3 | 0 \leq i \leq N - 1\}$  and  $\mathcal{T}$  is a set of triplets  $(i_1, i_2, i_3) \in \{0 \dots N - 1\}^3$  representing the indices of vertices forming a triangle. When subdividing a triangular mesh, each edge is cut into two new edges and each triangle is therefore split into four new triangles (recently, another way of splitting triangular meshes has been proposed in [21], only increasing the number of triangles by a factor 3 instead of 4 but in our case, we will only consider the “quadrisection” of triangles). Splitting a mesh with this technique leads to *semi-regular* meshes, i.e. having only vertices of valence 6, except in isolated places. As for the univariate case, the notions of interpolating subdivision, stationarity, linearity and data-independence remain valid. The translation invariance would be more difficult to express, given that each vertex can have a different number of neighbors. In the case of linear schemes, a weighted sum of the vertices belonging to the 1-ring is performed to compute the new vertex position. The “Butterfly” scheme belongs to the class of stationary, interpolating and linear subdivision schemes. While performed on regular triangular meshes, it can be seen as a tensor product of the four-point scheme. It has been first presented in [14] and then extended to irregular meshes in [27], and generates  $\mathcal{C}^1$  surfaces. Another example is the “Loop” scheme [25], which is not interpolating, but gives a smoother limit surface.

Whereas many studies concerning linear and stationary subdivision operators can be found, works concerning non-linear and/or non-stationary/non-uniform cases remain relatively sparse. Some results about the non-stationary linear case have been shown in [15,4]. Several works concerning particular non-linear schemes have also been performed. In [17], the case of schemes made of a linear part and a non-linear part (in this case the harmonic mean of a linear combination of sample) is addressed. A similar technique was developed in [23] and [24] to design convexity and monotonicity preserving schemes. Another approach has been proposed, based on the essentially non-oscillatory (ENO) methods in [5].

While most of the stationary linear techniques have been extended to discrete triangulated surfaces, non-linear surface subdivision schemes are not common. An method of convexity-preserving surface subdivision has been proposed in [16]. Another interpolating approach has been described in [20], which resembles by some aspects the proposed scheme. However, according to the author, the results achieved in terms of smoothness of the limit surface are not satisfactory and require a smoothing step.

The non-linear subdivision technique described in sections 3 and 4 aims at bypassing the adaptations that have to be performed when trying to deal with irregular meshes, while keeping a four-point support in the univariate case. The local coordinate system on which the method is based makes the scheme easily adaptable to triangular meshes, regardless of the valence of the vertices.

### **3 Non-linear subdivision using local spherical coordinates for univariate data**

In this section, a non-linear subdivision algorithm for univariate data is presented. The derived surface subdivision algorithm will be presented in section 4. We first present the details of the subdivision algorithm for discrete univariate functions defined on strictly increasing grids, i.e. applied to data samples  $(x_k^j, f_k^j) \in \mathbb{R} \times \mathbb{R}^d$ , verifying  $x_k^j < x_{k+1}^j$ , for all  $(j, k) \in \mathbb{N} \times \mathbb{Z}$ .

#### *3.1 Subdivision for strictly increasing grids*

In order to perform the proposed subdivision technique, we need to define a local coordinates system around each point, as well as several local parameters, which are detailed in this section. Those definitions are illustrated by figure 1. When possible, the level index  $j$  will be omitted. We will assume

that the samples  $f_k^j$  belong to  $\mathbb{R}$  although the proposed method can be trivially extended to  $\mathbb{R}^d$ . Let  $\mathcal{R}$  be the canonical basis for  $\mathbb{R}^2$ . Let us denote by  $P_k$  the point of the plane of coordinates  $(x_k, f_k)$  in  $\mathcal{R}$ . In the following, we will denote the Euclidean distance between the points  $P_k$  and  $P_{k+1}$  by  $r_k = \sqrt{(x_{k+1} - x_k)^2 + (f_{k+1} - f_k)^2}$ . Since we have to deal with more than one coordinate system, we will add the name of the basis as a superscript when needed to avoid confusion, e.g.  $\mathbf{v}^{\mathcal{B}}$  will denote the coordinates of vector  $\mathbf{v}$  in the basis  $\mathcal{B}$ .

In order to define the angles that will be used in our method, we need to define a local coordinate system at  $P_k$  by approximating the tangent and normal vectors at this point. An approximation of the *normal vector*  $\mathbf{n}_k$  at point  $P_k$  can be obtained by averaging the unit normals to the edges  $P_{k-1}P_k$  and  $P_kP_{k+1}$ , weighted by the edges length  $r_{k-1}$  and  $r_k$ . The unit normal  $\mathbf{n}_{k-1,k}$  to the edge  $P_{k-1}P_k$  can be expressed (modulo the sign) as

$$\mathbf{n}_{k-1,k} = \frac{1}{r_{k-1}} \begin{pmatrix} f_{k-1} - f_k & x_k - x_{k-1} \end{pmatrix}^T,$$

which yields

$$\mathbf{n}_k = \frac{r_{k-1}\mathbf{n}_{k-1,k} + r_k\mathbf{n}_{k,k+1}}{\|r_{k-1}\mathbf{n}_{k-1,k} + r_k\mathbf{n}_{k,k+1}\|}. \quad (4)$$

Since

$$r_{k-1}\mathbf{n}_{k-1,k} + r_k\mathbf{n}_{k,k+1} = \begin{pmatrix} f_{k-1} - f_{k+1} & x_{k+1} - x_{k-1} \end{pmatrix}^T,$$

the expression of  $\mathbf{n}_k$  can be simplified as

$$\mathbf{n}_k = \sqrt{1 + f_k^{[1]s^2}} \begin{pmatrix} -f_k^{[1]s} & 1 \end{pmatrix}^T, \quad (5)$$

where  $f_k^{[1]s}$  denotes the *symmetrical* first order finite difference at  $P_k$ , i.e.

$$f_k^{[1]s} = \frac{f_{k+1} - f_{k-1}}{x_{k+1} - x_{k-1}}.$$

The computation of  $\mathbf{t}_k$  is similar and yields

$$\mathbf{t}_k = \frac{1}{\sqrt{1 + f_k^{[1]s^2}}} \begin{pmatrix} 1 & f_k^{[1]s} \end{pmatrix}^T.$$

Without restricting the problem, we can assume that  $f_k^{[1]s}$  is bounded. Therefore, there exists a unique  $\theta_k \in \left(-\frac{\pi}{2}; \frac{\pi}{2}\right)$  such that  $f_k^{[1]s} = \tan \theta_k$  for all  $k$ . In fact,  $\theta_k$  is nothing but the angle between  $\mathbf{t}_k$  and the horizontal axis in  $\mathcal{R}$ .

Let us denote the local coordinates system at point  $P_k$  by  $\mathcal{R}_k = (\mathbf{t}_k, \mathbf{n}_k)$ , for all  $k$  and let  $\mathbf{v}_k$  be the vector from  $P_k$  to  $P_{k+1}$ . Since we have assumed that the

grid is strictly increasing, it is clear that there exists a unique  $\gamma_k \in \left(-\frac{\pi}{2}; \frac{\pi}{2}\right)$  such that  $\mathbf{v}_k^{\mathcal{R}} = r_k \begin{pmatrix} \cos \gamma_k & \sin \gamma_k \end{pmatrix}^T$ .  $(r_k, \gamma_k)$  are the polar coordinates of  $P_{k+1}$  in the canonical basis centered at  $P_k$ . We can now rewrite the expression of  $\mathbf{v}_k$  in the  $\mathcal{R}_k$  basis. There exists a unique  $\alpha_k \in (-\pi; \pi)$  such that  $\mathbf{v}_k^{\mathcal{R}_k} = r_k \begin{pmatrix} \cos \alpha_k & \sin \alpha_k \end{pmatrix}^T$  and  $\alpha_k$ ,  $\gamma_k$  and  $\theta_k$  are linked through the following relation

$$\gamma_k = \alpha_k + \theta_k. \quad (6)$$

Similarly, we denote by  $\mathbf{w}_k$  the vector from  $P_k$  to  $P_{k-1}$  and there exists a unique  $\beta_k \in (-\pi; \pi)$  such that  $\mathbf{w}_k^{\mathcal{R}_k} = -r_{k-1} \begin{pmatrix} \cos \beta_k & \sin \beta_k \end{pmatrix}^T$ , and a relation similar to (6) holds

$$\gamma_{k-1} = \beta_k + \theta_k. \quad (7)$$

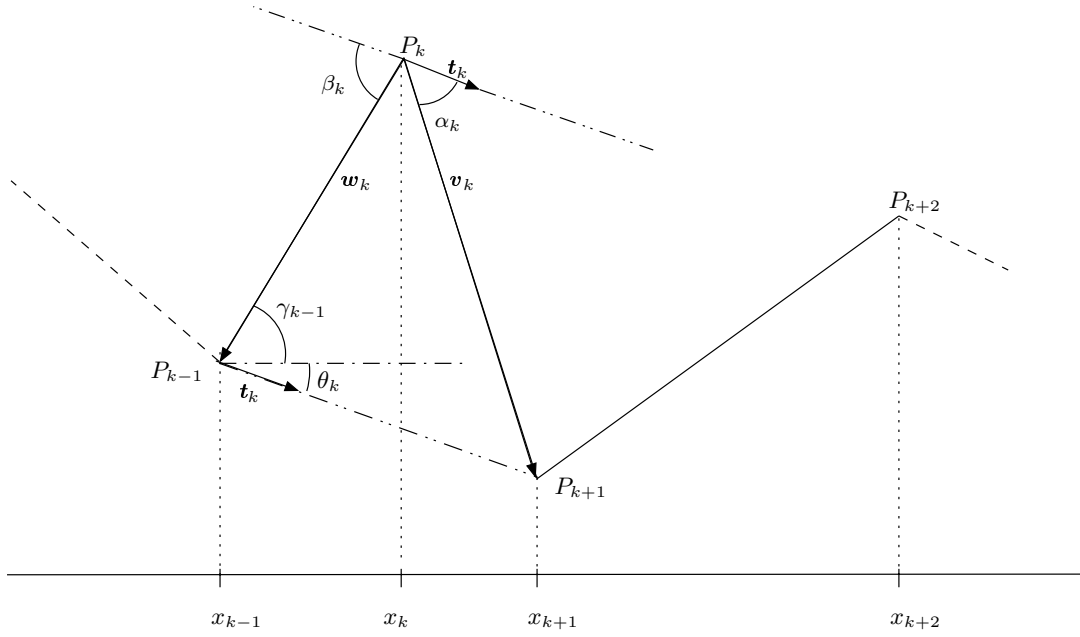


Fig. 1. Local coordinates system and angles at  $P_k$ .

Using the angles and local coordinate system defined above, we can now describe the algorithm to build the level  $j+1$  from the samples at level  $j$ . As many popular subdivision algorithms, the proposed method belongs to the class of binary *interpolating* subdivision schemes. Therefore, we will have  $x_k^j = x_{2k}^{j+1}$  and  $f_k^j = f_{2k}^{j+1}$ .

Let us now describe how the new points of level  $j+1$  are obtained. In the following, all steps necessary to compute the midpoint of edge  $P_k^j P_{k+1}^j$  will be performed with  $r_k^j$  and  $\alpha_k^j$  in the  $\mathcal{R}_k^j$  system coordinate. A similar development can be made in  $\mathcal{R}_{k+1}^j$ , with  $r_k^j$  and  $\beta_{k+1}^j$ . Let us develop how the point  $P_{2k+1}^{j+1}$  (i.e. the “midpoint” of the edge  $P_k^j P_{k+1}^j$ ) is computed.

In  $\mathcal{R}_k^j$ , the neighboring points of  $P_k^j$  are uniquely defined by their local spherical (here polar) coordinates, i.e. by  $r_k^j$  and  $\alpha_k^j$ . A trivial interpolation, only resulting in a  $\mathcal{C}^0$  limit function, would be achieved by taking the middle of the edge  $P_k^j P_{k+1}^j$ . The coordinates in  $\mathcal{R}_k^j$  of such a midpoint would be  $(\frac{r_k^j}{2}, \alpha_k^j)$ . Another possible choice would be to use the ‘‘middle’’ of the edge in the parameter domain, i.e.  $(\frac{r_k^j}{2}, \frac{\alpha_k^j}{2})$ . While the choice of  $\frac{r_k^j}{2}$  for the first coordinate is rather logical (since we want the newly inserted point’s  $x$ -coordinate to lie close to the middle of the edge’s  $x$ -coordinates), the operation to be performed over the  $\alpha_k^j$  coordinate is less intuitive. Since the goal is to have a limit function that is the smoothest, i.e. to have the  $\alpha_k^j$  and  $\beta_k^j$  become as small as possible as  $j$  increases, thus the intuitive choice of  $\frac{\alpha_k^j}{2}$  for the second coordinate of the midpoint. While this solution would be elegant, it cannot be applied directly. In fact, the angle  $\alpha$  can vary over the whole range  $(-\pi; \pi)$  and cases where the principle of having a strictly increasing grid at each level would not be valid may occur. Since the  $x$ -coordinate of the new midpoint depends on  $\cos \alpha_k^j$ , a function having strong variations would lead to values of this quantity close to  $-1$ , finally leading to  $x_{2k+1}^{j+1} < x_{2k}^{j+1} = x_k^j$ , which is obviously an undesirable result. As a consequence, we have chosen to define the ‘‘midpoint’’ as the point of coordinates  $(\frac{r_k^j}{2}, h(\alpha_k^j))$ , where  $h$  is a  $\mathcal{C}^1$  piecewise polynomial function, defined as follows

$$h(\alpha) = \begin{cases} \alpha & \text{if } \pi < \alpha \leq -\frac{\pi}{2} \\ -\frac{1}{\pi} \left(\alpha + \frac{\pi}{4}\right)^2 \left[\frac{24}{\pi} \left(\alpha + \frac{\pi}{4}\right) + 10\right] + \frac{\alpha}{2} & \text{if } -\frac{\pi}{2} < \alpha < -\frac{\pi}{4} \\ \frac{\alpha}{2} & \text{if } -\frac{\pi}{4} \leq \alpha \leq \frac{\pi}{4} \\ \frac{1}{\pi} \left(\alpha - \frac{\pi}{4}\right)^2 \left[\frac{-24}{\pi} \left(\alpha - \frac{\pi}{4}\right) + 10\right] + \frac{\alpha}{2} & \text{if } \frac{\pi}{4} < \alpha < \frac{\pi}{2} \\ \alpha & \text{if } \pi > \alpha \geq \frac{\pi}{2} \end{cases} \quad (8)$$

Function  $h$  is a scaling by a factor  $\frac{1}{2}$  for small values of  $\alpha$  (i.e.  $|\alpha| \leq \frac{\pi}{4}$ ), and the identity when  $|\alpha| \geq \frac{\pi}{2}$ . Two cubic polynomials provide the  $\mathcal{C}^1$  link between these two parts. Figure 2 shows the aspect of  $h$ . The coordinates  $(\hat{x}_{2k+1}^{j+1}, \hat{f}_{2k+1}^{j+1})$  of this midpoint in  $\mathcal{R}_k^j$  are given by

$$\begin{pmatrix} \hat{x}_{2k+1}^{j+1} - x_k^j & \hat{f}_{2k+1}^{j+1} - f_k^j \end{pmatrix}^{\mathcal{R}_k^j} = \frac{r_k^j}{2} \begin{pmatrix} \cos h(\alpha_k^j) & \sin h(\alpha_k^j) \end{pmatrix}. \quad (9)$$

Another midpoint can be computed in a similar way, using  $\mathcal{R}_{k+1}^j$  and  $\beta_{k+1}^j$  instead of  $\mathcal{R}_k^j$  and  $\alpha_k^j$ , which yields

$$\begin{pmatrix} \tilde{x}_{2k+1}^{j+1} - x_{k+1}^j & \tilde{f}_{2k+1}^{j+1} - f_{k+1}^j \end{pmatrix}^{\mathcal{R}_{k+1}^j} = -\frac{r_k^j}{2} \begin{pmatrix} \cos h(\beta_{k+1}^j) & \sin h(\beta_{k+1}^j) \end{pmatrix}. \quad (10)$$

In order to be able to combine these two ways of computing the midpoint of

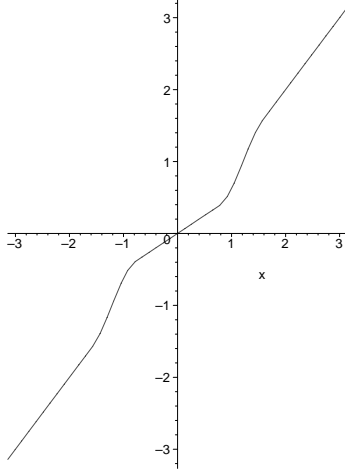


Fig. 2. Plot of the function  $h$  over the interval  $[-\pi; \pi]$

the edge  $P_k^j P_{k+1}^j$ , let us rewrite equations (9) and (10) in  $\mathcal{R}$ ,

$$\begin{aligned} \begin{pmatrix} \hat{x}_{2k+1}^{j+1} - x_k^j & \hat{f}_{2k+1}^{j+1} - f_k^j \end{pmatrix}^{\mathcal{R}} &= \frac{r_k^j}{2} \begin{pmatrix} \cos(\theta_k^j + h(\alpha_k^j)) & \sin(\theta_k^j + h(\alpha_k^j)) \end{pmatrix}, \\ \begin{pmatrix} \tilde{x}_{2k+1}^{j+1} - x_{k+1}^j & \tilde{f}_{2k+1}^{j+1} - f_{k+1}^j \end{pmatrix}^{\mathcal{R}} &= \\ &= -\frac{r_k^j}{2} \begin{pmatrix} \cos(\theta_{k+1}^j + h(\beta_{k+1}^j)) & \sin(\theta_{k+1}^j + h(\beta_{k+1}^j)) \end{pmatrix}. \end{aligned}$$

The most natural way to take into account these two ways of computing the midpoint is to average the two contributions. Therefore, the final expression of the midpoint in  $\mathcal{R}$  will be

$$\begin{aligned} x_{2k+1}^{j+1} &= \frac{\hat{x}_{2k+1}^{j+1} + \tilde{x}_{2k+1}^{j+1}}{2} = \frac{x_k^j + x_{k+1}^j}{2} \\ &+ \frac{r_k^j}{4} \left[ \cos(\theta_k^j + h(\alpha_k^j)) - \cos(\theta_{k+1}^j + h(\beta_{k+1}^j)) \right], \end{aligned} \quad (11)$$

$$\begin{aligned} f_{2k+1}^{j+1} &= \frac{\hat{f}_{2k+1}^{j+1} + \tilde{f}_{2k+1}^{j+1}}{2} = \frac{f_k^j + f_{k+1}^j}{2} \\ &+ \frac{r_k^j}{4} \left[ \sin(\theta_k^j + h(\alpha_k^j)) - \sin(\theta_{k+1}^j + h(\beta_{k+1}^j)) \right]. \end{aligned} \quad (12)$$

Those relations, along with the interpolation rule for even samples fully define the level  $j+1$  from the level  $j$ .

The proposed scheme is clearly non-linear, due to the sine and cosine terms added to a linear part. A consequence of the method used to compute the  $x_{j+1}$ , is that the grids generated are not uniform. However, numerical results tend to show that the grid gets closer to a uniform grid as the level increases. It is also non-stationary, in the sense that the computation of the midpoint depends on the data. However, the midpoint computation method is “translation invariant”, i.e. the same method is applied for all  $k$ . An obvious property



of this scheme is also its ability to reproduce polynomials up to degree 1, since in this case the non-linear part of equations (11) and (12) will be zero.

As stated in the previous section, the reason for having a function  $h$  instead of a scalar factor is to avoid pathological cases that may occur when the initial grid has sharp transitions (i.e. values of  $\gamma_k^j$  close to  $-\frac{\pi}{2}$  or  $\frac{\pi}{2}$ ). In order to illustrate what occurs in such cases, let us rewrite equation (11)

$$x_{2k+1}^{j+1} - x_{2k}^{j+1} = \frac{x_{k+1}^j - x_k^j}{2} + \frac{r_k^j}{4} \left[ \cos(\theta_k^j + h(\alpha_k^j)) - \cos(\theta_{k+1}^j + h(\beta_{k+1}^j)) \right].$$

One can construct cases where the quantity  $\cos(\theta_k^j + h(\alpha_k^j)) - \cos(\theta_{k+1}^j + h(\beta_{k+1}^j))$  becomes negative and may lead to a grid that is no longer increasing at the level  $j + 1$ . Numerical experiments showed that the introduction of  $h$  removed the irrelevant points that occurred when applying the scheme (with  $h(x) = \frac{x}{2}$ ) on highly-varying functions. We refer the reader to section 5 for the results obtained.

### 3.2 Non-linear subdivision for plane curves

The method proposed in the above section can be generalized to generic plane curves (e.g. using planar polygons as initial data). The computation of the normal vector at each sample point is done using equation (5). The closed form derived in equations (11) and (12) cannot be applied directly since assumptions regarding the monotonicity of the  $x_k^j$  have been done. The algorithm 1 describes the steps needed to compute the coordinates of the midpoint of each edge. The test performed to determine whether  $\|P_k^j M_{k+1}^j\|$  is greater than an  $\varepsilon$  is useful to avoid pathological cases when computing the angles  $\alpha_k^j$  or  $\beta_k^j$  (in our experiments we have chosen  $\varepsilon = 10^{-10}$ ). An example of limit curve obtained using this method is shown in figure 11.

## 4 Surface subdivision algorithm

In this section, we extend the method detailed in section 3 to discrete 2-surfaces in  $\mathbb{R}^3$ . In the following, we will assume that the surface is *manifold*, i.e. each edge belongs to at most two triangles (if this is an interior edge, and only one in the case of a boundary edge) and that the intersection of two triangles is either empty or exactly two vertices and one edge. This assumption is not restrictive since a non-manifold model can always be turned into a set of manifolds version by “splitting” the faulty edges/vertices. This also means

**Algorithm 1.** Midpoint computation for plane curves

- 1: **for all** edges  $P_k^j P_{k+1}^j$  **do**
- 2:   Compute the normal vector  $\mathbf{n}_k^j$  at  $P_k^j$  using relation (5)
- 3:   Compute the equation of the line  $\mathcal{L}$ , normal to  $\mathbf{n}_k^j$  passing at  $P_k^j$
- 4:   Compute the projection  $M_{k+1}^j$  of  $P_{k+1}^j$  on  $\mathcal{L}$  along  $\mathbf{n}_k^j$
- 5:   **if**  $\|P_k^j M_{k+1}^j\| < \varepsilon$  **then**
- 6:      $\hat{P}_{2k+1}^{j+1} = \frac{P_k^j + P_{k+1}^j}{2}$
- 7:   **else**
- 8:     Compute the angle  $\alpha_k^j = \pm \arctan\left(\frac{\|P_{k+1}^j M_{k+1}^j\|}{\|P_k^j M_{k+1}^j\|}\right)$
- 9:     Compute the polar coordinates  $\left(\frac{r_k^j}{2}, h(\alpha_k^j)\right)$  of  $\hat{P}_{2k+1}^{j+1}$  in  $\mathcal{R}_k^j$
- 10:    Compute the coordinates of  $\hat{P}_{2k+1}^{j+1}$  in  $\mathcal{R}$
- 11:   **end if**
- 12:   Repeat all the steps from line 2 to 11 using  $P_{k+1}^j$  as basis to compute  $\tilde{P}_{2k+1}^{j+1}$
- 13:    $P_{2k+1}^{j+1} = \frac{\hat{P}_{2k+1}^{j+1} + \tilde{P}_{2k+1}^{j+1}}{2}$
- 14: **end for**

that the faces surrounding a vertex are topologically equivalent to a disk (or a half-disk in case of a boundary vertex).

It is also necessary to have an *orientable* surface. In the case of a triangulated surface, this means that for each vertex  $v_i$ , the faces surrounding this vertex can be oriented, i.e. their normal vectors are all pointing to the “same side” of the surface. Global orientation of a surface can be easily achieved, for instance by using the spanning tree of the dual graph of the mesh. Dealing with manifold meshes implies that the spanning tree is a binary tree, since each face has at most one “parent” and two “children”. The traversal of the spanning tree permits the computation of a consistent normal vector for each face (i.e. each node of the spanning tree). Non-orientable surfaces, such as the Möbius strip or the Klein bottle, can be subdivided using the proposed algorithm with a few additional precautions.

As for the univariate case presented in section 3, it is possible to define a local coordinate system for a surface. The local coordinates at each vertex can be easily defined provided that the normal vector to the surface can be estimated at each vertex. Since we are dealing with an orientable surface, all the faces surrounding vertex  $v_i$  have a normal vector oriented in a consistent way. The normal vector  $\mathbf{n}_{v_i}$  at vertex  $v_i$  can be estimated by averaging the normal vectors of the faces having  $v_i$  as vertex, weighted by the area of each triangle. If we denote by  $\mathcal{F}_{v_i}$  the set of faces having  $v_i$  as vertex, by  $\mathbf{n}_f$  the unit normal to the face  $f$  and by  $|f|$  the area of face  $f$ , the expression of  $\mathbf{n}_{v_i}$

is

$$\mathbf{n}_{v_i} = \frac{\sum_{f \in \mathcal{F}_{v_i}} |f| \mathbf{n}_f}{\|\sum_{f \in \mathcal{F}_{v_i}} |f| \mathbf{n}_f\|},$$

which is quite similar to the expression (4) used in the univariate case. At each vertex  $v_i$ , it is now easy to define a local coordinates system  $\mathcal{R}_{v_i} = (\mathbf{t}_{v_i}, \mathbf{u}_{v_i}, \mathbf{n}_{v_i})$ , where  $(\mathbf{t}_{v_i}, \mathbf{u}_{v_i})$  is an orthonormal basis of the tangent plane at vertex  $v_i$ . Let  $v_k$  be a vertex sharing an edge with  $v_i$ . For each edge  $\{v_i, v_k\}$  we define  $\mathbf{v}_{i,k}$  as the vector from  $v_i$  to  $v_k$ , as well as the corresponding unit vector  $\mathbf{w}_{i,k} = \frac{1}{r_{i,k}} \mathbf{v}_{i,k}$ , where  $r_{i,k} = \|\mathbf{v}_{i,k}\|$ . Since  $\mathbf{w}_{i,k}$  is a unit vector, we can define two angles  $\theta_{i,k}$  and  $\phi_{i,k}$  which are the usual spherical coordinates

$$\mathbf{w}_{i,k}^{\mathcal{R}_{v_i}} = \begin{pmatrix} \cos \phi_{i,k} \sin \theta_{i,k} & \sin \phi_{i,k} \sin \theta_{i,k} & \cos \theta_{i,k} \end{pmatrix}^T.$$

Therefore, the spherical coordinates of  $\mathbf{v}_{i,k}$  in  $\mathcal{R}_{v_i}$  are  $(r_{i,k}, \theta_{i,k}, \phi_{i,k})$ . Figure 3 illustrates the various notations proposed.

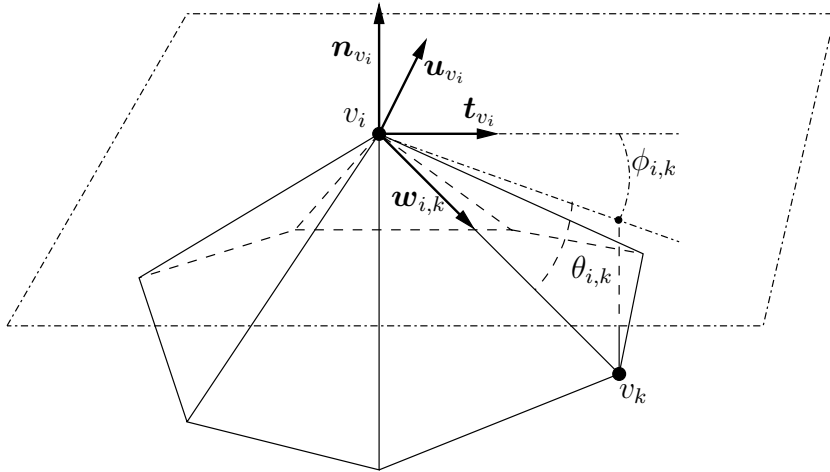


Fig. 3. Local coordinate system around vertex  $v_i$ . (NB: the “dash-dotted” lines all belong to the same plane)

Using these definitions, we can describe the subdivision method for triangulated surfaces. Let us denote by  $\mathcal{S}^j = (\mathcal{V}^j, \mathcal{T}^j)$  the discrete surface at level  $j$ . The midpoint computation is closely related to the method described in section 3. Let us consider a vertex  $v_i^j$  belonging to  $\mathcal{V}^j$ . For each vertex  $v_k^j \in \mathcal{E}^{v_i^j}$ , the spherical coordinates of the vector  $\mathbf{v}_{i,k}^j$  in  $\mathcal{R}_{v_i^j}$  are  $(r_{i,k}^j, \theta_{i,k}^j, \phi_{i,k}^j)$ . Let us define the vector  $\hat{\mathbf{v}}_{i,k}^{j+1}$  such that its spherical coordinates in  $\mathcal{R}_{v_i^j}$  are

$$\hat{\mathbf{v}}_{i,k}^{j+1} = \left( \frac{r_{i,k}^j}{2}, h(\theta_{i,k}^j), \phi_{i,k}^j \right). \quad (13)$$

According to the definition of  $\mathbf{v}_{i,k}^j$ , there exists a unique vertex  $\hat{v}_{i,k}^{j+1}$  such that  $\hat{\mathbf{v}}_{i,k}^{j+1}$  is the vector from  $v_i^j$  to  $\hat{v}_{i,k}^{j+1}$ . A similar computation can be done by

writing the spherical coordinates of the vector  $\mathbf{v}_{k,i}^j$  in  $\mathcal{R}_{v_k^j}$ , which leads to the definition of  $\tilde{\mathbf{v}}_{k,i}^{j+1}$

$$\tilde{\mathbf{v}}_{k,i}^{j+1} = \left( \frac{r_{i,k}^j}{2}, h(\theta_{k,i}^j), \phi_{k,i}^j \right). \quad (14)$$

This relation leads to the definition of a unique  $\tilde{v}_{k,i}^{j+1}$  such that  $\tilde{\mathbf{v}}_{k,i}^{j+1}$  is the vector from  $v_k^j$  to  $\tilde{v}_{k,i}^{j+1}$ . An interesting property of the proposed method, derived from equations (13) and (14) is that the scheme is locally invariant with respect to a rotation around the normal vector, since the  $\phi$  angles are not modified in the subdivision process.

Finally, the coordinates of the new vertex  $v_{i,k}^{j+1}$  in the canonical basis of  $\mathbb{R}^3$ , inserted between  $v_i^j$  and  $v_k^j$  is computed using the following relation

$$v_{i,k}^{j+1} = \frac{\hat{v}_{i,k}^{j+1} + \tilde{v}_{k,i}^{j+1}}{2}.$$

The actual computation of the midpoint can be achieved using an algorithm close to algorithm 1 (except you have to compute projection into tangent plane instead over the tangent line).

When boundary edges are encountered, the proposed method can still be applied although the definition of normal vectors has to be modified. Still the normal vector at a vertex belonging to a boundary edge is computed using relation (4). The difference lies in the definition of the normal vertex for a boundary edge. Let us consider a triangle formed by vertices  $(v_k, v_l, v_m)$ , where  $v_k v_l$  is a boundary edge. If we denote by  $\mathbf{n}$  the normal vector to the considered triangle, we define the normal  $\mathbf{n}_{k,l}$  to edge  $v_k v_l$  by

$$\mathbf{n}_{k,l} = \frac{\mathbf{n}_{k,l} \times \mathbf{n}}{\|\mathbf{n}_{k,l} \times \mathbf{n}\|},$$

which is a unit vector orthogonal to both  $\mathbf{n}$  and  $\mathbf{v}_{k,l}$ .

## 5 Results - Analysis

In this section we discuss the results obtained with the proposed method and analyze them from different points of view. As it is the case for many other modalities, the methodology used to assess the performance of subdivision surfaces depends on the underlying application. Here we provide an as generic as possible methodology, and for a broad class of applications. Elements of our assessment methodology will take into account complexity and memory bandwidth, the total memory requirements, but the emphasis will be put on efficiency in terms of distortions produced.

### 5.1 Objective comparison of subdivision schemes

The main class of applications for subdivision schemes is computer-aided modeling. The requirements in terms of memory and memory bandwidth are quite similar for the most common subdivision schemes. The non-linearity of the scheme adds complexity with respect to a linear scheme having the same support. Experiments showed that the overhead introduced remains acceptable. A typical result is that 0.4 seconds are needed to subdivide 82000 triangles, against 0.2 for Butterfly scheme, obtained on a Pentium III running at 666 MHz. Thus, the study will focus here on the convergence of the scheme, and on the regularity of the limit function or surface, which are often stated as desirable properties for subdivision schemes. In this particular case, it is not possible to use the methods [26] based on the analysis of the subdivision matrix used for stationary linear schemes such as the Butterfly scheme. Having no simple closed-form for computing the new point, we will focus on numerical experiments.

In this section, we present numerical methods to estimate the approximation order of the interpolation scheme, and to estimate the regularity of the limit function.

#### i) Uniform convergence criterion

Let  $\mathcal{G} = \{g_k\}_{k \in \mathbb{N}}$  be a sequence of real-valued functions. A sufficient condition for  $\mathcal{G}$  to converge uniformly toward a limit function is that there exist  $0 < \alpha < 1$  and  $\beta > 0$  such that for all  $k \in \mathbb{N}^*$

$$\|g_k - g_{k-1}\|_\infty < \beta \alpha^{k-1}. \quad (15)$$

Taking the logarithm of this condition leads to

$$\log \|g_k - g_{k-1}\|_\infty < \log \beta + (k - 1) \log \alpha. \quad (16)$$

Let us now consider a sequence of piecewise linear functions  $f_j$  obtained through subdivision. In that case, using the fact that we are dealing with interpolating subdivision, the quantity  $\|f_j - f_{j-1}\|_\infty$  can be expressed as

$$\max_k \left| f_{j,2k+1} - \left( f_{j-1,k} + \frac{f_{j-1,k+1} - f_{j-1,k}}{x_{j-1,k+1} - x_{j-1,k}} (x_{j,2k+1} - x_{j-1,k}) \right) \right|.$$

We denote by  $D_f(j)$  the quantity  $\log \|f_j - f_{j-1}\|_\infty$ . An upper bound for  $\alpha$  and  $\beta$  can be inferred from the plot of  $D_f$  versus the level of subdivision  $j$ , which should be close to linear.

#### ii) Approximation order estimate

**Definition 1** Let  $g$  be a sufficiently smooth real-valued function. Let  $X_0$  be the coarsest grid on which subdivision will be performed. The data  $\{f_{0,k}\}$  living on  $X_0$  is built using  $f_{0,k} = g(x_{0,k}) \forall k$ . Let us define also  $\eta$  as follows

$$\eta = \max_k |x_{0,k+1} - x_{0,k}|.$$

Let  $\tilde{f}$  be the limit function obtained when  $k \rightarrow \infty$ . The approximation order is defined by the biggest integer  $p$  for which the following formula holds

$$\|\tilde{f} - g\|_\infty \leq C\eta^p, \text{ with } 0 \leq C < +\infty.$$

Assuming that  $\|\tilde{f}_j - g\|_\infty$  provides a good estimate for  $\|\tilde{f} - g\|_\infty$ , and that the bound is attained, let us define  $\Delta_j(\eta) = \|f_j - g\|_\infty$  for a given  $\eta$ . Under the previous assumption, the following approximation holds

$$\Delta_j(\eta) \approx C\eta^p.$$

Therefore, we have

$$\log \Delta_j(\eta) \approx \log C + p \log \eta,$$

which implies that the plot of  $\log \Delta_j(\eta)$  against  $\log \eta$  should be a linear function having a slope  $p$ , providing an estimator of the approximation order.

iii) Regularity estimate

**Definition 2** A function  $f : I \subset \mathbb{R} \rightarrow \mathbb{R}$  has  $n + \alpha$  ( $n \in \mathbb{N}$ ,  $0 \leq \alpha \leq 1$ ) regularity (or Hölder-regularity) if

$$\exists C < +\infty \text{ s.t. } \left| \frac{\partial^n f}{\partial x^n}(x_1) - \frac{\partial^n f}{\partial x^n}(x_2) \right| \leq C|x_1 - x_2|^\alpha, \forall (x_1, x_2) \in I^2.$$

This definition can be adapted for subdivision schemes as proposed in [22] (only for regular grids). We adapt the method to non-uniform grids. Let us define  $\rho_j^{[l]} = l! \max_k |f_{j,k+1}^{[l]} - f_{j,k}^{[l]}|$ , and  $\beta_j = \max_k |x_{j,k+1} - x_{j,k}|$ .

**Definition 3** A subdivision scheme has  $l + \alpha_l$  Hölder regularity if

$$\exists C < +\infty \text{ such that } \rho_j^{[l]} \leq C(\beta_j)^{\alpha_l}.$$

Assuming that the maximal value is attained,  $\alpha_l$  can be approximated by

$$\alpha_l = \lim_{j \rightarrow +\infty} \frac{\log \frac{\rho_{j+1}^{[l]}}{\rho_j^{[l]}}}{\log \frac{\beta_{j+1}}{\beta_j}}. \quad (17)$$

## 5.2 Univariate subdivision results

The proposed univariate scheme is compared with the four-point scheme from equation (3), using  $w = \frac{1}{16}$ , in several ways. Starting from a coarse regular grid, sampled from an analytic smooth function, subdivision is performed several times and the evolution of several criteria are studied.

- The evolution of the  $L_2$  error between to the reference function and the piecewise linear approximations obtained through subdivision. The  $L_\infty$  norm usually reaches its maximum value when the level is equal to 1. Given the fact that we use interpolating subdivision, the value remains constant for all levels  $> 1$ . This experiment will be performed using uniform and non-uniform initial grids. In the non-uniform case, the Lagrange four-point scheme is replaced by the *adaptive Lagrange* scheme studied in [6]. The  $L_2$  error is computed using a numerical approximation of the integral of the squared difference of the two functions.
- The estimates of  $\alpha$  and  $\beta$  in order to check the rate of convergence of the subdivision schemes, as defined in equation (16). Uniform and non-uniform coarsest grids will be used to perform this test. This experiment is performed with both uniform and non-uniform initial grids.
- The estimate of the subdivision scheme's approximation order proposed in section 5.1.
- an estimate of the regularity of the limit function, i.e. the  $\alpha_l$  defined in equation (17). This estimate will be computed with both uniform and non-uniform coarsest grids.

In the proposed tests, two analytic functions  $f$  and  $p$  have been used. We have chosen a Gaussian function  $f(x) = e^{-x^2}$  and the polynomial  $p(x) = x^2 + \frac{x^4}{1000}$ . Both functions present a sufficient regularity,  $f$  being  $\mathcal{C}^\infty$  and  $p$  being  $\mathcal{C}^3$ .

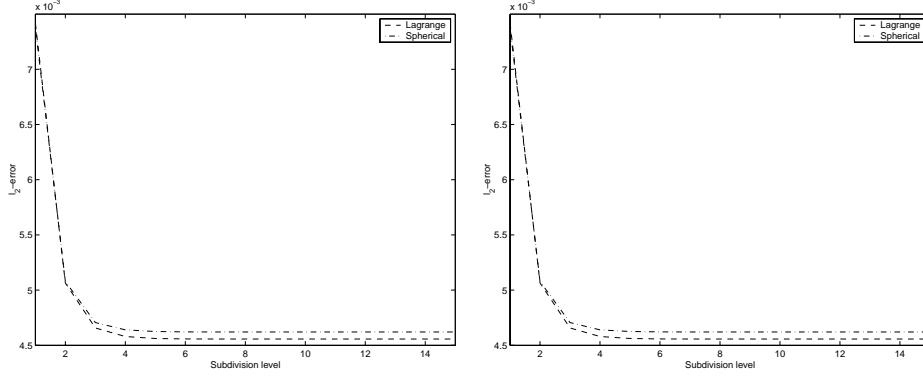
### i) $L_2$ error evolution – convergence

The functions  $p$  and  $f$  have been sampled regularly over the interval  $[-2; 2]$ . The coarsest grid contains 10 points. The  $L_2$  error between the piecewise linear function obtained through subdivision and the reference function is shown in figures 4(a) and 4(b). The convergence parameters obtained with these uniform initial datasets are shown in table 1.

Similar experiments have been performed, using a non-uniform initial grid, given in table 2. Figure 5 show the evolution of the  $L_2$  norm between the function obtained through subdivision and the reference function. Table 3 shows the convergence parameters obtained with the non-uniform initial grid.

### ii) Approximation order

In this experiment, we considered the same functions  $f$  and  $p$  over the interval  $[-5; 5]$ . The parameter  $\eta$  varies approximately between 1 and



(a)

(b)

Fig. 4. Evolution of the  $L_2$  error between reference function ( $f$  for figure 4(a) and  $p$  for figure 4(b)) and approximations obtained through subdivision (uniform initial grid)

Table 1

Convergence parameters (uniform initial grid)

		$\beta$	$\alpha$
$f$	Lagrange	0.042294	0.275498
	Spherical	0.041820	0.276377
$p$	Lagrange	0.064700	0.287693
	Spherical	0.060222	0.282725

Table 2

Non-uniform initial grid

$k$	0	1	2	3	4	5	6	7	8	9
$x_{0,k}$	-2	-1.5	-1.4	-1	0	0.2	0.6	0.9	1.3	2

Table 3

Convergence parameters (non-uniform initial grid)

		$\beta$	$\alpha$
$f$	adaptive Lagrange	0.182563	0.275909
	Spherical	0.035442	0.291331
$p$	adaptive Lagrange	0.231814	0.279569
	Spherical	0.119682	0.282817

$10^{-3}$ . As shown by figure 6 the plot of  $\log \Delta_j$  against  $\log \eta$  is linear with a slope  $p$  for high values of  $\eta$ , whereas a saturation occurs for small  $\eta$ . As a consequence, the approximation order is computed using only the non-saturated part of the curve. Table 4 shows the values of the approximation



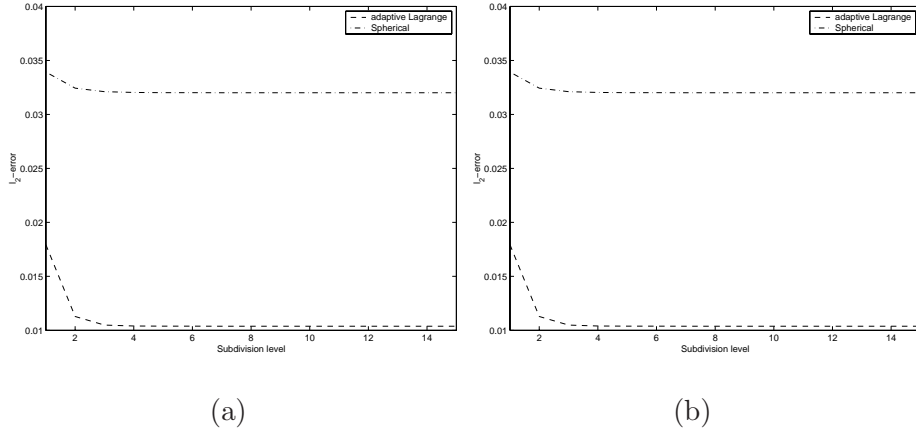


Fig. 5. Evolution of the  $L_2$  error between a reference function and approximations obtained through subdivision, with a non-uniform initial grid. (a) uses  $f$  as reference and (b) is obtained using  $p$ .

order obtained.

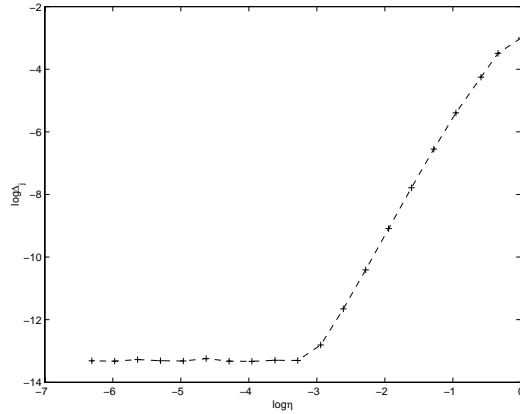


Fig. 6. Plot of  $\log \Delta_j$  vs.  $\log \eta$  for Lagrange scheme. Using the spherical scheme instead gives very similar results (linear part followed by a saturation when  $\eta$  is small)

Table 4

Numerical values of approximation order

		Approx. order
$f$	Lagrange	3.4124
	Spherical	3.2664
$p$	Lagrange	1.9350
	Spherical	1.9349

iii) Hölder regularity of the limit functions

The values of  $\alpha_0$  and  $\alpha_1$  have been computed using both uniform and

non-uniform initial grids used for the estimation of the convergence parameters. Not surprisingly, values of  $\alpha_0$  remain very close to 1 for all levels. The evolution of  $\alpha_1$  is shown in figures 7 (uniform starting grid) and 8 (non-uniform).

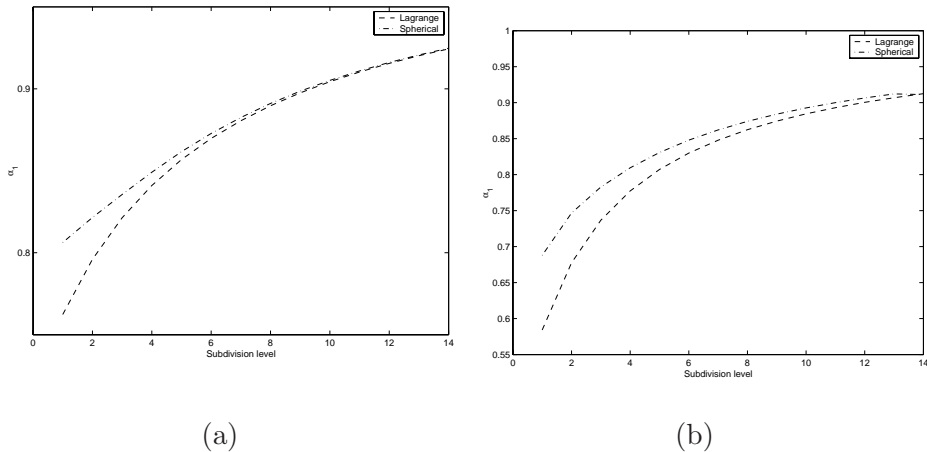


Fig. 7. Evolution of  $\alpha_1$  for a uniform initial grid sampled from  $f$  (a) and  $p$  (b)

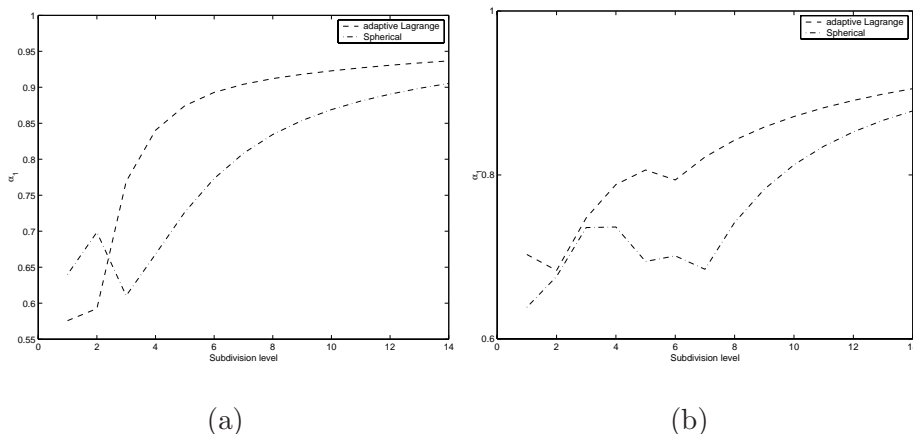


Fig. 8. Evolution of  $\alpha_1$  for a non-uniform initial grid sampled from  $f$  (a) and  $p$  (b)

### 5.3 Surface subdivision results

The proposed surface subdivision scheme is compared with two well known schemes : the *Butterfly* scheme (extended as proposed in [27]) and the *Loop* scheme. While both the Butterfly and the proposed scheme are interpolating schemes, the Loop scheme is only approximating, i.e. it modifies the position of the vertices between levels  $j$  and  $j + 1$ .

In order to be able to adapt some of the proposed criteria to triangular meshes, we need to have a distance measurement between two triangular meshes. The approach described in [1], based on the Hausdorff distance provides both peak and RMS values for distortion, providing a reliable approximation of  $\ell_2$  and  $\ell_\infty$  norms for this type of data. A reduced set of experiments is performed using the surface subdivision schemes

- as for univariate subdivision, the  $\ell_2$  and  $\ell_\infty$  error (computed using the symmetrical Hausdorff distance) between the subdivided surface and a reference surface (having an analytic expression) is computed.
- the parameters  $\alpha$  and  $\beta$  defined in equation (16) are also estimated in order to check the convergence of the proposed scheme. In addition to the previous experiment, a “real world” surface is also used to study the convergence parameters in a concrete case.

In our experiments, the reference surfaces are a torus and a sphere, which have both analytic expressions. The sphere at the coarsest level is approximated by an icosahedron.

The initial icosahedron has been subdivided 7 times, and the initial torus 6 times. The reason for not performing a greater number of subdivision steps is the exponential growth of the number of triangles, which leads to huge processing times when computing the error. Figure 9 shows the errors obtained using the torus as base-model, and figure 10 gives the results obtained using the icosahedron as a base model. The initial torus had 64 triangles.

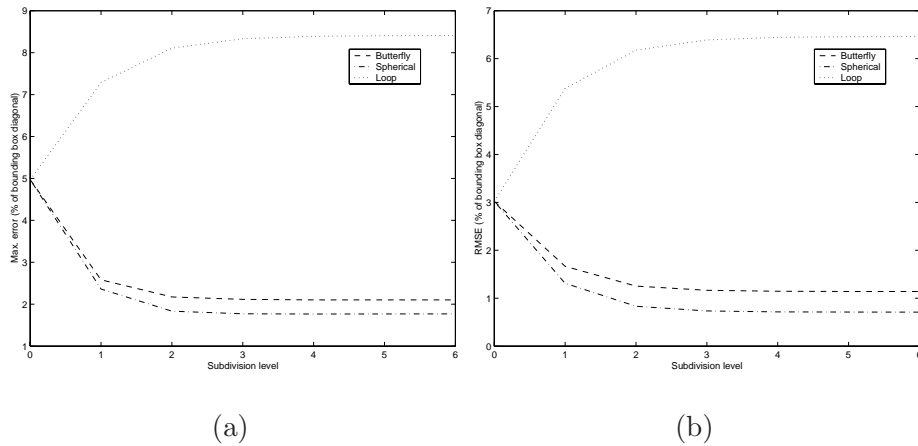


Fig. 9. Relative error (in percentage of the length of the bounding-box diagonal) between the reference torus and those obtained through subdivision ( $\ell_\infty$ -error in (a),  $\ell_2$ -error in (b))

As in the univariate case, the convergence parameters have been estimated for the tested subdivision methods. The results are shown in table 5. A visual com-

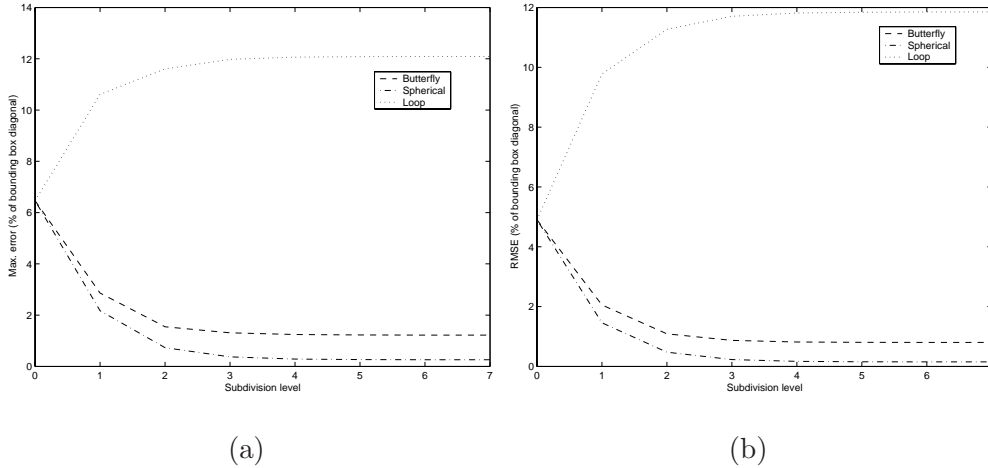
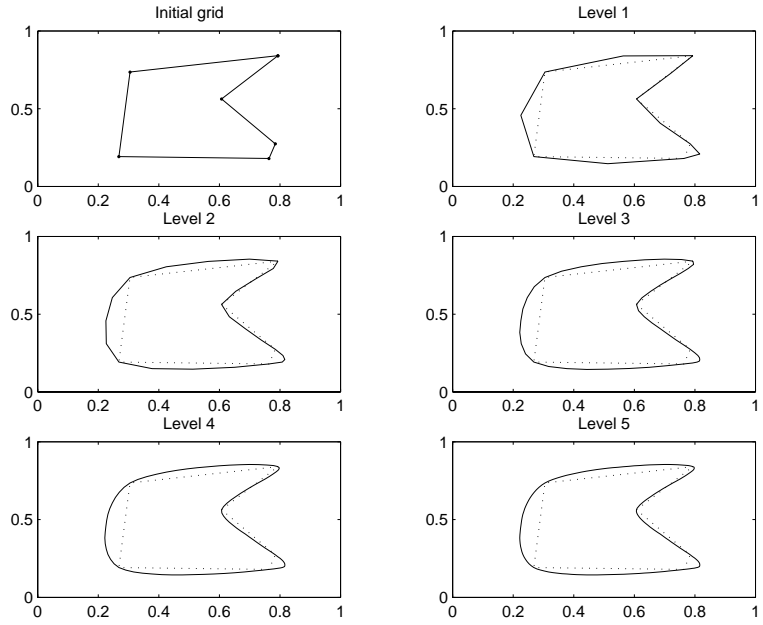


Fig. 10. Relative error (in percentage of the length of the bounding-box diagonal) between the reference sphere and those obtained through subdivision ( $\ell_\infty$ -error in (a),  $\ell_2$ -error in (b))

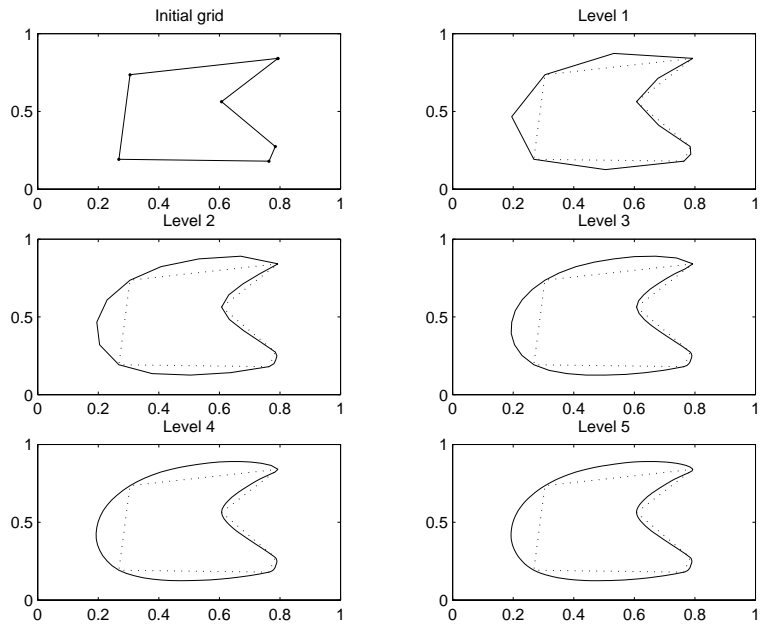
Table 5  
Surface subdivision convergence parameters

		$\beta$	$\alpha$
Torus	Butterfly	3.7487	0.3367
	Spherical	4.003	0.3251
	Loop	6.1148	0.2509
Sphere	Butterfly	4.0939	0.3364
	Spherical	7.4925	0.2522
	Loop	5.0838	0.3167
Rabbit	Butterfly	4.1737	0.4016
	Spherical	3.9139	0.4640
	Loop	7.2925	0.2721

parison of the limit curves obtained with the univariate scheme is presented in figure 11. The convergence parameters have also been computed using a “real-world” surface. We have chosen to use a coarsened version of the “rabbit” model (courtesy of CyberWare). The original model has approximately 134 000 triangles, and the coarsened version has 50 triangles. The original model has been simplified using QSlim[18]. The coarse model has been subdivided 6 times. Table 5 shows the results obtained using this model. Figures 12 and 13 show the differences between limit surfaces, starting respectively from an icosahedron and from the coarse “rabbit” model as initial meshes.



(a) Four-point scheme,  $w = \frac{1}{16}$ , 5 levels of subdivision



(b) Proposed method, 5 levels of subdivision

Fig. 11. Plane curve subdivision - visual comparison

#### 5.4 Analysis

The results developed in section 5 reveal many interesting properties of the proposed scheme.

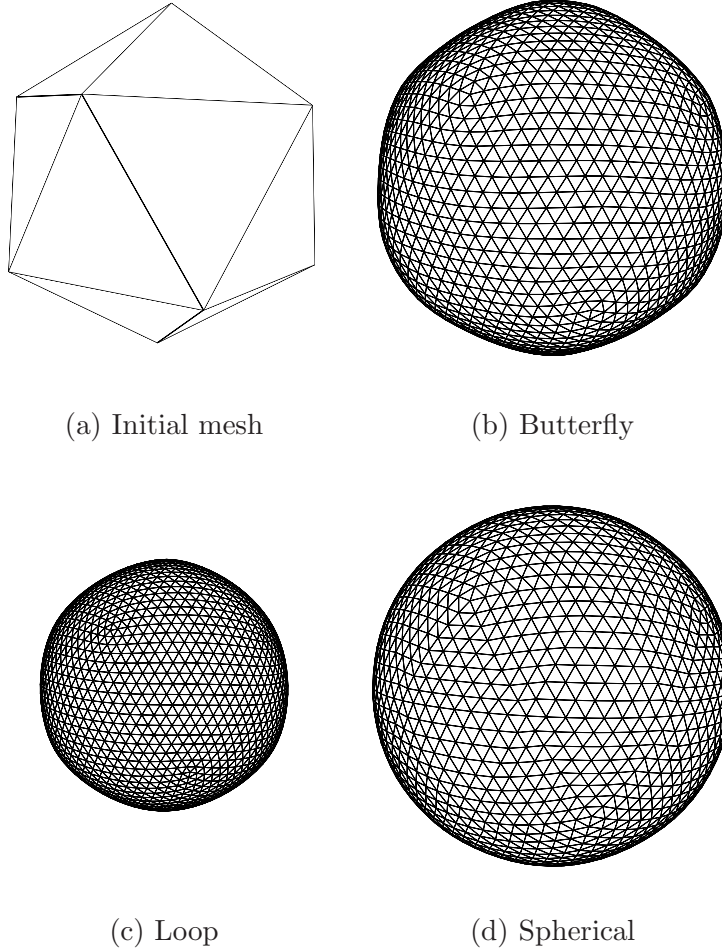
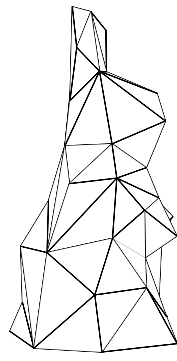


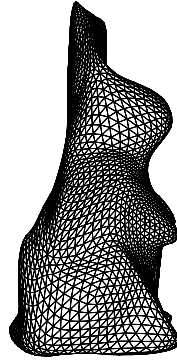
Fig. 12. Icosahedron subdivision (4 levels)

The figures presented in table 1 indicate that the Lagrange and Spherical scheme behave closely in terms of convergence. From equation (15), it is clear that  $\alpha$  is the parameter that determines whether the scheme converges or not. This equation also implies that the closer  $\alpha$  is to 0, the faster the convergence will be. Figures 4(a) and 4(b) provide an additional information concerning the convergence, showing that the  $\ell_2$ -error between the reference function and the approximations obtained through subdivision decreases when the subdivision level increases. These two results show that the proposed scheme converges toward a limit function, and that this limit is close to the reference function from which the initial grid is sampled.

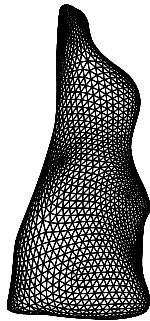
The approximation order gives an additional indication concerning the variation of the residual error with respect to the sampling step of the coarsest grid. The results presented in table 4 show that in the case of a uniform initial grid, the Lagrange and the proposed scheme behave very closely. It is important to keep in mind that the numerical method we have proposed only gives a lower bound of the real approximation order. For instance, the ap-



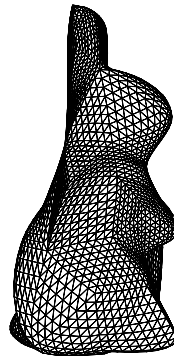
(a) Initial mesh



(b) Butterfly



(c) Loop



(d) Spherical

Fig. 13. Coarse “rabbit” model subdivision (3 levels)

proximation order of the 4-point scheme is 4 for  $w = \frac{1}{16}$ , since the scheme reproduces cubic-polynomials. The estimators computed in our experiments remained lower than this theoretical bound, most likely because of the  $\ell_\infty$  norm in definition 1, and because of the choice of the functions from which the initial grids are built.

When dealing with non-uniform grids, the differences between the proposed scheme and the adaptive Lagrange are greater than those observed in the uniform case. The adaptive Lagrange scheme slightly outperforms the Spherical scheme, as shown by figure 5 and table 3. The approximation order in the non-uniform case has not been computed since it has no real meaning to vary the “step” of a non-uniform grid, although an experiment close to the one performed on uniform grids could be imagined.

The estimate of the Hölder regularity of the limit function, plotted in figures 7 (uniform case) and 8 (non-uniform case), clearly show that the Lagrange and Spherical schemes behave closely. The regularity of the limit function is

close to 2, which is consistent with the fact that the Lagrange scheme leads to  $\mathcal{C}^{2-\epsilon}$  functions, both in the uniform and non-uniform cases (see [6] for more details). In the uniform case, the proposed scheme slightly outperforms the Lagrange scheme. Although in this case the limit value of the Hölder exponent is very close, figure 7 shows that the proposed scheme provides more regular functions at low subdivision levels. In the non-uniform case, the adaptive Lagrange clearly outperforms the proposed scheme in terms of regularity. The irregularities in the value of  $\alpha_1$  are most likely due to the fact that the regularity estimate is not sharp.

The experiments performed using the univariate schemes show that the proposed scheme performs almost identically with the four-point scheme on uniform grids, which is confirmed by the visual comparison of figure 11. The adaptive Lagrange seems to be slightly better when dealing with non-uniform grids. However, while the Spherical scheme has been successfully adapted to surface subdivision with a relatively small overhead, such a transposition would not be possible directly for the adaptive Lagrange scheme. Another feature of the proposed surface subdivision scheme is that it does not involve a re-computation of a new mask when dealing with “irregular” vertices. While this property is desirable on highly irregular meshes, it loses some interest when the level of subdivision increases, since the new vertices introduced at each level are all of valence 6, therefore leading to *semi-regular* meshes, i.e. where only isolated vertices are not of valence 6.

The evolution of the error between the reference model and the approximations obtained through subdivision showed in figures 9 and 10 is very similar to the univariate case. The distance between the reference and the approximation decreases as the subdivision level increases, except for the Loop scheme where it increases. This is due to the fact that both the Butterfly and Spherical schemes are *interpolating*, while the Loop scheme is *approximating*. The limit surfaces generated by the Loop scheme are smoother than the one obtained using the Butterfly or the Spherical schemes, but this is obtained at the cost of an error between the reference and the approximations larger than the error achieved using interpolating schemes. It is also interesting to note that the Spherical scheme’s limit surface is closer to the reference surface than the Butterfly’s.

The convergence parameters in table 5 show that all three schemes are close to each other. The Loop and Spherical have a smaller  $\alpha$  than the Butterfly scheme; the Loop scheme seems to converge faster on the torus while the Spherical converges faster on the sphere, most likely because of the use of spherical coordinates in the computation of the midpoint. The test performed on the “rabbit” model shows that the spherical subdivision seems to converge slower than other methods, probably due to the high-curvature points that are present in the model. The visual comparison presented in figures 12 and 13



show that the Butterfly and Spherical scheme behave closely. The proposed method leads to better results in terms of smoothness on the icosahedron whereas the results obtained using the “rabbit” model do not give a clear advantage to either interpolating methods. Not surprisignly, Loop subdivision produces smoother limit surfaces. The “patches” visible on the surface in figure 13(d) are not linked to discontinuities in the limit surface. This artifacts in the triangulation may be caused by the independent treatment (except when computing the normal vector) of each edge.

## 6 Conclusion

In this paper, we have proposed a non-linear subdivision scheme that is suitable for univariate data and discrete triangulated surfaces, possibly with boundaries. The scheme is more complex than a linear one, but complexity remains acceptable for most applications. Several numerical criteria have been proposed to study the most important features of the proposed scheme. Those estimators tend to prove that the univariate scheme behaves in a very similar way to the Lagrange four-point scheme. In addition, the univariate scheme performs decently on non-uniform grids, and can be easily adapted to subdivide discrete surfaces. The performance obtained on triangulated surfaces show results similar to the Butterfly scheme in terms of convergence and approximation error, but at the same time are closer to the Loop scheme in terms of regularity of the limit surface, although for this criterion, only a qualitative analysis has been performed. Several questions remain open concerning the proposed schemes, namely the analytic proof for convergence, approximation order and regularity of the limit function or surface.

## References

- [1] N. Aspert, D. Santa-Cruz, and T. Ebrahimi. MESH: Measuring Errors between Surfaces using the Hausdorff distance. In *Proceedings of the IEEE International Conference on Multimedia and Expo*, volume I, pages 705 – 708, 2002. <http://mesh.epfl.ch>.
- [2] A.S. Cavaretta, W. Dahmen, and C.A. Micchelli. *Stationary subdivision*, volume 93 of *Memoirs of the AMS*. American Mathematical Society, 1991.
- [3] G. M. Chaikin. An algorithm for high-speed curve generation. *Computer Graphics and Image Processing*, 3:346–349, 1974.
- [4] A. Cohen and N. Dyn. Nonstationary subdivision schemes and multiresolution analysis. *SIAM Journal of Mathematical Analysis*, 27(6):1745–1769, November 1996.

- [5] A. Cohen, N. Dyn, and B. Matei. Quasilinear subdivision schemes with applications to ENO interpolation. Preprint - submitted to Applied and Computational Harmonic Analysis, 2001.
- [6] I. Daubechies, I. Guskov, and W. Sweldens. Regularity of irregular subdivision. *Constructive Approximation*, 15:381–426, 1999.
- [7] I. Daubechies and J.C. Lagarias. Two-scale difference equations. II. Local regularity, infinite product of matrices, and fractals. *SIAM Journal of Mathematical Analysis*, 23:1031–1079, 1992.
- [8] Georges de Rham. Un peu de mathématiques à propos d’une courbe plane. *Revue de Mathématiques Élémentaires*, II, 1947. Oeuvres complètes, pp. 678–689.
- [9] Georges de Rham. Sur les courbes limites de polygones obtenus par trisection. *Enseignement Mathématique*, V(1), 1959. Oeuvres complètes, pp. 729–743.
- [10] G. Deslauriers and S. Dubuc. Symmetric iterative interpolation. *Constructive Approximation*, 5:49–68, 1989.
- [11] N. A. Dodgson, M. A. Sabin, L. Barthe, and M. F. Hassan. Towards a ternary interpolating subdivision scheme for the triangular mesh. Technical Report 539, University of Cambridge Computer Laboratory, July 2002.
- [12] N. Dyn. Subdivision schemes in computer-aided geometric design. In W. Light, editor, *Advances in Numerical Analysis*, volume II, pages 36–104. Clarendon Press, 1992.
- [13] N. Dyn, J. A. Gregory, and D. Levin. A 4-point interpolatory subdivision scheme for curve design. *Computer-Aided Geometric Design*, 4:257–268, 1987.
- [14] N. Dyn, J.A Gregory, and D. Levin. A Butterfly subdivision scheme for surface interpolation with tension control. *ACM Transactions on Graphics*, 9(2):160–169, 1990.
- [15] N. Dyn and D. Levin. The subdivision experience. In P.J. Laurent, A. Le Méhauté, and L.L. Schumaker, editors, *Wavelets, Images and Surface Fitting*, pages 229–244. A. K. Peters, Wellesley, MA, 1994.
- [16] N. Dyn, D. Levin, and D. Liu. Interpolatory convexity-preserving subdivision schemes for curves and surfaces. *Computer-Aided Design*, 24(4):211–216, april 1992.
- [17] M. S. Floater and C. A. Michelli. Nonlinear stationary subdivision. In N. K. Govil, R. N. Mohapatra, Z. Nashed, A. Sharma, and J. Szabados, editors, *Approximation theory : in Memory of A. K. Varma*, pages 209–224. Marcel Dekker, New-York, 1998.
- [18] M. Garland and P. S. Heckbert. Surface simplification using quadric error metrics. In *SIGGRAPH 97 Proceedings*, pages 209–216, August 1997. <http://www.cs.cmu.edu/~garland/quadrics/>.

- [19] M. Hassan, I. Ivrişsimtziş, N. A. Dodgson, and M. Sabin. An interpolating 4-point  $C^2$  ternary stationary subdivision scheme. *Computer Aided Geometric Design*, 19(1):1–18, February 2002.
- [20] S. Karbacher, S. Seeger, and G. Häusler. A non-linear subdivision scheme for triangle meshes. In G. Greiner, H. Niemann, H.-P. Steinel, and B. Girod, editors, *Vision Modeling and Visualization*, pages 163–170, 2000.
- [21] L. Kobbelt.  $\sqrt{3}$ -Subdivision. In *Proceedings of SIGGRAPH*, pages 103–112. ACM SIGGRAPH, 2000.
- [22] F. Kuijt. *Convexity Preserving Interpolation – Stationary Nonlinear Subdivision and Splines*. PhD thesis, University of Twente, Faculty of Mathematical Sciences, 1998.
- [23] F. Kuijt and R. van Damme. Smooth interpolation by a convexity preserving non-linear subdivision algorithm. In A. Le Méhauté, C. Rabut, and L.L. Schumaker, editors, *Surface Fitting and Multiresolution Methods*, pages 219–224. Vanderbilt University Press, Nashville, TN, 1997.
- [24] F. Kuijt and R. van Damme. Shape preserving interpolatory subdivision schemes for nonuniform data. *Journal of Approximation Theory*, 114(1):1–32, January 2002.
- [25] C. Loop. Smooth spline surfaces based on triangles. Master’s thesis, University of Utah, Department of Mathematics, 1987.
- [26] D. Zorin. A method for analysis of  $C^1$ -continuity of subdivision surfaces. *SIAM Journal of Numerical Analysis*, 37(5):1677–1708, 2000.
- [27] D. Zorin, P. Schröder, and W. Sweldens. Interpolating subdivision for meshes with arbitrary topology. In *Proceedings of SIGGRAPH*, pages 189–192. ACM SIGGRAPH, 1996.