Subdivision Surfaces – Can they be Useful for Geometric Modeling Applications?

Dan Gonsor* and Marian Neamtu†

December 10, 2001

Abstract

This report summarizes the findings and recommendations of the authors concerning the usefulness of subdivision surfaces for geometric modeling, and in particular for engineering applications. The work described is a result of a three-month collaboration of the authors during the visit of the second author to Boeing in the Summer of 2001.

Executive Summary

The objective of this project was to investigate the usefulness of subdivision surfaces for geometric modeling, and in particular for engineering applications. Subdivision surfaces are a new type of surface representation that has several advantages over traditional surface representations, most notably the ability to model surfaces of an arbitrary topological structure. As a result of researching the subject, and implementing, extending and experimenting with a particular subdivision algorithm, we arrived at the following two conclusions. First, subdivision surfaces, in their current state of development, are not ripe for a major utilization in Boeing geometric modeling systems. We base this conclusion on the following observations

- There are currently no subdivision algorithms that generate sufficiently smooth surfaces.
- There are too many unanswered questions related to critical requirements for engineering applications. See Section 7 for more details.
- A thorough comparison has not been made between subdivision surfaces and traditional surface representations.

*The Boeing Company, daniel.e.gonsor@boeing.com.

†Center for Constructive Approximation, Department of Mathematics, Vanderbilt University, Nashville, TN 37240, neamtu@math.vanderbilt.edu.

Our second conclusion is that subdivision surfaces have enough potential to warrant further investigation. To that effect, we make the following recommendations:

- Investigate ways to overcome the more severe shortcomings related to engineering applications.
- Conduct a thorough comparison between subdivision surfaces and traditional surface representations.
- Incorporate a limited subdivision capability into a Boeing geometric modeling system. The purpose of this would be to allow users to experiment with subdivision surfaces in order to provide feedback for further evaluation.

TABLE OF CONTENTS

1 Introduction and Goals of the Project

During the last two decades, *subdivision surfaces* have emerged as an interesting alternative to the classical and better known *parametric surfaces* (such as non-uniform rational B-spline (NURBS) surfaces) or *implicit surfaces*. The large number of publications on subdivision surfaces (see the attached list of references) attests to their ever increasing popularity, especially amongst the computer-graphics, visualization, and animation communities. In view of these facts, a natural question arises as to whether the computer-aided geometric design (CAGD) community and, in particular, existing Boeing geometric modeling systems, might benefit from these recent developments in the field of surface construction and representation.

This report summarizes research completed over the period June-August 2001, during which one of us was a visiting professor at the Boeing Company. The overall objective of the project was to investigate subdivision methods and identify how/if they may be useful in geometric modeling.

Our plan was to focus on subdivision surfaces with respect to mainly engineering applications (as opposed to the more established applications of subdivision such as visualization). We set for ourselves two basic tasks: to investigate the mathematical issues regarding subdivision methods in an engineering setting (*e.g.*, what exactly can one say regarding behavior near the so-called extraordinary points), and secondly to experiment with the subdivision technology (*i.e.*, develop software, experiment with applications, run test cases). We were also interested in getting feedback from engineers throughout the company, and so decided to investigate implementing a subdivision capability into the Aero Grid and Panel System (AGPS). AGPS is a Boeing developed surface geometry system with approximately 300 users throughout the company (see [53] for an introduction to AGPS, and [6] for a description of some advanced applications).

To better understand the requirements of a possible addition to AGPS, we met with several people from the Mathematics and Computing Technology organization (Dave Ferguson, Fred Holt) and the AGPS group (Roger Pomeroy, Bill Capron, Randy Maier). The purpose of the meeting was to discuss the process of implementing a subdivision capability into AGPS. The following two requirements were identified as important:

- The subdivision scheme must be interpolatory *i.e.*, interpolate the initial set of data.
- C^1 (tangent plane continuity) is sufficient for prototyping, but C^2 (curvature continuity) is necessary for serious design (Dave Ferguson noted that Boeing has been working around not having C^2 for 60 years so it can be handled, but modeling would be greatly simplified by having it. Also C^2 at all but a finite number of points (which is the situation for the most common subdivision algorithms) would be a big improvement over current methods).

The minimum capabilities of an initial implementation were agreed to be:

- create a subdivision surface (from scattered data or data with some type of structure *e.g.*, a set of points along a set of airfoil sections where each section can have a different number of points)
- display the surface
- intersect the surface with a plane

In order to identify areas that may benefit from subdivision surfaces, we developed the following criteria:

- current solution methods are inadequate
- current solution methods are satisfactory but improved methods would be welcome
- the application is new and there are no current solution methods
- the application seems ideally suited for subdivision methods

Based on the above criteria, and discussions with the above mentioned persons, we identified the following applications of subdivision methods as appropriate:

- grafted surfaces *i.e.*, surfaces formed by connecting pieces of individual surfaces
- scattered data fitting
- surface intersection
- grid generation
- generation of a nested sequence of grids e.g., for multigrid applications
- multi-level representation of surfaces

Typical aircraft surfaces that are troublesome for existing methods, and hence useful as test cases for subdivision methods are:

- wing-body fairing (according to AGPS experts, C^1 would be enough here, at least initially)
- flat cab windshields in nicely contoured noses
- hsct (hyper sonic commercial transport)
- droop portions of leading/trailing edge of wing
- aft end

From Roger Pomeroy, we obtained data sets of a wing-body fairing, and a group of surfaces representing a body, with cab and horizontal tail wiping surface. The first of these data sets was used as a basic test case for our subdivision software (see Section 5).

It should be emphasized that we were not working under a preconceived notion of the superiority of subdivision surfaces to other surface representation schemes. Instead, we hoped to gain an understanding as to where this technology might be useful, if anywhere.

The report is organized as follows. In the following two sections we discuss unstructured meshes - in general and in terms of surface construction. A main reason for the popularity of subdivision surfaces is the ability to generate a surface from an unstructured mesh (we will make this more precise forthwith). In Section 4 we give a description of the subdivision surface paradigm. Section 5 describes the software we developed, along with the results of our experiments. Our analysis of the strengths and weaknesses of the subdivision method is presented in Section 6. Readers interested only in our final conclusions can skip to Section 7 which is followed by a section on future work.

2 Unstructured Meshes

Surfaces in Boeing applications are typically obtained by fitting a set of points that are connected to form a *mesh*. Figures 1 and 2 show examples of meshes. To form a *surface* from a mesh, the points of the mesh are typically interpolated or approximated (*e.g.*, least-squares approximated) by a collection of "simple" surfaces, called *patches*, such that their union gives rise to a single composite surface corresponding to the mesh.

The easiest way to describe a mesh is as a collection of *faces* that are connected along *edges* such that the totality of all the faces forms a 3D continuous surface. A face is defined as a surface primitive determined by its edges, which themselves are line segments connecting 3D points, called face *vertices*. A face is thus a surface (usually piecewise linear or bilinear) that interpolates the vertices and the edges and fills the hole enclosed by the edges. For example, a *quadrilateral* face has four vertices and four edges and can be thought of as the bilinear surface interpolating these four vertices. A *triangular* face is simply the triangle interpolating three 3D points (vertices), see Figure 4. In general, mesh faces can be arbitrary polygons.

The most common types of meshes are the so-called *structured* (or *regular*) meshes, defined by the property that all faces have the same number of vertices (or, equivalently, number of edges) and all non-boundary vertices have the same *valence i.e.*, number of faces (or edges) sharing the vertex. A *boundary vertex* is a vertex that is *not* completely surrounded by the faces attached to (*i.e.*, sharing) it. For example, most meshes used in AGPS are *regular quadrilateral* meshes, *i.e.*, each (non-boundary) vertex has valence four and each face is a quadrilateral. Another example of a structured mesh is a *regular triangular* mesh, consisting of triangular faces and vertices of valence six.

Regular quadrilateral meshes are the most popular meshes in applications since their structure allows one to employ *tensor-product* splines and NURBS surfaces when it comes to constructing *smooth* surfaces out of the given mesh. In particular, AGPS uses

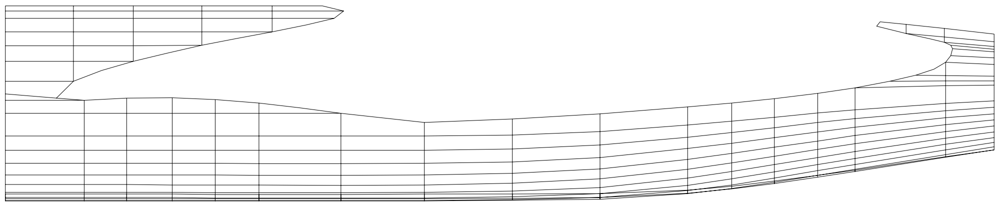

Figure 1: Mesh corresponding to the body of an aircraft.

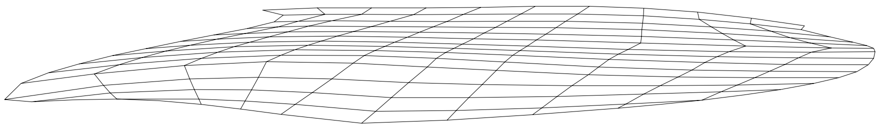

Figure 2: Mesh corresponding to the wing-body fairing of an aircraft.

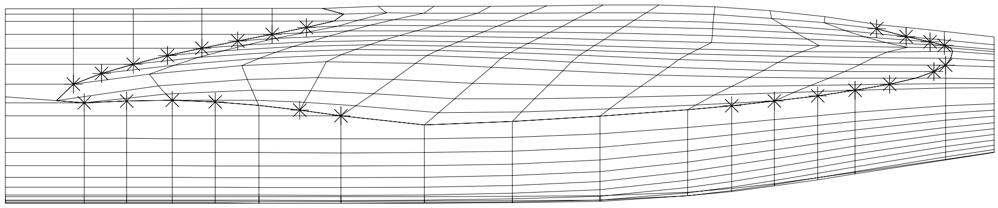

Figure 3: Joined meshes beget an irregular mesh.

quadratic, *cubic* and *quintic* tensor-product splines. Regular triangular meshes, while not being as frequently used in Boeing applications, also lend themselves to various well-understood and robust surface construction techniques, such as the so-called three-direction box splines or triangular NURBS patches.

While regular meshes have their well-deserved place in many industrial software packages (including CATIA and AGPS), due to their relative simplicity, ease of programming, and hence reasonable robustness, it has been noted by many (theoretical and applied scientists alike) that any robust geometric design/gridding system should have the capability of handling *unstructured* meshes for the simple reason that a majority of surfaces are not compatible with the topology of regular meshes. In fact, strictly speaking, only surfaces that are topologically equivalent to a portion of the plane or a torus can be modeled well using regular meshes. Even a simple surface such as a sphere poses a significant obstacle since, as is well known, it is impossible to cover a sphere with a quadrilateral or triangular mesh.

As a result, existing modeling/gridding systems that are restricted to structured meshes (such as AGPS) must make a deliberate effort to either avoid such settings or employ special "tricks" to get around the problem. For example, one often uses the method of

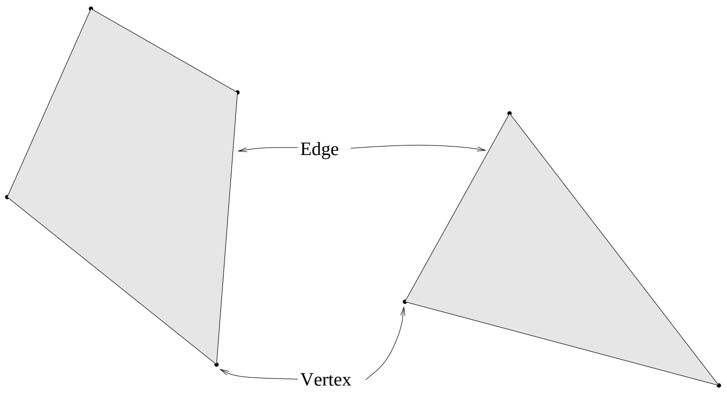

Figure 4: Quadrilateral face (left), and triangular face (right).

trimming, which means that the parameter domain of a rectangular (Bézier or spline) patch (*i.e.,* a patch parametrized by a rectangle), is "trimmed" to become a "triangular" patch. However, it is well known that this solution is far from optimal and has many problems associated with it. In particular, trimming is expensive and prone to numerical error, and it is in general difficult to maintain smoothness at the seams of the patchwork of trimmed (NURBS) surfaces [13]. The AGPS trimmed surface structure is not fully incorporated into AGPS and its functionality is limited (according to the AGPS manual, "the operations that may be performed on trimmed surfaces will be limited to those for which suitable modifications have been implemented. Currently only the CTS (Create-Trim-Surface) command will create a trimmed surface and only the DRA (Draw) and SHA (Shaded-Graphics) commands will accept them".)

Another difficulty with using structured meshes exclusively is that often one is forced to model surfaces using such meshes even when this is not "natural". For example, in the above mentioned example of a sphere, it would be natural to use a mesh with triangular faces in the vicinity of the poles, instead of trimmed quadrilateral patches. Another potential difficulty is with the parametrization. For example, AGPS constructs a patch in the shape of a circular disk (via the AGPS command file `cir2sur.com`) by subdividing the circle into four curves, which are then used as the sides to create a rectangular patch. This produces a parametrization that is *singular* at the corner points, which is not desirable.

There are indeed many other situations where unstructured meshes would be useful. A typical situation is when two or more surfaces are to be joined smoothly. Since such surfaces are usually created separately, the corresponding meshes will typically not join to form a regular mesh. For example, the two surfaces to be joined could be the body and the wing of an airplane (the so-called wing-body fairing problem), each of which is represented as a tensor-product spline (bicubic, say), corresponding to a regular quadrilateral mesh (see Figures 1 and 2). One would like to construct the merged mesh (and the corresponding surface) that is the union of the two meshes. Unfortunately, such a composite mesh is in general no longer regular (see Figure 3 where irregular vertices (vertices with valence other than 4) are highlighted with a "*"). Naturally, it would be useful if we could use this new composite mesh, even if it is no longer regular. Thus, we need a technique for creating surfaces for unstructured meshes.

The need for supporting unstructured meshes was also advocated by the AGPS group (see [6]). AGPS has experimented with data structures that can handle unstructured forms. Currently, according to [6], a part of AGPS can use triangulations to "subdivide surfaces based on chord height, maximum triangle area, and maximum edge length area criteria". To quote from [6], the "advantages of this approach include: less memory required by the flow solver because fewer elements are needed to define the flow region, there is better control of grid density, and a given geometry definition can be modeled more accurately and in less time".

Another reason why unstructured meshes are useful is for adaptive mesh refinement techniques (*e.g.*, for multi-resolution type applications). For example, consider the regular mesh in Figure 5 (left). In order to refine this mesh while preserving the regular structure, an entire row or column of vertices must be inserted into the mesh, as illustrated in Figure 5 (right). If we wish to add detail to the mesh locally, this type of (global) refinement is not appropriate or desirable for obvious reasons. Therefore, the mesh (and consequently, the surface) cannot be refined locally without the availability of unstructured meshes. Figure 6 illustrates how local refinements of a regular mesh can give rise to an unstructured mesh. Local adaptive refinement is also of interest for solving Euler and Navier-Stokes equations since it greatly enhances the possibility of generating meshes with varying spatial resolution and/or nested grids for multigrid-type solvers.

As we shall see below, the question of whether subdivision methods are the most appropriate technique for constructing surfaces over irregular (unstructured) meshes, and whether existing software systems at Boeing would benefit from such methods cannot be presently fully answered. However, we are convinced that the answer to the question of extending current design and grid generation methods to unstructured meshes, is a resounding yes. Unstructured meshes do away with the rigidity of the structured setting, thus allowing one to model surfaces of arbitrary topology, and we believe that any modern robust system should employ them in a fully functional way.

3 Surfaces Built from Unstructured Meshes

Once a mesh has been defined, in many settings we need to construct a surface which corresponds to the mesh in some sense. For example, we may want the surface to interpolate the mesh vertices or approximate them in a prescribed way. Thus, the mesh can be thought of as a "frame" that gives a basic idea of the shape of the surface, and its vertices are the "control points" since they control the location and shape of the surface.

For structured meshes (*e.g.*, regular quadrilateral or triangular), the traditional methods are based on tensor-product B-splines (NURBS) and the classical Bézier polynomial/rational rectangular or triangular patches. As for unstructured meshes, the most common constructions include:

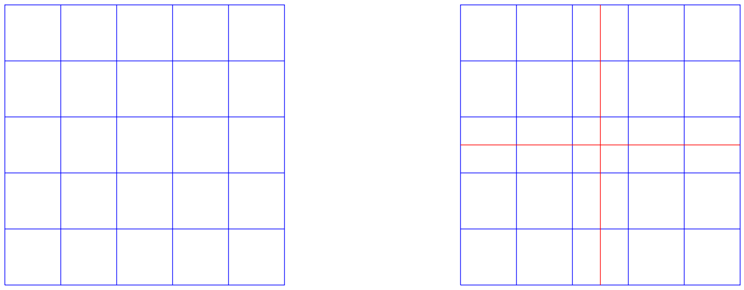

Figure 5: A regular mesh and its global refinement.

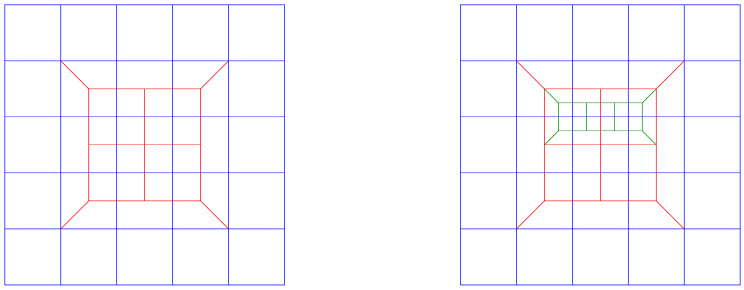

Figure 6: A local refinement of a regular mesh giving rise to unstructured meshes.

(1) Polynomial/Rational parametric patches (rectangular and triangular)

(2) Transfinite (or blending) methods

(3) Implicit patches

(4) Bivariate B-splines

(5) Variational surfaces (thin plate splines)

(6) Subdivision surfaces

Items (1) and (2) are the most common types of surfaces in computer-aided geometric design. In this case, the surface is represented as a collection of rectangular or triangular patches, each of which interpolates (or otherwise approximates) data associated with a given face. For the first class of methods, the data are usually the face vertices, possibly with surface normals and other information associated with the vertices, whereas for the second class, the data consist of a network of curves, for example the patch boundary curves (obtained in various ways), that are then "blended" together to form a patch. The patches are joined so as to form a globally smooth surface, where "smooth" depends on the application at hand. Surfaces that represent an aircraft model should be at least C^1 and preferably C^2 since curvature is known to have a major impact on both the quality of the surface and the numerical stability of CFD computations. We refer the

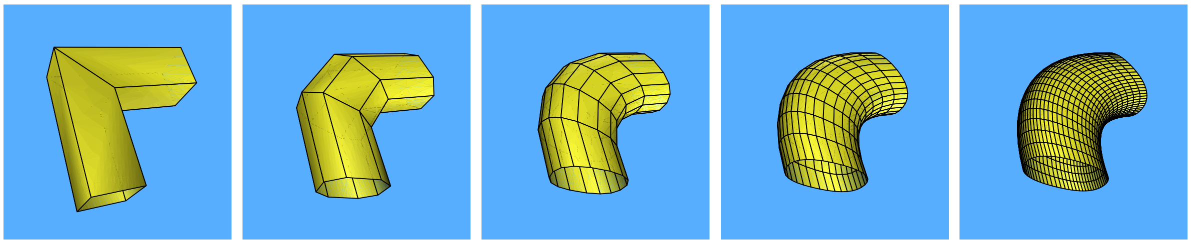

Figure 7: A prism and its smoothed out versions.

reader to [5] and [32] for surveys on these two types of methods. The idea behind the third class of surfaces is similar to the previous two types, except that implicit patches are used instead of polynomial/rational/transfinite patches, see [3] for a survey.

As for (4), the idea is to use generalizations of tensor-product B-splines to represent a surface as a linear combination of such B-splines. Methods based on generalized B-splines are not commonly used, but recent results in this direction show promise (see [31], for a survey).

Lastly, subdivision surfaces can be used to define smooth surfaces. They have emerged mainly as a reaction to the fact that for the traditional types (1) and (2) it is difficult to construct C^1 surfaces from unstructured meshes (let alone C^2), and subdivision surfaces showed promise in this direction. We will discuss this issue in detail after we look into the basic ideas behind subdivision surfaces.

4 What is Subdivision?

While the mathematical description and analysis of subdivision surfaces is far from elementary, the basic idea describing such a surface is very intuitive. In fact, "subdivision surfaces" have been used long before mathematicians have formalized them. Namely, sculptors and wood carvers use the "subdivision technique" to "smooth out" a piece of material by chopping off corners and edges until the material looks and feels smooth. Thus, using our terminology, the initial mesh can be smoothed by cutting off corners and edges, thereby making the mesh finer (*i.e.*, containing more faces) and smoother. An example of a typical subdivision process is illustrated in Figure 7, where we depict a repeated application of a "corner cutting" algorithm.

Translated into a mathematical language, a subdivision algorithm is a process of a mesh refinement, whereby the vertices (control points) of the new mesh are obtained as (usually convex) local combinations of "old" mesh vertices *i.e.*, vertices of the previous refinement. Applying a subdivision scheme repeatedly, a sequence of meshes (*i.e.*, denser and denser organized *sets of discrete points*) is generated which, in the "limit" converge to a continuous surface. The smoothness and the nature of the limit surface is determined by the specific rules for obtaining new vertices from old vertices.

Subdivision is a fast growing area of applied mathematics and computer aided geometric

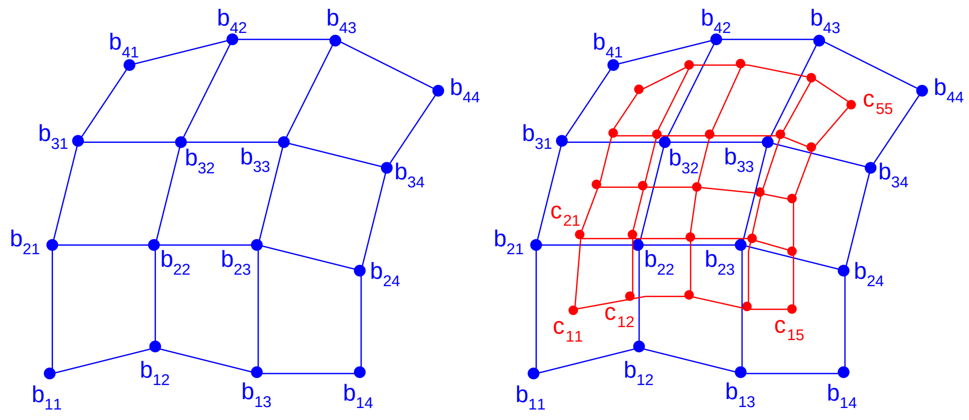

Figure 8: Regular quadrilateral mesh (left), and one refinement (right).

design and it is beyond the scope of this report to survey and describe in detail all aspects of the subdivision theory. Instead, we refer the reader to the following sources for a basic description of subdivision and a survey of relevant results. The paper [9] is a historical survey of existing subdivision algorithms and their mathematical development up to 1988. A newer introduction to subdivision, from the perspective of modeling and animation applications, is the recent SIGGRAPH course notes [56]. A current and fairly comprehensive treatment of the subject can be found in the book [61]. The course notes [19] is a current treatment of multiresolution in geometric design, with a detailed discussion of subdivision techniques in this context.

4.1 Cubic Splines

Although it would seem from the above description that a subdivision surface (*i.e.*, the limit surface of a sequence of meshes) is an entirely new type of surface representation, this is not the case. In fact, the classical univariate and bivariate (tensor-product) splines, among others, can be obtained through a subdivision process. For example, suppose we have a regular quadrilateral mesh and consider the tensor-product cubic spline whose control points are the vertices of the mesh, see Figure 8 (left). Thus, if the control points are labeled as $b_{ij} \in \mathbb{R}^3$, and $\mathbb{Z}$ denotes the set of integers, then the parametric surface corresponding to the cubic spline can be written as

$$s(u, v) = \sum_{i,j \in \mathbb{Z}} b_{ij} B_i^3(u) B_j^3(v),$$

where u, v are the parameters varying over the parameter domain of s, say the entire plane $\mathbb{R}^2$ or a part of it, and B_k^3 stands for the univariate cubic B-spline with integer knots $k-2, k-1, k, k+1, k+2$. Now suppose we insert a new knot half-way between each pair of old knots. If $\mathbb{Z}/2$ denotes the set of half-integers, then the spline s can be expressed over this finer set of knots as

$$s(u, v) = \sum_{i,j \in \mathbb{Z}/2} c_{ij} \tilde{B}_i^3(u) \tilde{B}_j^3(v),$$

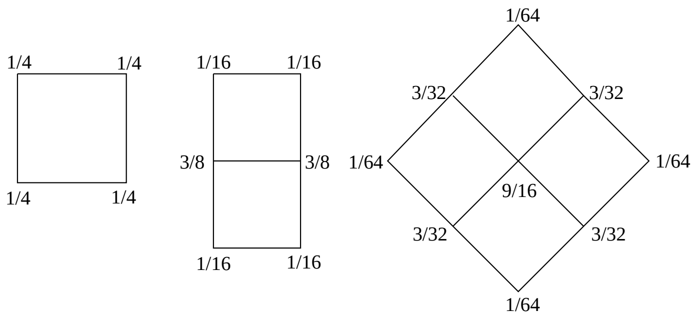

Figure 9: Mask for cubic spline subdivision.

where c_{ij} are the new control points and $\tilde{B}_k^3$ is the univariate cubic B-spline associated with the knots $\frac{k-2}{2}, \frac{k-1}{2}, \frac{k}{2}, \frac{k+1}{2}, \frac{k+2}{2}$. The new control points define a refined mesh as shown in Figure 8 (right). Furthermore, by equating the above two formulations for s (and working through the details) one finds that the new control points c can be expressed as a linear, local combination of the old control points b. The specific rules for computing new control points is given in diagrammatical form in Figure 9 (these diagrams or rules are often referred to as the *mask* of the subdivision algorithm). The leftmost diagram gives the rule for computing a new *face point* (as the average of the vertices of the face). The center diagram gives the rule for computing a new *edge point* (as a weighted average of vertices of the edge and vertices of neighboring faces), and the rightmost diagram gives the rule for computing a new *vertex point* (as a weighted average of the vertex with surrounding vertices). What we have then, is a process for refining (regular) meshes where the vertices of the new mesh are obtained as a local combination of the vertices of the previous mesh, *i.e.*, a subdivision algorithm. The question now is, does the limit surface exist, and if so what are its properties? It is a well-known and remarkable property of splines that this process will indeed converge to a surface, and that this limit surface is the spline s itself (this can be shown to be a consequence of the *convex hull* property of splines which states that a spline curve or surface lies within the convex hull of its control points).

4.2 The Catmull-Clark Subdivision Algorithm

In 1978, E. Catmull and J. Clark had the idea to extend the cubic spline mask to handle irregular vertices (*i.e.*, vertices with valence other than 4), and thus define a general subdivision algorithm for quadrilateral meshes [8]. The mask they defined is dependent on the valence of the vertex, and agrees with the cubic spline mask in the case where the vertex is regular (has valence 4), see Figure 10. Consequently, the Catmull-Clark algorithm produces surfaces with bicubic polynomial patches in the regular part of the mesh, and therefore surfaces that are C^2 everywhere except at the irregular vertices (commonly referred to as *extraordinary points*). By analyzing the mask in matrix form, as is done *e.g.*, in [36], it can be shown that the surface is in fact C^1 at the extraordinary points.

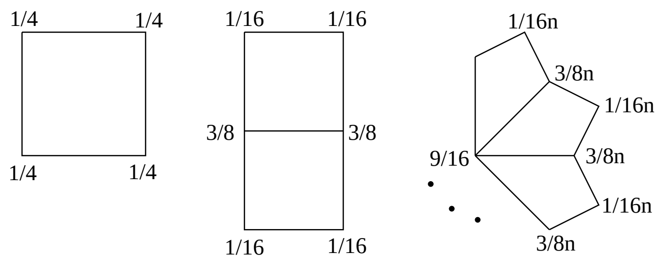

Figure 10: Mask for Catmull-Clark subdivision.

The Catmull-Clark algorithm contains the basic idea found in many other subdivision algorithms (for surfaces), namely the idea of extending a regular subdivision process to irregular meshes.

Notes:

1. The mask in Figure 10 is the *original* Catmull-Clark mask and can be found in [15]. By the time Catmull and Clark had published their paper, they had an *improved* mask which was defined for n-sided faces, and agreed with the original mask when $n = 4$, see [8] or [15].

2. If the initial mesh contains non-four-sided faces, one iteration of the Catmull-Clark algorithm will produce a mesh with all faces four-sided.

3. The number of extraordinary points will become fixed after one iteration of the Catmull-Clark algorithm *i.e.*, the first iteration may introduce additional irregular vertices, but subsequent iterations will not.

4.3 Other Types of Subdivision Algorithms

Subdivision algorithms can be classified into two broad categories; approximating and interpolating. By interpolating, we mean that the vertices of the original mesh are vertices for all subsequent meshes (and therefore lie on the limit surface). Any algorithm that is not interpolating is called approximating. Most subdivision algorithms are approximating for the main reason that approximation is generally superior to interpolation in terms of surface quality. Another difficulty with interpolating algorithms is that the masks are usually "larger" (*i.e.*, new mesh vertices depend on larger sets of old vertices) which results in less local control of the surface. Following is a list of the more common algorithms (see [56] for a more detailed classification).

Approximating algorithms

1. Catmull-Clark algorithm : see above

2. Doo-Sabin algorithm [15]: this is the quadratic analogue of the Catmull-Clark algorithm (*i.e.*, it is based on tensor-product quadratic splines).

3. Loop algorithm [27]: this is an algorithm for triangular meshes based on the three directional box spline.

Interpolating algorithms

1. Butterfly algorithm [18] and its improvements [56]: this algorithm is defined for triangular meshes and is an example of a scheme that does not produce piecewise polynomial limit surfaces.

2. Kobbelt algorithm [25]: this algorithm is defined for quadrilateral meshes, and is a tensor-product version of a subdivision algorithm for generating curves [17]. This scheme also does not produce piecewise polynomial limit surfaces.

5 Work Done

As a first step in our investigation of subdivision surfaces we coded a subdivision algorithm in a stand-alone program. Our plan was to run various test cases ranging from simple self-designed meshes to meshes derived from actual Boeing data (*e.g.*, wing-body fairing). We were also interested in getting user feedback, and decided to incorporate a subdivision capability into AGPS. The reason that we chose AGPS is that AGPS is a surface geometry system focusing on engineering analysis, and therefore a natural setting to experiment with subdivision surfaces in terms of engineering applications. Also, with approximately 400 users throughout the company, it would provide considerable feedback.

5.1 Unstructured Meshes

Incorporating subdivision surfaces into AGPS necessitated programming in Java since the next release of AGPS will be in Java (with a gradual phasing out of the pre-Java version of AGPS). This actually turned out to be beneficial, as the object oriented setting was well suited for designing a data structure for unstructured meshes. Our intent was to make the data structure as general as possible to allow for a wide variety of applications, not just subdivision surfaces. Consequently, there are *e.g.*, instance variables in particular classes that we do not make use of, but are there for possible future use. It should also be noted that our main goal was to get a subdivision algorithm working, and not to spend a lot of time on perfecting this data structure. Therefore, there may be areas where one can improve the data structure.

The basic structure revolves around four classes - the vertex class, edge class, face class, and mesh class. The following table lists the instance variables, and what they represent (the table lists only those instance variables necessary to understand the structure of the class).

Class	Instance variables	Description
vertex	coordinates	3D location
	neighbors	neighboring vertices
	edges	neighboring edges
	faces	neighboring faces
edge	vertices	edge vertices
	faces	neighboring faces
face	vertices	face vertices
	edges	face edges
	neighbors	neighboring faces
mesh	vertices	vertices of the mesh
	edges	edges of the mesh
	faces	faces of the mesh

As can be seen from the table, each vertex, edge, face object keeps track of surrounding objects. There may be redundancy in this setup, but it adds convenience and, as mentioned above, can be modified if there is a cleaner structure. We make no restrictions on this general structure in the sense that a vertex can have any number of neighboring edges (equivalently vertices), and a face can have any number of edges (equivalently vertices). This allows for great versatility, not only in programming various subdivision algorithms but for unstructured mesh applications in general.

5.2 Implementation of Subdivision

The subdivision algorithm that seemed best suited for our needs is the Catmull-Clark algorithm. We decided on this algorithm for four reasons. First, this algorithm is based on quadrilaterals (*i.e.*, they generalize uniform tensor-product cubic B-splines) which would make it easier to incorporate into AGPS since AGPS (and other standard commercial software) is based on regular rectangular meshes. Second, Catmull-Clark subdivision results in C^1 surfaces that are also C^2 everywhere but at the extraordinary vertices, unlike the Doo-Sabin scheme, which generalizes piecewise quadratic surfaces, and does not produce C^2 surfaces. Hence Catmull-Clark surfaces seem to be closest to the preferred curvature smoothness (see Section 6 where we explain why obtaining C^2 continuity everywhere seems beyond reach with the current methods). Third, away from the extraordinary vertices, the surface is a standard spline (piecewise polynomial), which is not the case with some other subdivision surfaces (*e.g.*, the Butterfly scheme). Last, quadrilaterals are more suited (than triangles) for preserving symmetries of modeled surfaces (*e.g.*, the symmetries of cylinder-like surfaces such as a wing or fuselage). However, there are cases where triangular meshes may be preferred and, due to the generality of the mesh data structure, it would not be difficult to add subdivision schemes based on triangles *e.g.*, the Loop algorithm.

Currently, we have the capability to evaluate a subdivision surface exactly, patch-by-patch where each patch is parametrized over the unit square [54], [55]. If the initial mesh contains a non-four-sided face, the mesh will be subdivided once globally to produce a mesh with four-sided faces. In this case the parametrization will not correspond to the faces from the initial mesh but to the faces from the refined mesh generated by one subdivision. We also have the capability to calculate Gaussian, mean and principal curvatures, along with surface normals. Using this information we can generate high quality curvature plots, as can be seen throughout this report.

5.3 Interpolation

As was noted in Section 1, a requirement of AGPS was interpolation of the original control mesh. There are two ways one can achieve interpolation - use an interpolatory algorithm (*e.g.*, the butterfly algorithm), or modify an existing algorithm to be interpolatory (see Section 6 for a more detailed description of interpolation and associated issues). Since we wanted to use the Catmull-Clark algorithm (which is not interpolatory), we opted for the modification route. The simplest approach is to leave the original algorithm intact, and instead create a new initial mesh that would produce a surface that interpolates the original mesh. This involves solving a global system, and adds considerably to the execution time. Associated with this problem is interpolating the boundary. This can be handled in a manner analogous to splines, namely add coincident vertices to the mesh at the boundary. We implemented this as a separate method so that we have at this time three types of data fit:

- approximate everywhere
- interpolate the boundary, approximate the interior
- interpolate the entire original mesh

We will provide examples of these fits in the next section.

5.4 Experiments

We next discuss some of the test cases we ran. We experimented with several meshes, most of which were relatively simple and designed by hand. The results for two of these are shown above. The first is a simple mesh obtained by joining four cubes, shown in Figure 11 (left). The mesh consists of a total of 18 faces and has 20 vertices. Note that even though this is a simple test case, it is useful in that the mesh contains extraordinary vertices of valences 3, 5 and 6, and also contains several regular vertices. The next two plots depict the Catmull-Clark surface generated by this mesh. The first of these, Figure 11 (right), shows a wire frame view of the isoparametric lines (obtained by evaluating each of the 18 patches on a 10×10 grid of the unit square), whereas the second, in Figure 12 (left), shows a shaded view of the surface. The next three figures show color-coded plots of the Gaussian, mean, and the maximum principal curvatures, respectively, calculated on a 30×30 uniform grid for each patch. The curvature is

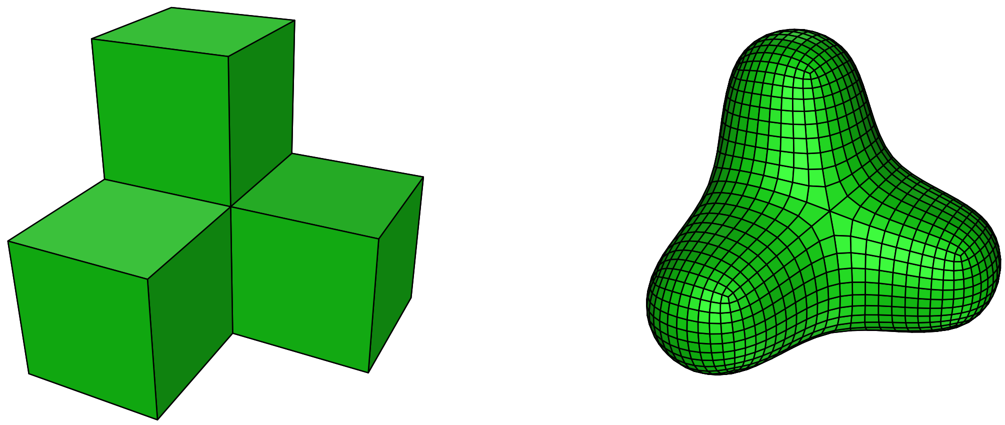

Figure 11: Original mesh and a plot of isoparametric lines for the Catmull-Clark surface.

normalized to be between -1 and 1, and the color scheme is red shades for negative values, green shades for positive values, and blue shades for values near 0.

As far as we can judge from these plots, the bad behavior of curvature near extraordinary vertices, predicted by the mathematical theory, does not seem to show. Note for example, the vertices of valence three located at the tip of the "legs" of the surface. The surface has flat spots at these points but this is not evident from the plots, perhaps with the exception of the plots of mean and maximum curvatures in Figure 13, where the lower-left end part of the surface has a bluish spot. However, this may also be due to the shading effect of the rendering since a light source is pointing directly at the spot.

One would expect even more serious visual defects in the surface in the vicinity of the extraordinary vertices of valence five and six, since in these cases the theory predicts that the curvature is infinite. Again, a visual examination of the surface does not seem to accord with the theory. We know from the theory that the visual defects should be most pronounced in the vicinity of the extraordinary vertex of valence six (there is only one such vertex for the mesh in question, centrally located on the surface). Namely, the theory predicts that the curvature will approach infinity faster than for vertices of valence five. To be more specific, as we half the distance from the extraordinary vertex, the maximal principal curvature is increased (in the limit), by the factor μ/λ^2 (these factors are derived from the matrix representation of the subdivision algorithm). The following tables lists these factors for valences 3, 5 and 6.

valence	μ	λ	μ/λ^2
3	0.166667	0.410097	0.991006
5	0.340107	0.549988	1.124370
6	0.410097	0.579682	1.220413

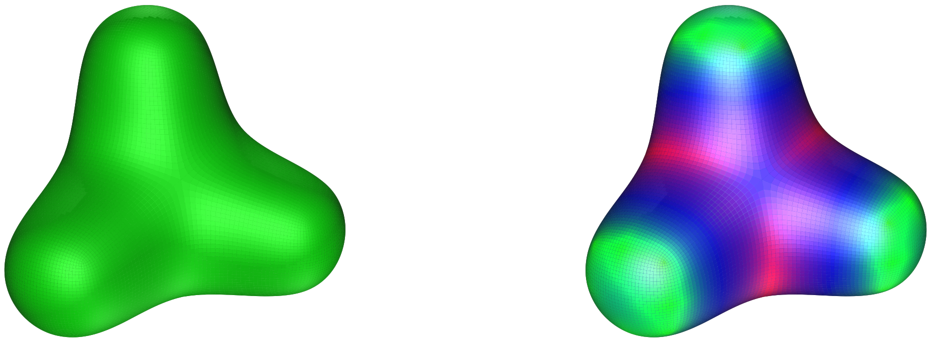

Figure 12: Another view of the Catmull-Clark surface (left) and the same surface color coded with Gaussian curvature.

Thus, for valence 6 the curvature grows approximately 10% faster than for valence 5. Also note that the μ/λ^2 factor for valence 3 is less than 1, which explains why the curvature is zero in the limit. Although we have trouble verifying these mathematical predictions visually, they are confirmed numerically. In Figure 14 below, we show the behavior of the maximum principal curvature for points on the surface approaching the extraordinary vertex of valence 5. These values were obtained for 1,000 points on the surface uniformly distributed along an edge of an extraordinary patch emanating from the extraordinary vertex. The plot clearly shows the increase of curvature as we approach that vertex. On the other hand, the figure also explains why Figures 12 and 13 do not show exceptional curvature singularities - our resolution is far to coarse. In fact, Figure 14 indicates that for a patch of length one meter, we would need a resolution smaller than 1mm in order to observe (visibly) the behavior of the curvature (since as close as 1mm from the extraordinary vertex, the maximum curvature value is still barely larger than the curvature in the middle of the graph). Thus, we can conclude that, at least for this test case, the undesired curvature behavior is not easily recognizable on the actual surface. Perhaps a more thorough testing would be possible with a more powerful graphics workstation than was available to us. The next set of figures (Figures 15, 16, 17) show the same information as above, but for an *interpolating* Catmull-Clark surface (as described in Section 5.3).

The next test case is of a wing/body fairing. It consists of a surface representing the middle section of a body, and a fairing trimmed to where it intersects the body. We gridded each piece by hand to generate a mesh containing 361 faces. Figure 1 depicts the gridded body, and Figure 2 depicts the gridded fairing. Figure 3 shows the grids merged (extraordinary vertices are designated by a "*"). Using this mesh, we generated both the approximating and interpolating Catmull-Clark subdivision surface (see Figure 18, extraordinary patches are colored green). It is difficult to give precise comments about

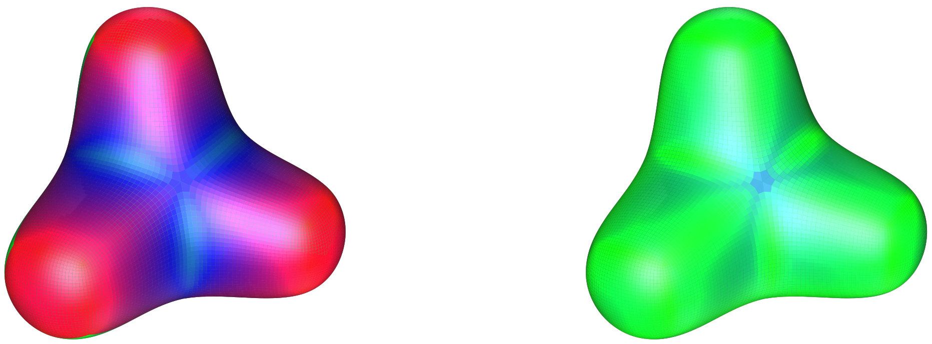

Figure 13: The plot of the mean and maximal principal curvatures.

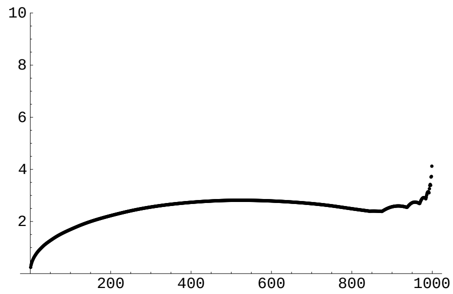

Figure 14: Curvature plot.

the quality of the surface, but one somewhat surprising observation is the apparent quality of the interpolating fit. One would expect the interpolation restriction to cause more of a "wavy" effect.

5.5 Implementation of subdivision into AGPS.

At the conclusion of this project we had not yet incorporated a subdivision capability into AGPS. Consequently, we did not receive user feedback from every-day applications. However, based on the knowledge gained from our investigations thus far, it seems the most natural application for subdivision surfaces is surface grafting. This is due to the fact that grafting a pair of subdivision surfaces requires only combining the initial meshes. This will create a single surface with no gaps, no overlaps, etc. (with no need to partition the parameter domain). Of course, combining the initial meshes may not be an easy task, but it may not be overly difficult either. We envision a system to merge meshes, thereby arriving at a global parametrization (the parameter domain being the original mesh). Thus, for example, to grid the surface, one would only need to grid the mesh (viewed as a piecewise linear surface).

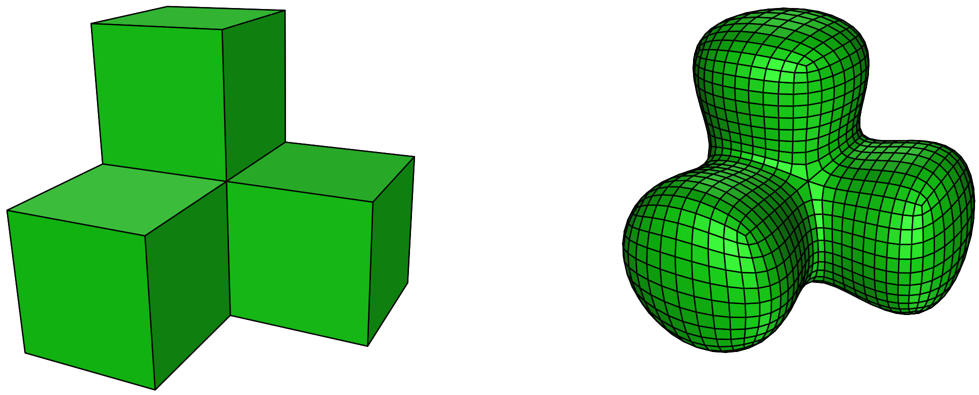

Figure 15: Original mesh and a plot of isoparametric lines for the interpolating Catmull-Clark surface.

As for implementing a subdivision capability into AGPS, we see as the main difficulty interfacing AGPS's regular mesh structure with the unstructured mesh setting required by subdivision. If AGPS is to eventually develop an unstructured mesh capability (as recommended by the authors in Section 7), then it should be fairly straightforward to incorporate subdivision surfaces.

6 The Pros and Cons of Subdivision

The following summary of the pros and cons of subdivision is provided as a quick reference guide. Details are provided in the remainder of the section.

Pros

1. Subdivision surfaces can easily model surfaces of arbitrary topology
2. Subdivision surfaces consists of collections of patches
3. Subdivision surfaces are generally tangent plane continuous
4. Subdivision surfaces are generally curvature continuous almost everywhere
5. Subdivision surfaces do not require trimming
6. Subdivision surfaces can be parametrized globally
7. Subdivision surfaces are robust
8. Subdivision surfaces can be refined locally

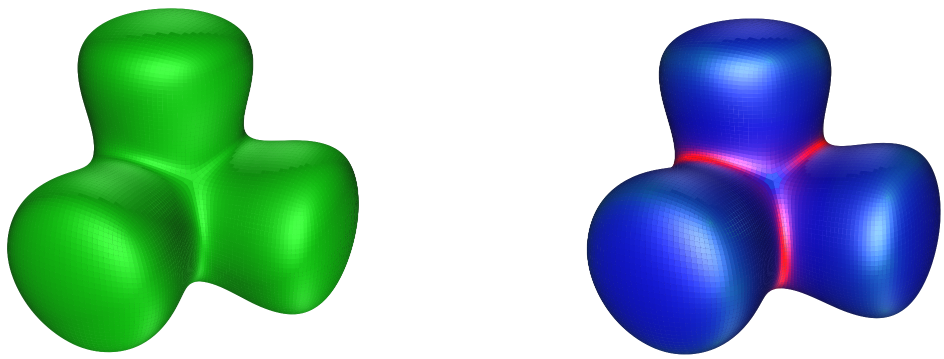

Figure 16: Another view of the interpolating Catmull-Clark surface (left) and the same surface color coded with Gaussian curvature.

Cons

1. Current parametrizations are inflexible
2. The accuracy of subdivision approximation is not well studied
3. Subdivision patches surrounding extraordinary vertices are not polynomial patches
4. Interpolatory subdivision is less understood than approximating subdivision
5. The uneven structure of the subdivision mesh can result in low surface quality near extraordinary points
6. There are no subdivision schemes that produce surfaces with satisfactory curvature
7. Subdivision schemes based on tensor-product splines are generally not refinable

6.1 The Pros of Subdivision

In computer graphics and animation literature, subdivision surfaces are thought of very highly since, among other things, "they can represent robustly and efficiently complex geometries" [56]. Indeed, subdivision surfaces have already proved useful in several applications in the computer graphics world. For example, according to [4], a number of commercial systems use subdivision as a surface representation, including Alias—Wavefront's `Maya`, Pixar's `RenderMan` and `Marionette`, Nichimen's `Mirai`, and Micropace's `LightWave3D`.

Much of the excitement associated with subdivision is certainly deserved due to the many excellent properties of subdivision surfaces, the most important of which, from the engineering perspective, are listed below.

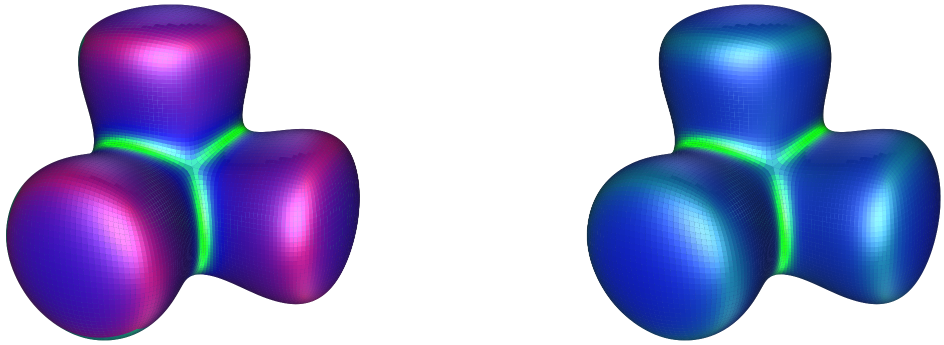

Figure 17: The plot of the mean and maximal principal curvatures.

1. Subdivision surfaces can easily model surfaces of arbitrary topology

Perhaps the single most important property of subdivision surfaces is that they can be defined for unstructured meshes (which makes modeling surfaces of arbitrary topology an easy process). Thus subdivision surfaces free the user from worrying about the topological restrictions that are faced by NURBS modelers.

2. Subdivision surfaces are collections of patches

Subdivision surfaces do not deviate too far from the standard paradigm in CAGD, according to which a surface is represented as a collection of patches. Thus, just like NURBS surfaces, a subdivision surface is a union of patches (triangular, rectangular, or of more general shapes). Moreover, for the most common types of subdivision schemes, *e.g.*, for schemes that are "based on B-splines" (see [36]), these patches are either polynomial, or *piecewise polynomial* (in which case the number of polynomial pieces can be infinite). The latter types of patches, called *extraordinary*, correspond to irregular regions of the mesh. In typical applications meshes are often obtained by joining a finite collection of regular meshes, in which case there is a relatively small number of extraordinary patches (see Figure 3 where the faces of the mesh with an irregular vertex (designated by a "*") give rise to extraordinary patches — all the remaining faces, in fact the majority of all faces of the mesh, are associated with the ordinary bicubic patches). As a result, most algorithms for manipulating and working with such surfaces (*e.g.*, evaluation, surface/surface intersection) can utilize, at least in principle, existing methods.

3. Subdivision surfaces are generally tangent plane continuous

For most schemes (including the Catmull-Clark scheme), C^1 smoothness of subdivision surfaces is guaranteed even as the surface model is modified (unlike some methods based on Bézier patches, in which modifying a patch requires recomputing the Bézier coefficients of the adjoining patches so as to preserve a smooth join). In particular, there is

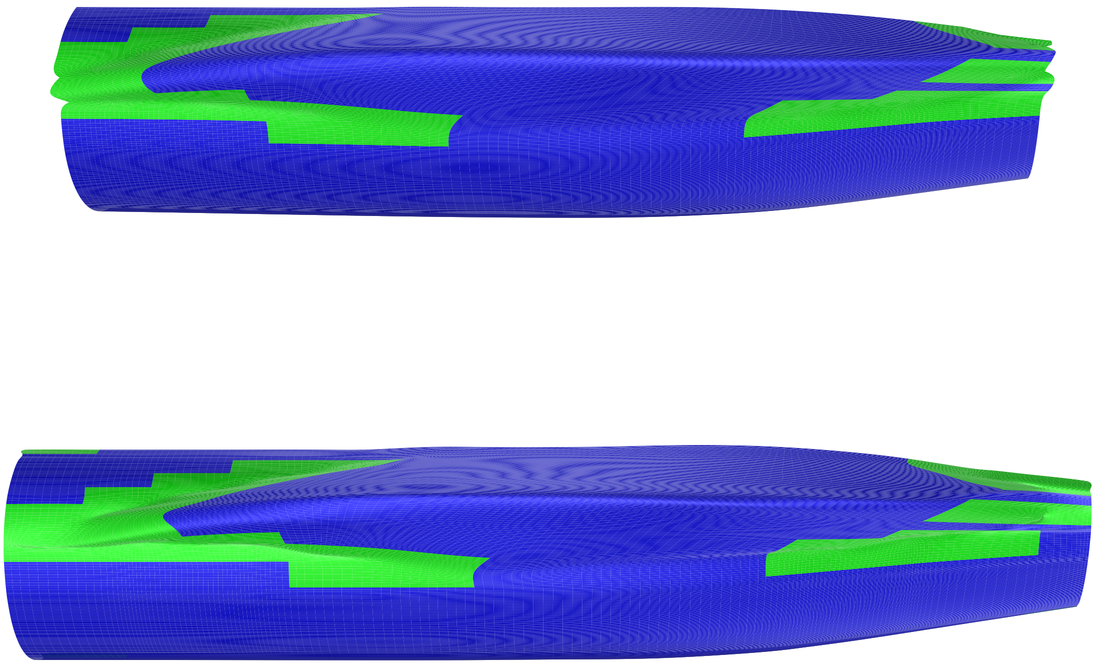

Figure 18: Catmull-Clark surface for the fairing mesh - approximating (top), interpolating (bottom).

no need to globally recompute control points — any set of control points will generate a C^1 surface. In retrospect, this property of subdivision surfaces seems to us to be the main driving force behind the popularity of these surfaces in computer graphics related areas. Namely, more traditional methods of C^1 surface representation appear less intuitive than methods that are capable of producing smooth surfaces simply by "averaging control points".

4. Subdivision surfaces are curvature continuous almost everywhere

Many subdivision schemes (including the Catmull-Clark scheme) generate surfaces that are C^2 everywhere except at a fixed, finite number of points. Hence, if C^2 *everywhere* on the surface is not a strict requirement for a given application, then subdivision surfaces are an attractive alternative to other existing surface representations.

5. Subdivision surfaces do not require trimming

Unlike NURBS surfaces, subdivision surfaces do not require trimming. This is an important feature since it is nearly impossible to enforce tangent plane continuity between trimmed surface patches.

6. Subdivision surfaces can be parametrized globally

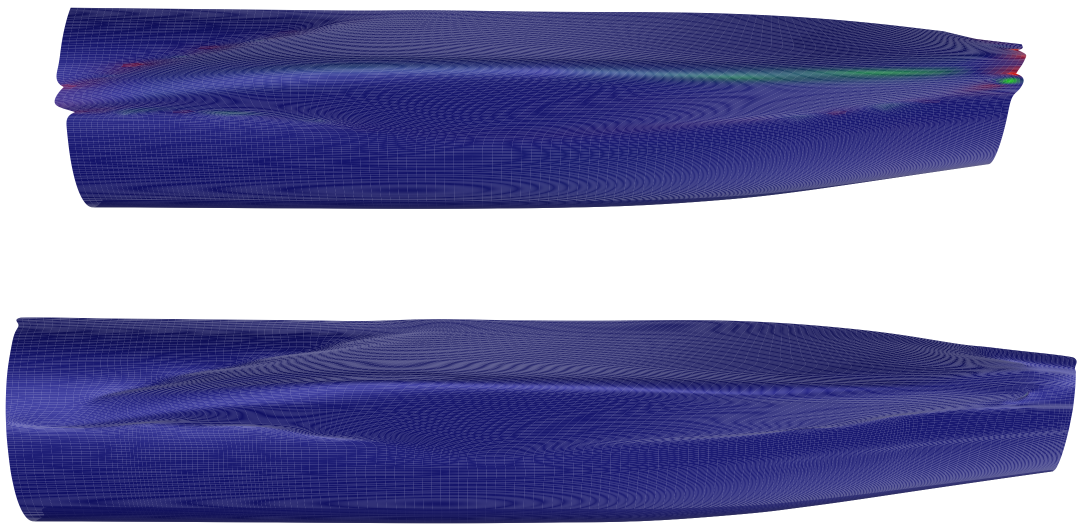

Figure 19: Gaussian curvature of the Catmull-Clark surface for the fairing mesh - approximating (left), interpolating (right).

As is well known, surfaces of arbitrary topology cannot be parametrized using a single planar domain. Instead they are usually divided into segments (*e.g.*, the individual surface patches), each of which is then parametrized using a planar parameter domain (*e.g.*, a square). A useful property of subdivision surfaces is that they can be parametrized *globally*. Namely, one can think of the subdivision mesh, viewed as a piecewise (bi)linear surface interpolating the vertices of the mesh, as a natural parameter domain for the corresponding subdivision surface. For example, in the Catmull-Clark case, each face of the mesh can be thought of as a bilinear surface interpolating the four vertices of the face. Then the collection of all these faces (excluding the boundary faces) is a global parameter domain for the associated Catmull-Clark surface. (Note that each Catmull-Clark patch corresponds uniquely to a non-boundary face of the mesh and thus can be parametrized by it.) As a result, there is a global parametetrization of the subdivision surface.

7. Subdivision surfaces are robust

Subdivision surfaces, and in particular Catmull-Clark surfaces, are robust not only because they can model surfaces of arbitrary topology. They are robust also in the sense that their *representation* is consistent with the classical spline representation. Thus, just as for splines, the subdivision surface can be expressed in terms of *control points* that share many properties with the classical control points, *e.g.*, they have the convex-hull property (the surface is locally contained in the convex hull of these points), they are affinely invariant (an affine change of the coordinate system will have no effect on the control points), and in many other ways they "behave" just like the ordinary control points. As a consequence, the designer/user of subdivision surfaces can remain within the usual spline paradigm (in which moving the control points has a predictable effect on the surface) and hence does not require a major retraining.

Subdivision surfaces are also robust in the sense that numerical manipulation with them resembles the traditional splines. For example, letting control points coalesce results, just as in the spline case, in a decrease of continuity. This is of interest in modeling surface features (such as creases and darts) and in controlling the boundary behavior of the surfaces. Moreover, many spline algorithms, such as evaluation and knot insertion (mesh refinement) algorithms have their counterparts for subdivision surfaces.

Another potential benefit of subdivision surfaces that might contribute to their robustness is the possibility of adding an extra dimension (or dimensions) to the control points (*i.e.,* considering the control points as being n-dimensional rather than three-dimensional). In this way one can represent vector fields over a subdivision surface. This could be useful in cases where the modeling/gridding of a surface is done in conjunction with a finite-element analysis since then an identical representation is employed for both tasks. There is an advantage of a unified framework for geometric design and mechanical analysis, pointed out in several papers, including [11, 23, 26].

8. Subdivision surfaces can be refined locally

A major advantage of subdivision surfaces over tensor-product splines is that they are *locally refinable*, while the traditional splines are not. Thus, a designer using a tensor-product spline can add detail to the surface only by changing the surface representation *globally*. On the other hand, subdivision supports local refinement since it can handle unstructured meshes, which inevitably result from changing the mesh locally. As a consequence, subdivision supports *multiresolution i.e.,* a hierarchical surface representation in which surfaces can be decomposed as a superposition of various levels of detail, with potential applications in multigrid computations, data compression, and wavelet analysis.

As we have mentioned, computer graphics and animation practitioners are highly positive concerning the utility of subdivision surfaces. In particular, the paper [13] describes a successful application of subdivision surfaces (in fact, Catmull-Clark surfaces) in making *Geri's Game* (an animation project initiated by Ed Catmull, the co-inventor of Catmull-Clark surfaces) by Pixar Animation Studios. To illustrate the high level of support for subdivision received from this community, it will be instructive to quote from the paper [13] describing the experience of Pixar employing Catmull-Clark subdivision surfaces in creating Geri's game: "Our experience using subdivision surfaces in production has been extremely positive. The use of subdivision surfaces allows our model builders to arrange control points in a way that is natural to capture geometric features of the model, without concern for maintaining a regular gridded structure as required by NURBS models. This freedom has two principal consequences. First, it dramatically reduces the time needed to plan and build an initial model. Second, and perhaps more importantly, it allows the initial model to be refined locally. Local refinement is not possible with a NURBS surface, since an entire control point row, or column, or both must be added to preserve the gridded structure. Additionally, extreme care must be taken either to hide the seams between NURBS patches, or to constrain control points near the seam to create at least the illusion of smoothness".

Despite the above praise, subdivision methods have several shortcomings that should be pointed out and carefully studied. These shortcomings are sometimes ignored by the computer graphics community as being less relevant for the types of applications that arise in their work. However, they require a thorough investigation when it comes to engineering applications. We next discuss some of the (in our view most serious) shortcomings of subdivision surfaces in more detail.

6.2 The Cons of Subdivision

1. Current parametrizations are inflexible

A common concern of the CAGD and engineering community concerning subdivision surfaces is the existence of a well-defined and useful parametrization. This is a legitimate concern since, as we have seen, subdivision surfaces, unlike NURBS, are not defined parametrically. In fact, it is generally believed that because subdivision surfaces are defined in terms of discrete sets of points, they do not have a natural parametrization. However, as we have already pointed out, this is not the case since these discrete points can be viewed as control points determining well-defined patches, each of which (in the Catmull-Clark case) is parametrized by the unit square. On the other hand, the question remains whether this parametrization is "good" or not.

We believe that this question has not been fully answered in the literature, and in fact there are certain implicit assumptions concerning the parametrization that need to be investigated. In particular, the assumption in most subdivision schemes is that each region of the surface is parametrized in some sense uniformly. To clarify this remark, consider the case of a regular mesh giving rise to a piecewise bicubic parametric surface of the form

$$s(u,v) = \sum_{i,j \in \mathbb{Z}} c_{ij} B_i^3(u) B_j^3(v),$$

where c_{ij} are the control points, and (u, v) vary over a planar domain. As mentioned above, each polynomial patch of this surface is parametrized by the unit square, whether or not the control points c_{ij} are "evenly distributed". For example, one of the surface patches could correspond to a region in the mesh in which all vertices are clustered together, whereas other patches might be associated with vertices of the mesh that are spread out. Therefore, it would seem more natural to use a "non-uniform" parametrization (*i.e.*, a non-uniform spacing of the knot-lines of the B-splines) whenever the mesh has an uneven spacing of vertices (even if the topological structure of the mesh is uniform).

To illustrate this point, consider the case of cubic subdivision for curves (*i.e.*, the univariate version of the Catmull-Clark algorithm). Figure 21 (left) shows the (discrete) curve resulting from subdividing several times the set of control points depicted in Figure 20 (left). It can be seen from the plot of the limiting subdivision curve shown in Figure 21 (right), which turns out to be a cubic spline, that it does not exhibit the same behavior as the curve in Figure 20 (right), a parabola, which was sampled to obtain

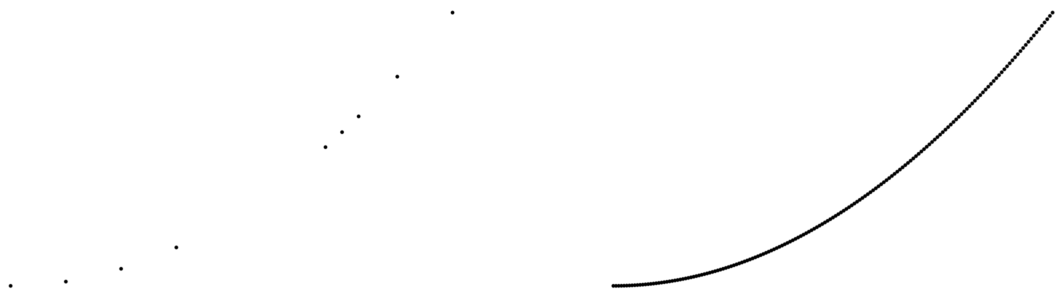

Figure 20: Irregularly spaced control points (left) lying on a parabola (right).

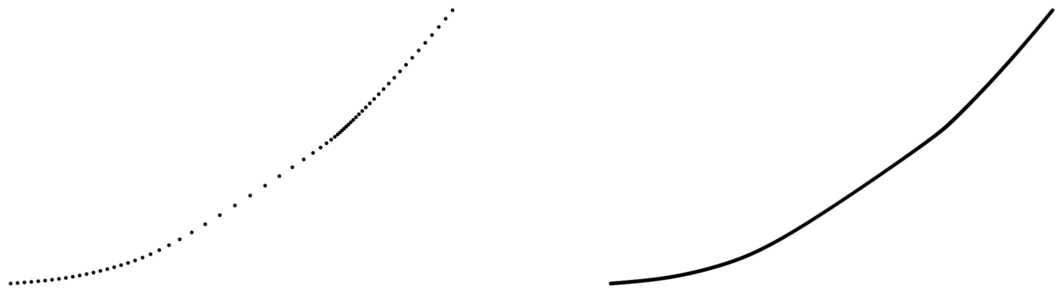

Figure 21: Several cubic subdivision steps applied to the above control points.

the original set of control points of Figure 20 (left). Obviously, the shape quality of the parabola is superior to that of the subdivision curve. In particular, the subdivision curve appears flat in the region with a low density of control points and has a somewhat sharper turn where the control points are clustered. This phenomenon can be attributed to the rigidity of the uniform parametrization (or, equivalently, the uniformity of the knot-set of the cubic spline that is the result of the subdivision) that is implicitly assumed in constructing the subdivision rules. Namely, by sampling the above mentioned parabola non-uniformly, the quadratic curve is written, in effect, as a linear combination of *non-uniform* B-splines. On the other hand, the cubic subdivision algorithm attempts to fit this function with *uniform* B-splines. This explains the qualitative difference in the shapes of the subdivision curve and the parabola. As this example illustrates, the implied uniformity can have visible effects, especially in situations where the vertices (*i.e.,* control points) of the mesh are unevenly distributed. However, in many situations an uneven distribution of control points may in fact be desired *e.g.,* local refinements of a mesh in areas with large variations (see *e.g.,* the discussion in Section 2 on why local refinements may be of interest).

Surprisingly, the above problem of artifacts in the surface caused by inappropriate parametrization does not seem to have been duly addressed in the subdivision literature. We know of only one paper dealing with the possibility of non-uniform parametrizations [52]. In that paper, the authors introduce non-uniform subdivision surfaces that generalize non-uniform tensor-product splines to arbitrary meshes. They use the flexibility provided by the non-uniform parametrization to introduce features such as cusps, creases and darts, while elsewhere maintaining the same order of continuity as in ordinary subdivision. However, the subject of the paper is limited to the topic of introducing features and leaves the question of optimal parametrization (and the choice of the knot-spacing) unanswered.

2. The accuracy of subdivision approximation is not well studied

Related to the parametrization issue, and also relatively unaddressed in the subdivision literature (but see [1] for some recent results) is the issue of accuracy. Theoretical considerations in subdivision research are largely restricted to continuity properties of the limiting surface. While continuity clearly has a decisive impact on the quality of surfaces, it is by no means the only important factor. For example, in the classical Hermite scheme in which a C^1 piecewise cubic curve is obtained by interpolating a set of points along with tangent directions associated with those points, *every* nonzero choice of the lengths of the tangent vectors corresponding to those directions will result in a C^1 curve. Yet, it is well known that not every such curve will "preserve the shape" of the data unless the lengths of the tangent vectors are chosen judiciously.

Another important factor that has an impact on surface quality is *(polynomial) accuracy.* This can be roughly defined as the highest degree polynomial that is preserved by an approximation procedure. For example, every meaningful scheme should have accuracy of at least degree one, which means that if the data (say, a set of points in $\mathbb{R}^3$) all lie in a plane, the approximating/interpolating surface must also be planar. The above failure of the cubic subdivision scheme to preserve the shape of the parabola can be attributed to the fact that the scheme does not have quadratic accuracy since it does not give a quadratic curve when all control points lie on a single quadratic.

There are several reasons to use the criterion of polynomial accuracy for judging the quality of surface approximation schemes. For example, a classical result in approximation theory states that low polynomial accuracy negatively effects the approximation order of the scheme. This means, roughly, that with low polynomial accuracy we cannot expect "fast" convergence of the approximant to the approximated function, as the sampling resolution of the function (from which the data are acquired) increases. While fast convergence is not a guarantee that the scheme performs well for a particular set of data, it is usually a good heuristic principle for constructing data fitting schemes.

The polynomial accuracy issue also arises in connection with subdivision algorithms for regular meshes. There are several results (see *e.g.*, Theorem 8.4 in [10]) that show that accuracy and smoothness (for regular uniform meshes) go hand in hand. In particular, these results show that, under some mild conditions on the subdivision mask, subdivision cannot generate C^k smooth surfaces unless the subdivision scheme preserves polynomials of (total) degree k (in the sense that the subdivision surface obtained for any polynomial sequence of control points is itself a polynomial). Thus, even though this is no longer true for arbitrary meshes (*i.e.*, smoothness does not imply accuracy, as we have seen in the cubic subdivision example), accuracy should nevertheless play a role in the construction and analysis of subdivision methods. In fact, we think it is conceivable that the curvature continuity problem (discussed below) is related to the failure of subdivision schemes to be accurate of degree at least two (see also [40] (Theorem 3.1), for a discussion of this issue).

3. Subdivision patches surrounding extraordinary vertices are not polynomial patches

Another question that does not seem to have been extensively addressed in the literature is whether it is possible to use pure polynomial patches to define the surface around extraordinary vertices. There are at least three natural questions that come to mind in connection with Catmull-Clark surfaces:

(a) Can one use bicubic polynomial patches (instead of patches obtained by subdivision) joined with curvature continuity around an extraordinary vertex?

(b) If (a) turns out to be impossible, can one solve the problem in (a) by allowing a modification of the bicubic patches immediately surrounding the extraordinary patches, while still preserving the overall smoothness away from the extraordinary vertices?

(c) If the answer to (a) and (b) is no, would allowing polynomial patches of (bi)degree larger than three work?

We point out that it is known that (c) has a solution if the polynomial degree is sufficiently high. In fact, the method of Prautzsch [38] uses degree six to solve (c) (Reif also has a method for solving (c), but he uses singular parametrization). However, the question remains whether this is the best one can do.

There are two reasons why the above issue may not be as serious as it seems. First, it belongs to theory of splines, and should be viewed as a shortcoming of splines rather than a drawback of subdivision surfaces. Second, it is improbable that (a) and (b) have a solution (which would explain why we have not seen these questions addressed in the literature, since it is unlikely that such negative results would be published). Moreover, finding an optimal solution to (c) may be either difficult or irrelevant (if it is obvious, which to us it is not, that Prautzsch's solution cannot be significantly improved). However, the question concerning the existence of analogs of subdivision surfaces that are actually piecewise polynomial (*i.e.*, splines in the strict sense) remains. In other words, what is the justification for using subdivision surfaces in the presence of purely piecewise polynomial methods that accomplish (if they do) the same ends? We think that this question deserves an answer.

4. Interpolatory subdivision is less understood than approximating subdivision

The theory of interpolatory subdivision is lagging behind that of approximating subdivision. There are two principal techniques for defining an interpolatory scheme. The first is to construct specific masks that guarantee that after each subdivision step, the refined mesh will contain the original mesh vertices. As a consequence, the limit surface will also contain this mesh. There are only a few subdivision schemes of this type, none of which produce surface quality comparable to the best approximating schemes.

Moreover, none of the current schemes produce piecewise polynomial surfaces (even if the mesh is regular), which further limits their attractiveness for geometric modeling applications. See [56], for a more detailed discussion. Lastly, it is presently unknown whether it is even possible for these types of schemes to generate piecewise polynomial surfaces (at least in the regular regions of the mesh).

Another approach to generating interpolating surfaces is to start with a given approximating subdivision scheme and modify the physical coordinates of the vertices of the mesh in such a way that the subdivision surface for the modified mesh interpolates the vertices of the original mesh. Hence, strictly speaking, this method is not an "interpolatory subdivision scheme". One of the drawbacks of this approach is that it necessitates solving a global system of linear equations, which may be an expensive process for large meshes. Another drawback is that such a scheme is not local since changing a control point (*i.e.,* coordinates of a vertex of the mesh) leads in general to a global change in the surface. Also, there appears to be no results concerning the solvability of this system of equations. While there are somewhat more local versions of the above idea (see *e.g.,* [21]), these modifications still require solving a global system. Unfortunately, a good *local* scheme of the above type does not seem to be presently available.

5. The uneven structure of the subdivided mesh can result in low surface quality near extraordinary points

A typical subdivision scheme (such as Catmull-Clark's), applied several times to a given coarse mesh, produces a refined mesh (*i.e.,* a denser set of control points), which exhibits behavior near extraordinary vertices different from the regular regions of the mesh. In particular, in cases where the subdominant eigenvalue of the subdivision matrix differs from 1/2, the control points tend to cluster around the extraordinary vertex or, the opposite, they become less dense compared to the regular regions. While the structure of the mesh is not directly relevant (since it is the shape of the subdivision patches and not the underlying mesh that are of importance in engineering), the presence of this uneven structure of control points indicates possible defects in the parametrization of the patches. Figure 22 illustrates this case. The figure in the middle shows an example of the classical Catmull-Clark surface, corresponding to a simple mesh – the unit cube depicted on the left. The parameter lines clearly indicate that they tend to become dense near the extraordinary vertex (this phenomenon is even more pronounced for larger valences of extraordinary vertices, see *e.g.,* Figure 11 (right), in which the isoparametric lines for the vertex of valence six are less concentrated near the vertex compared to the rest of the surface — a consequence of the subdominant eigenvalue for this valence being strictly larger than 1/2). Figure 22 (right) shows a modified Catmull-Clark surface, where the weights in the subdivision mask were chosen such that the subdominant eigenvalues of the subdivision matrix are equal to 1/2. It is clear that in this case the structure of the isoparametric lines is much more uniform. On the other hand, one can see that the overall quality of the shape seems to have suffered. Some researchers also report the appearance of ripples near extraordinary vertices due to weight adjustment [56]. For the relatively simple and limited number of test cases that we have analyzed, we have not detected this phenomenon for the Catmull-Clark subdivision. In any case, as we shall

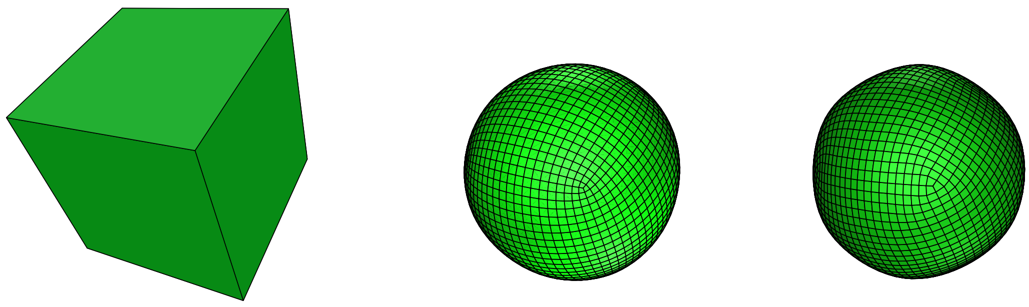

Figure 22: Structure of isoparametric lines near extraordinary vertices for two different subdivision rules (the right two figures).

discuss in paragraph 6 below, it is not clear how to optimize the choice of the weights, and it appears difficult to strike a balance between the various criteria of surface quality, since they seem to be in conflict.

6. There are no subdivision schemes that produce surfaces with satisfactory curvature

As we have discussed in Section 5.4, Catmull-Clark subdivision does not produce C^2 surfaces. In fact, the curvature is usually either zero or unbounded in the vicinity of extraordinary vertices. Moreover, it is not clear (to us) how to best modify the mask to ensure that the curvature is bounded, even though an effort in this direction has been made in [51].[1] Unfortunately, the results of [47, 40, 41] suggest that it is unlikely that one could modify the Catmull-Clark scheme so as to obtain a C^2 scheme (without flat spots). Thus, bounded curvature (but discontinuous at extraordinary vertices) seems to be the best that one can hope for.

The described limitation of Catmull-Clark surfaces also applies to other types of subdivision schemes. Currently, there are no schemes available that are "truly" C^2. In particular, the schemes in [42, 43, 44] are C^2, but they impose zero curvature at extraordinary vertices (*i.e.*, flat spots on the surface, unacceptable in some engineering applications). Similarly, the scheme in [49] is not ideal either, since it uses singular parametrization. A genuine C^2 scheme was proposed in [38], which seems to be superior to most other subdivision schemes. However, strictly speaking this scheme, while it can be generated by a subdivision algorithm, belongs more to the category of piecewise polynomial schemes, since it uses pure bisextic polynomial patches around extraordinary vertices to define a C^2 surface.

There are two other problems closely related to the curvature problem, often cited in the literature (see *e.g.*, [56]), namely the worsened quality of the surface and the

[1]Together with Fred Holt, we have also attempted to choose appropriate weights for the mask of Catmull-Clark subdivision to achieve bounded curvature at extraordinary vertices. This can be done by prescribing the eigenvalues of the corresponding subdivision matrices. In fact, the scheme that produced the surface in Figure 22 (right) has bounded curvature. However, there are many degrees of freedom that need to be set and we do not yet know how to make an optimal choice.

parametrization near the extraordinary vertices as the valence increases, and the presence of ripples in the surface close to an extraordinary point. Both problems can become severe for large valences. However, if a mesh contains vertices with very large valences, then the mesh is most likely ill designed. For our meshes (with valences at most six) and for the Catmull-Clark subdivision, we have not observed such defects visually.

7. Subdivision schemes based on tensor-product splines are generally not refinable

Next to the various benefits of using quadrilateral subdivision schemes (mentioned in Section 5.2), there is also a drawback that should be pointed out. In particular, Catmull-Clark surfaces (or subdivision surfaces based on tensor-product splines) are not refinable. This means that it is in general not possible to represent a Catmull-Clark surface, associated with a given mesh, as a surface on an arbitrary refinement of this mesh. In other words, the function spaces corresponding to successive mesh refinements (*e.g.*, obtained by inserting more and more vertices into an original coarse mesh) are in general not nested. Such surfaces therefore do not support the standard multiresolution paradigm, which requires nestedness. This shortcoming of quadrilateral meshes suggests that using triangulations may be preferable in cases where nestedness is a crucial requirement. This is because subdivision schemes based on triangular meshes typically employ polynomials of total order, which are affinely invariant, as opposed to tensor-product polynomials. A consequence is that piecewise polynomials on triangulations are generally nested *i.e.*, a piecewise polynomial on a given triangulation is also a piecewise polynomial (of the same type) on every subtriangulation.

At this time, it is unclear whether the nestedness problem for subdivision surfaces based on quadrilaterals is a major obstacle. This is because, except for local refinements, we do not know of (and we have not been able to identify) a useful application of multiresolution analysis, wavelets, and hierarchical surface representations (for which the nestedness is typically a crucial requirement) in AGPS, nor any other geometric modeling/gridding system. The one typical application of multiresolution analysis that comes to mind is the so-called level-of-detail editing, which allows separate editing of different "frequencies" of a surface, and is of major interest in computer graphics, visualization, and animation. However, it is not clear whether this type of capability could be useful in current geometric modeling applications at Boeing.

As for the possibility of local refinements of subdivision surfaces, the nestedness of subdivision surfaces is, strictly speaking, a necessary requirement. To explain this, recall that in order to be able to add detail to the surface, the first step is to refine the mesh, M say, corresponding to this surface, which we denote here by $S(M)$. After the refinement we obtain a new mesh, M' say, containing more vertices/faces in regions where we want to modify the surface $S(M)$. In this way we arrive at a new surface, $S(M')$, corresponding to the refined mesh. A natural desire is for the new surface $S(M')$ to be consistent with the old surface $S(M)$ in the sense that the physical coordinates of the vertices of M' can be modified so that $S(M')$ is identical with the original surface $S(M)$. This property may be useful for providing the designer a default choice for the coordinates of the new vertices (or control points) inserted into the mesh M. Using such default locations of the control points will then correspond to not changing the original (non-refined) surface. In order to be able to have this choice for the control points, it is necessary that the underlying piecewise polynomial spaces are nested (which they are not in the case of subdivision schemes based on tensor-product splines). Again, we are not sure whether this drawback is critical. Namely, even if it is not possible to make $S(M)$ and $S(M')$ identical, it is likely that they can be made close enough for practical purposes. Moreover, if the designer decides not to change the shape of the surface after all, he can always go back to the original mesh M.

7 Conclusions and Recommendations

Having discussed a set of theoretical and practical issues pertaining to subdivision surfaces, it is natural to ask whether these types of surfaces can be useful in Boeing engineering applications. In particular, could the AGPS system benefit by employing subdivision surfaces?

We began our investigation into subdivision surfaces well aware of the potential benefits that subdivision methods offer. We were also aware of voices from computer graphics and animation circles applauding the wonders of subdivision for many years. It is somewhat ironic that subdivision has not been widely employed in CAGD applications, since the subdivision theory was mainly developed by researchers in splines who had an eye on potential applications in CAGD. On the other hand, it is not surprising that subdivision has been received with less enthusiasm from the CAGD community than from computer graphics, since the criteria on surface quality are traditionally much less rigorous in the animation world, say, than in engineering applications. For obvious reasons, the surface quality of an airplane must be put under more scrutiny than the surface of a comic character in an animation movie.

The basic conclusion that we can draw from our investigation is that subdivision surfaces, in their present state, are not ripe for a major utilization in geometric modeling applications and are not ready for a large investment in software development in Boeing geometry systems. The main reasons for this conclusion are as follows:

(1) While subdivision surfaces have many excellent properties (see the list in Section 6), most of these properties are also shared (at least in principle) by the more traditional surface representations by rational/polynomial patches. In particular, there are many surface representation schemes using such patches (see *e.g.,* [32] and [12]) that can be used for meshes of arbitrary topology. Moreover, *all* these schemes support subdivision since polynomials and rational functions (surfaces) are refinable. While we are not claiming their superiority over subdivision surfaces (since we have not experimented with such schemes), a basic comparison with the more traditional schemes is in order before making a serious commitment to the subdivision paradigm. Therefore, the introduction of subdivision surfaces is not warranted unless one is convinced (which we are presently not — see item 3 of the list of shortcomings in Section 6) that the same task cannot be effectively accomplished using traditional piecewise polynomials.

(2) Curvature continuity is a property of fundamental interest for geometric modeling and currently there are no C^2 subdivision schemes available, with the exception of schemes that produce flat spots or use singular parametrization. Unless a genuine C^2 subdivision scheme is discovered, there is little justification for using subdivision surfaces, for two reasons. First, even though many schemes are almost C^2, *i.e.*, are C^2 at all but a fixed, finite number of points, the curvature behavior at these points may be unacceptable since the usual schemes (including the Catmull-Clark scheme) may have unbounded curvature. While it is in principle possible to obtain bounded curvature (see the discussion of curvature continuity in Section 6), this issue has not been satisfactorily settled. Second, there are methods for generating C^2 surfaces using traditional polynomial patches (*e.g.*, [33, 38]) which makes choosing a non-polynomial representation which is not strictly C^2 questionable at best.

(3) The current the list of shortcomings and open questions is too extensive (see the list in Section 6).

Our stand formulated above may seem more negative than it really is. The fact that we do not recommend a major commitment to the subdivision technology at this time is based on the current state of development. Future breakthroughs in this young and active research area could well warrant the incorporation of subdivision surfaces into geometric modeling and/or other Boeing software systems.

Also, to be fair, it should be pointed out that traditional surface representations (using quadrilateral and triangular polynomial/rational patches) suffer from many of the same types of problems that plague subdivision surfaces. For example there are not many C^2 schemes (with the notable exceptions of [33, 38] and references therein). Hence, subdivision surfaces could still be very competitive, even with the above shortcomings.

Also, as we discuss in the next section, there are good reasons to continue a focused research effort on a number of specific problems whose solution would remove major obstacles to employing subdivision methods in Boeing geometric modeling systems. Additionally, we recommend continued research on the benefits of unstructured meshes, and their incorporation into AGPS along with a limited subdivision capability based on various versions of the Catmull-Clark algorithm Note that even though subdivision surfaces are not ideally smooth, they are superior in some ways to the currently employed surface representations (for example, at present in AGPS two NURBS surfaces are not easily joined to get a C^0 joint, not to mention C^1 and C^2).

Lastly, we want to emphasize that this report concerns the usefulness of subdivision methods with respect to geometric modeling applications. Therefore, our recommendations should not be extrapolated to other areas or applications (*e.g.*, visualization).

8 Outlook and Directions for Future Work

As we have noted, the findings of our investigation do not imply that subdivision surfaces should be left ignored. There may well be future developments that trigger a justified effort to implement them as part of commercial CAD systems. In fact, given the enormous ongoing research activity in the field of subdivision (as attested to by the many publications in the area), it is very likely that many improvements of the currently available methods as well as new discoveries are forthcoming.

It should be stressed that subdivision methods are an expression of a concentrated effort to represent surfaces over unstructured meshes. Hence, while the type of surface representation provided by subdivision may not be, for the time being, ideally suited for geometric modeling applications, the efforts of the developers of subdivision algorithms communicate quite strongly and unequivocally the need to represent surfaces over unstructured meshes. For this reason alone (although there are many other reasons, see Section 2), we strongly recommend that unstructured meshes be implemented into AGPS and possibly other Boeing software packages. We also recommend further investigation into several issues, detailed later in this section, with the aim of improving the usefulness of subdivision surfaces and/or providing additional evidence as to their utility in various Boeing applications.

In the course of developing our recommendations concerning future directions we have adhered to the following principles regarding geometric modeling/gridding systems. Namely, we believe such systems should be

- robust *i.e.*, must handle unstructured meshes
- expandable *i.e.*, should not be based on a single type of surface representation and must allow incorporation of new methods of representation
- consistent with the spline paradigm, according to which the surface is controlled by control points (the vertices of the mesh)
- hard-wired smoothness *i.e.*, the desired continuity (C^1 or C^2) should be obtained automatically, without requiring user interaction.

In view of the above desirable attributes of a geometric modeling/gridding system, the following list contains recommendations on future research and short-term goals (in both cases we feel that the tasks listed can be accomplished with a relatively modest commitment of resources). The first five involve expanding AGPS to adapt unstructured meshes, and incorporate a limited subdivision capability. We believe this would prove helpful in acquiring a better sense of the usefulness of these new capabilities to the AGPS user in particular, and geometric modelers in general. The last three items deal with specific theoretical/practical problems related to subdivision surfaces that we believe are important to geometric modeling, in addition to being tractable and relatively easily resolved.

1. Incorporate unstructured meshes into AGPS

We suggest following the guidelines given in Section 5.1. This is mostly a matter of incorporating our software into AGPS. In addition, it would be desirable to add basic capabilities to allow storage, display, and basic manipulation of these meshes.

We also recommend an implementation that allows both triangular and rectangular meshes (or their combination). The reason that we recommend including triangular meshes is that, as noted previously, rectangular patches do not give rise to nested spaces whereas triangular patches do. Moreover, it is a much more difficult task to grid (*i.e.*, partition into quadrilaterals) a region or set of points than it is to triangulate it. In fact, for a set of points in the plane it is, in general, not possible to obtain a quadrilateral partition of the points (in contrast it is always possible to triangulate the points). However, this gridding difficulty may not be a serious problem in the context of AGPS since unstructured meshes would most likely be generated by combining structured meshes, and not from gridding a domain or set of points.

2. Investigate joining regular meshes

A common problem in CAGD is joining surfaces to form a single smooth surface. Unfortunately, with the current capabilities of AGPS, this is generally not possible. For example, a wing-body fairing surface in AGPS is obtained by grafting a fairing surface (in Figure 2) onto the body (Figure 1). The grafting procedure constructs the wing-body fairing by first finding the intersection curve of the two surfaces, which is then used to "trim" the domain of the fairing so as to guarantee a continuous abutment. However, by this construction the join is not C^1, let alone C^2.

An alternative to the above grafting process would be to first form a regular mesh out of each surface based on the tensor-product structure of AGPS's surfaces. Next, join the two meshes to create an (usually) unstructured mesh, as seen in Figure 3, giving rise *automatically* to a smooth surface (*e.g.*, a Catmull-Clark surface). Therefore, it would definitely be of interest to have an algorithm for joining regular grids. Note that due to a lack of such an algorithm, the mesh in Figure 3 was hand-gridded – an extremely time-consuming and tedious process.

3. Investigate local mesh refinement techniques

One of the main advantages of unstructured meshes is the ability to refine locally. Hence, we recommend developing algorithms for local refinement of quadrilateral (and possibly triangular) meshes, and incorporating these into AGPS.

4. Incorporate Catmull-Clark surfaces into AGPS

A major benefit of incorporating Catmull-Clark surfaces into AGPS (possibly an improved version that leads to interpolating surfaces with bounded curvature) is gaining experience (along with a user feedback) with surfaces that are defined for unstructured meshes. Implementing these surfaces would be relatively straightforward and low-risk since (1) we already have stand-alone Java code that constructs such surfaces and (2) it would not be a major undertaking to employ a different surface type if it turns out that the Catmull-Clark surfaces are not satisfactory.

5. Investigate gridding subdivision surfaces

Methods should be investigated for gridding subdivision surfaces. This is a challenging problem since surfaces defined by unstructured meshes are no longer parametrized by simple rectangular coordinates. One possibility for gridding a subdivision surface is to use the global parameter domain (the mesh) to define in parameter space, which then defines a grid on the surface.

6. Find optimal weights for Catmull-Clark subdivision

Find optimal weights for Catmull-Clark subdivision that give bounded curvature near extraordinary vertices.

7. Investigate interpolatory Catmull-Clark subdivision

Determine if there is a good *local* scheme for Catmull-Clark subdivision that does not require solving a global system of equations.

8. Investigate the polynomial solution to the C^2 problem

Investigate whether one can use pure polynomial patches to define a C^2 surface in the vicinity of extraordinary vertices of unstructured meshes (see item 3 in the list of shortcomings of subdivision in Section 6).

Finally, an important but more demanding project is conducting a comparison of various existing surface representations including subdivision surfaces, NURBS, bivariate splines, and implicit surfaces.

Acknowledgments.

We would like to thank Dave Ferguson for initiating and making this project possible. We also thank Fred Holt who has kindly devoted his time to discuss several issues concerning curvature continuity of subdivision schemes. Also, Roger Pomeroy, Bill Capron, Tom Dickens, and Randy Maier provided expert knowledge regarding AGPS, and we thank them. Likewise, we are grateful to Johan DeVilliers (University of Stellenbosch, South Africa) who spent a week at Boeing, and contributed to our understanding of interpolatory subdivision schemes and their possible improvements. The second author thanks The Boeing Company for supporting his three-month visit. He also wishes to express his appreciation to all colleagues at Boeing for providing a stimulating environment for conducting this research.

References

[1] Arden, G., Approximation properties of subdivision surfaces, Dissertation, Univ. of Washington, 2001.

[2] Bajaj, C., Surface fitting with implicit algebraic surface patches, in *Topics in Surface Modeling*, H. Hagen (ed.), SIAM, 1992, 23–52.

[3] Bajaj, C., F. Bernardini, and G. Xu, Reconstructing surfaces and functions on surfaces from unorganized three-dimensional data, Algorithmica **19** (1997), 243–261.

[4] Biermann, H., A. Levin, and D. Zorin, Piecewise smooth subdivision surfaces with normal control, SIGGRAPH 2000, 113–120.

[5] Böhm, W., G. Farin, and J. Kahmann, A survey of curve and surface methods in CAGD, Comput. Aided Geom. Design **1** (1984), 1–60.

[6] Capron, W. K. and K. L. Smit, Advanced Aerodynamic Applications of an interactive geometry and visualization system, 29th Aerospace Sciences Meeting, AIAA, Reno, Nevada, 1991.

[7] Catmull, E., A subdivision algorithm for computer display of curved surfaces, Dissertation, Univ. of Utah, 1974.

[8] Catmull, E. and J. Clark, Recursively generated B-spline surfaces on arbitrary topological meshes, Computer Aided Design **10** (1978), 350–355.

[9] Cavaretta, A. S. and C. A. Micchelli, The design of curves and surfaces by subdivision algorithms, in *Mathematical Methods in Computer Aided Geometric Design*, T. Lyche and L. L. Schumaker (eds.), Academic Press, Boston, 1989, 115–153.

[10] Cavaretta, A. S., W. Dahmen, and C. A. Micchelli, Stationary Subdivision, *Memoirs of the American Mathematical Society* **93(453)**, 1991.

[11] Cirak, F., M. Ortiz, and P. Schröder, Subdivision surfaces: a new paradigm for thin-shell finite-element analysis, Internat. J. Numer. Methods Engrg. **47** (2000), 2039–2072.

[12] DeRose, T., M. Lounsbery, S. Mann, Parametric surface interpolation, IEEE Computer Graphics and Applications **12(5)** (1992), 45–72.

[13] DeRose, T., M. Kass, and T. Truong, Subdivision surfaces in character animation, in [56].

[14] Doo, D. W. H., A subdivision algorithm for smoothing down irregular shaped polyhedrons, in *Proceedings: Interactive Techniques in Computer Aided Design*, 1978, 157–165.

[15] Doo, D. W. H. and M. Sabin, Behaviour of recursive division surfaces near extraordinary points, Computer Aided Design **10** (1978), 356–360.

[16] Dyn, N., Subdivision schemes in computer-aided geometric design, in *Advances in Numerical Analysis, vol. II*, W. Light (ed.), Clarendon Press, Oxford, 1992, 36–103.

[17] Dyn, N., D. Levin, and J. Gregory, A 4-point interpolatory subdivision scheme for curve design, Comput. Aided Geom. Design **4** (1987), 257–268.

[18] Dyn, N., D. Levin, and J. Gregory, A butterfly subdivision scheme for surface interpolation with tension control, ACM Trans. Graphics **9** (1990), 160–169.

[19] Floater, M., A. Iske, E. Quak, Principles of multiresolution in geometric modeling, Summer School Lecture Notes, Munich, Germany 2001.

[20] Eck, M., T. DeRose, T. Duchamp, H. Hoppe, M. Lounsbery, W. Stuetzle, Multiresolution analysis of arbitrary meshes, in *Proceedings of the 22nd Annual ACM Conference on Computer Graphics*, 1995, 173–182.

[21] Halstead, M., M. Kass, and T. DeRose, Efficient, fair interpolation using Catmull-Clark surfaces, SIGGRAPH 1993, 35–44.

[22] Holt, F., Toward a curvature-continuous stationary subdivision algorithm, in *Numerical Analysis, Scientific Computing, Computer Science: ICIAM-GAMM* - Hamburg, July 3-7, 1995, Vol. 76, Akademie Verlag, 1996, 423–425.

[23] Kagan, P., A. Fischer, and P. Z. Bar-Yoseph, New b-spline finite element approach for geometrical design and mechanical analysis, Internat. J. Numer. Methods Engrg. **41** (1998), 435–458.

[24] Kobbelt, L., A variational approach to subdivision, Comput. Aided Geom. Design **13** (1996), 743–761.

[25] Kobbelt, L., Interpolatory subdivision on open quadrilateral nets with arbitrary topology, in *Proceedings of Eurographics 96*, Computer Graphics Forum, 1996, 409–420.

[26] Kobbelt, L., T. Hesse, H. Prautzsch, K. Schweizerhof, Iterative mesh generation for fe-computations on free form surfaces, Engng. Comput. **14** (1997), 806–820.

[27] Loop, C. T., Smooth subdivision surfaces based on triangles, Master thesis, Univ. of Utah, 1987.

[28] Lounsbery, M., T. DeRose, and J. Warren, Multiresolution analysis for surfaces of arbitrary topological type, ACM Trans. Graph. **16(1)** (1997), 34–73

[29] Mandal, C., H. Qin, and B. C. Vemuri, Dynamic smooth subdivision surfaces for data visualization, in *Proceedings of the Conference on Visualization '97*, 1997, 371–380.

[30] Nasri, A., Free-form curve generation by recursive subdivision of polygonal strip complexes, SIGGRAPH 1999, 243–244.

[31] Neamtu, M., What is the natural generalization of univariate splines to higher dimensions?, in *Mathematical Methods for Curves and Surfaces: Oslo 2000*, T. Lyche and L. L. Schumaker (eds.), Vanderbilt University Press, Nashville, TN, 2001, 355–392.

[32] Peters, J., Local smooth surface interpolation: a classification, Comput. Aided Geom. Design **7** (1990), 191–195.

[33] Peters, J., Curvature continuous spline surfaces over irregular meshes, Comput. Aided Geom. Design **13** (1996), 101–131.

[34] Peters, J. and A. Nasri, Computing volumes of solids enclosed by recursive subdivision surfaces, Eurographics '97, **16** (1997), 89–94.

[35] Peters, J. and U. Reif, The simplest subdivision scheme for smoothing polyhedra, ACM Trans. Graphics. **16(4)** (1997), 420–431.

[36] Peters, J. and U. Reif, Analysis of algorithms generalizing B-spline subdivision, SIAM J. Numer. Anal. **35** (1998), 728–748.

[37] Peters, J. and G. Umlauf, Gaussian and mean curvature of subdivision surfaces, in *The Mathematics of Surfaces IX*, Springer, 2000, 59–69.

[38] Prautzsch, H., Freeform splines, Comput. Aided Geom. Design **14** (1997), 201–206.

[39] Prautzsch, H., Smoothness of subdivision surfaces at extraordinary points, Adv. Comp. Math. **9** (1998), no. 3-4, 377-389.

[40] Prautzsch, H. and U. Reif, Necessary conditions for subdivision surfaces, 1996, Technical report, Universität Stuttgart, Sonderforschungsbereich 404, Bericht 97/04, 1997.

[41] Prautzsch, H., and U. Reif, Degree estimates for C^k-piecewise polynomial subdivision surfaces, Comp. Math. **10** (1999), 209–217.

[42] Prautzsch, H. and G. Umlauf, Improved triangular subdivision schemes, in *Proceedings of the CGI '98*, 1998, 626-632.

[43] Prautzsch, H. and G. Umlauf, A G^2-subdivision algorithm, Geometric Modelling (Dagstuhl, 1996), 217–224, Comp. Suppl. **13**, Springer, Vienna 1998.

[44] Prautzsch, H. and G. Umlauf, A G^1 and G^2-subdivision scheme for triangular nets, Int. J. Shape Modelling **6(1)** (2000), 21–35.

[45] Reif, U., A unified approach to subdivision algorithms near extraordinary vertices, Comp. Aided Geom. Design **12** (1995), 153–174.

[46] Reif, U., Some new results on subdivision algorithms for meshes of arbitrary topology,in *Approximation Theory VIII, Vol. 2: Wavelets and Multilevel Approximation*, C. K. Chui and L. L. Schumaker (eds.), World Scientific Publishing, 1995, 367–374.

[47] Reif, U., A degree estimate for subdivision surfaces of higher regularity, PAMS **124** (1996), 2167–2174.

[48] Reif, U., A refinable space of smooth spline surfaces of arbitrary topology, J. Approx. Th. **90** (1997), 174–199.

[49] Reif, U., TURBS — topologically unrestricted rational B-splines, Const. Approx. **14(1)** (1998), 57–77.

[50] Reif, U. and P. Schröder, Curvature integrability of subdivision surfaces, Adv. Comp. Math. **14** (2001), 157–174.

[51] Sabin, M. A., Cubic recursive division with bounded curvature, in *Curves and Surfaces*, P. J. Laurent, A. le Méhaté, and L. L. Schumaker (eds.), Academic Press, Boston, 1991, 411–414.

[52] Sederberg, T. W., D. Sewell, and M. Sabin, Non-uniform recursive subdivision surfaces, SIGGRAPH 1998, 387–394.

[53] Snepp, D. K. and R. C. Pomeroy, A geometry system for aerodynamic design, AIAA Systems and Operations Meeting, St. Louis, Missouri, 1987.

[54] Stam, J., Exact evaluation of Catmull-Clark subdivision surfaces at arbitrary parameter values, SIGGRAPH 1998, 395–404.

[55] Stam, J., Evaluation of Loop subdivision surface, SIGGRAPH 1998, CDROM Proceedings.

[56] Subdivision for Modeling and Animation, Course Notes, SIGGRAPH 1998. (URL: http://www.multires.caltech.edu/teaching/courses/subdivision/)

[57] Zorin, D., C^k continuity of subdivision surfaces, 1996, preprint.

[58] Zorin, D., Stationary subdivision and multiresolution surface representations, Dissertation, Caltech, 1998.

[59] Zorin, D. and P. Schröder, A unified framework for primal/dual quadrilateral subdivision schemes, Comput. Aided Geom. Design **18** (2001), 429–454.

[60] Zorin, D., P. Schröder, and W. Sweldens, Interpolating subdivision for meshes with arbitrary topology, SIGGRAPH 1996, 189–192.

[61] Warren, J. and H. Weimer, *Subdivision Methods for Geometric Design: A Constructive Approach*, Morgan Kaufmann Publishers, 2001.

[62] Weimer, H. and J. Warren, Subdivision schemes for thin plate splines, Eurographics '98, **17(3)** (1998), 303–313.