1

Introduction and Overview

Subdivision surfaces can be viewed from at least three different vantage points. A designer may focus on the increasingly smooth shape of *refined polyhedra*. The programmer sees *local operators applied to a graph data structure*. This book views subdivision surfaces as *spline surfaces with singularities* and it will focus on these singularities to reveal the analytic nature of subdivision surfaces. Leveraging the rich interplay of linear algebra, analysis and differential geometry that the spline approach affords, we will, in particular, be able to clarify the necessary and sufficient constraints on subdivision algorithms to generate smooth surfaces. Viewing subdivision surfaces as spline surfaces with singularities is, at present, an unconventional point of view. Visualizing a sequence of polyhedra or tracking a sequence of control nets appears to be more intuitive. Ultimately, however, both views fail to capture the properties of subdivision surfaces due to their discrete nature and lack of attention to the underlying function space. In Sections 1.1/7 and 1.2/8, we now briefly discuss the two points of view not taken in this book while in Section 1.3/10 the analytic view of subdivision surfaces as splines with singularities is sketched out. Section 1.4/12 delineates the focus and scope and Section 1.5/13 gives an overview over the topics covered in the book. A useful section to read is Section 1.6/13 on notation.

The trailing two sections are special. We felt a need to recall the state of the art in subdivision in the regular, shift invariant setting; and to give an overview on the historical development of the topic discussed in this book. In view of our own, limited expertise in these fields, we decided to seek prominent help. Nira Dyn and Malcolm Sabin, two pioneers and leading researchers in the subdivision community agreed to contribute, and their insightful overviews form Sections 1.7/14 and 1.8/17.

1.1 Refined polyhedra

For a graphics designer, subdivision is a tool for automatically cutting off sharp edges from a carefully crafted polyhedral object. The goal is to obtain a finer and finer faceted representation that converges to a visually smooth limit surface (see Figure 1.1/8). In effect, subdivision is viewed here as *geometric refinement and smooth-*

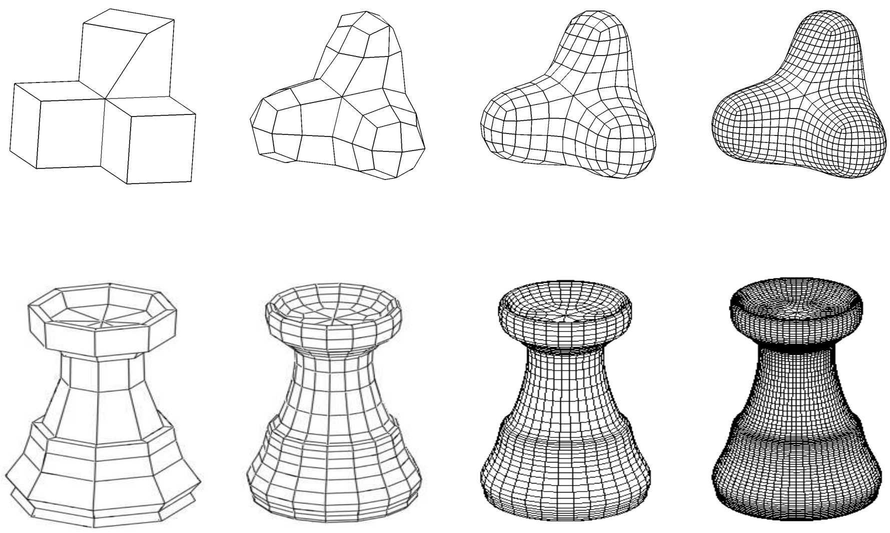

Fig. 1.1: Catmull-Clark algorithm: Starting from a given input mesh, iterated mesh refinement yields a sequence of control nets converging to a smooth limit surface. Vertices with $n \neq 4$ neighbors require *extraordinary* subdivision rules.

ing. This intuitive view of subdivision has made it popular for a host of applications. This book could be faulted for failing to celebrate the rich content that can be generated with such faceted representations that have taken, for example, movie animation by storm. Indeed, we neglect the graphics designer's faceted control polyhedron until Section 8.1/163. This is due to the fact that a number of restrictions and assumptions have to be placed on subdivision algorithms before the notion of a control polyhedron even makes sense. The cases where the control polyhedron is well-defined are therefore justifiably famous and popular.

The actual relationship between the properties of the finite control polyhedron and those of the limit subdivision surface is not straightforward, already for position and more so for higher-order differential geometric quantities. Moreover, in many design packages, the control polyhedron is ultimately replaced by projecting the vertices to points on the limit surface.

1.2 Control nets

For the computer scientist, subdivision is primarily a set of operations on a graph data structure. While the vertices still carry geometric meaning, the edges serve to encode connectivity. Facets play a subordinate role, relevant only for rendering. This point of view, subtly different from faceted approximation, was also taken by the early literature on subdivision surfaces. Subdivision surfaces were correctly characterized

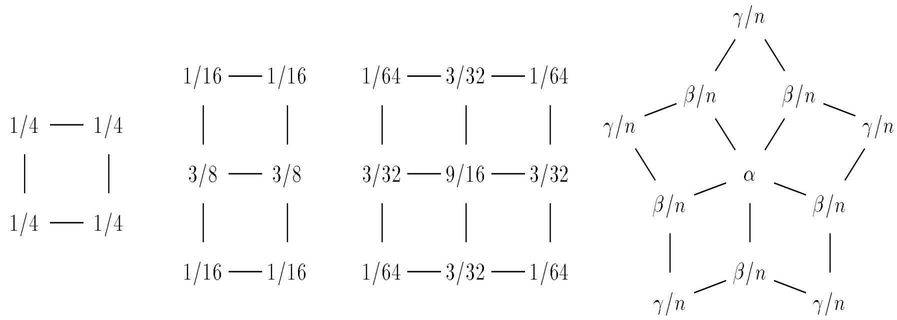

Fig. 1.2: Stencils for Catmull-Clark algorithm: (*left three*) Rules for determining new B-spline control points of a uniform bicubic spline from old ones after uniform knot insertion. The numbers placed at the grid points give the averaging weights: for example (*left*), a new point is generated as $1/4$ of each of four control points of a quadrilateral. The rules establish a new control point for each face, edge and vertex respectively. (*right*) Vertices with $n \neq 4$ neighbors require a rule generalizing the regular case $n = 4$.

as generalizing a property of tensor-product B-splines: where the vertices connected by edges form a regular grid, they are interpreted as the B-spline control net of a uniform tensor-product spline. Representing such splines on a subdivided domain, by a standard technique called ‘uniform knot insertion’, yields a finer regular grid. Figure 1.2/9, *left three*, illustrates this process for bicubic splines.

When the regular grid of control points is replaced by an irregular configuration, the rules of regular grid refinement can obviously no longer be applied. The contribution of the seminal papers [DS78, CC78] are ‘extraordinary subdivision rules’ that mimic the regular rules and apply to irregular networks of points. The vertices of the input polyhedron are taken to be control points and the edges determine how a *mesh refinement operator* is applied (Figure 1.2/9,*right*).

To analyze these extraordinary rules, the early subdivision literature viewed subdivision surfaces as the limit of a sequence of ever finer control nets. The rules of refinement correspond to smoothing operators that map a neighborhood of the control point to an equivalent neighborhood of the corresponding control point in the refined control net. To track the mesh near any given control point, all smoothing operators are placed into the rows of a subdivision matrix. Repeated refinement can then locally be viewed as repeated application of the subdivision matrix to a vector of control points of the neighborhood. This *discrete, linear algebraic* view immediately yields important guidelines for constructing extraordinary rules. In particular, it provides *necessary conditions* for a smooth limit surface that take the form of restrictions on the eigenvalues of the subdivision matrix.

However, the discrete, linear-algebraic point of view fails to provide sufficient conditions since it neglects the functions associated with the control points. The splines defined by the ever-increasing regular parts of the control net give a foothold to tools of analysis and differential geometry.

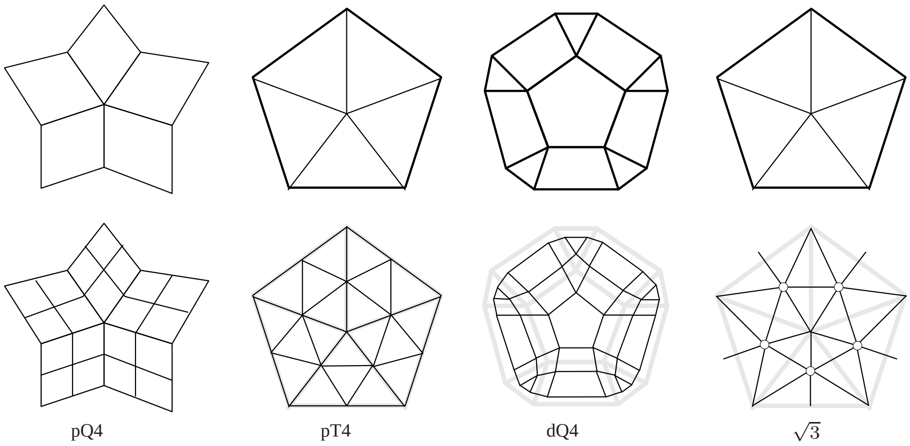

Fig. 1.3: Types of mesh refinement: *top* Initial mesh and *bottom* refined mesh. We focus on algorithms of type pQ4 and dQ4 that result in quadrilateral patches; the analysis and structure of other subdivision algorithms is analogous (see Chapter 9/181).

1.3 Splines with singularities

To adequately characterize the continuity properties of subdivision surfaces, this book emphasizes a third view. As before, where the connectivity allows, the points of the control net may be interpreted as, e.g., spline coefficients. Refinement isolates pieces of the surface where such an interpretation is not possible and extraordinary rules have to be applied. These pieces of the surface are defined as the union of *nested sequences of rings and their limit points*. This approach supplies a concrete parametrization that allows us to leverage tools of *analysis and differential geometry* to expose the structure of subdivision surfaces. We contrast the concepts as follows.

- *Mesh refinement* is generating a sequence of finer and finer control nets, converging to a limit surface. This is the appropriate setup for applications.
- *Subdivision* is generating a sequence of nested rings, whose union forms a spline in the generalized sense. This is the appropriate setup for analytic purposes.

Typically, the resulting objects coincide: the union of rings defines the same surface as the limit of control nets. Because this book investigates analytic properties, it focuses on subdivision in the sense of the second item. The exposition will use the following concepts.

Splines in a generalized sense. The common attribute of the numerous variants of splines appearing in the literature[1] is a segmentation of the domain. Hence, we use

[1] The 'zoo of splines' is a crowded place: A large part of the latin and greek alphabets is already reserved for one-letter prefixes such as for 'B-spline' or 'G-spline'. In addition, there are arc splines, box splines, Chebyshev splines, discrete splines, exponential splines, trigonometric splines, rational splines, simplex splines, perfect splines, monosplines, Euler splines, Whittaker splines . . .

the term ‘spline’ in the following, much generalized sense. A *spline* is a function defined on a domain which consists of indexed copies of a standard domain, such as the unit square for quadrilateral bivariate splines (cf. Definition 3.1/49). Beforehand, we make no assumptions on the particular type of functions to be used. Therefore our use of the word ‘spline’ covers not only linear combinations of B-splines or box-splines[2], but also a host of non-polynomial cases like piecewise exponentials or even wavelet-type functions. The key observation is that we can regard subdivision surfaces as a special case of these general splines. Because, ultimately, we are aiming at the representation of smooth surfaces, we assume throughout that splines are at least continuous.

Quadrilateral splines. We focus on splines defined on a union of indexed *unit squares*, called *cells*, and subdivision that iterates *binary refinement* of these cells. The analysis of subdivision surfaces based on a triangular domain partition (see Figure 1.3/10) is analogous; as is ternary or finer tessellation rather than dyadic refinement (see for example [IDS02, Ale02, ZS01] for various classifications of mesh refinement patterns). Even vector-valued subdivision does not require new concepts but is fully covered by the theory to be developed. Chapter 9/181 lists classes of subdivision algorithms that share the structure of subdivision based on quadrilateral splines and that therefore need not be developed separately.

Splines as union of rings. To properly characterize continuity, the spline domain is given the topological structure of a two-dimensional manifold. This avoids a more involved characterization by means of matching smoothness conditions for abutting patches. The key to understanding subdivision surfaces are the isolated singularities of splines on a topological domain. That is, we focus on the neighborhood of extraordinary domain points where $n \neq 4$ quadrilateral cells join.

In the language of control points and meshes, each refinement enlarges the ‘regular parts’ of the control mesh, i.e. the submeshes where standard subdivision rules apply. At the same time, the region governed by extraordinary rules shrinks. As this process proceeds, a nested sequence of smaller and smaller ring-shaped surface pieces is well-defined, corresponding to the newly created regular region. Eventually, these rings, together with a central limit point, cover all of the surface (see Figure 4.3/67). In the language of splines,

A spline in subdivision form is a nested sequence of rings.

Since all rings are mappings from the same annular domain to $\mathbb{R}^d$, where typically $d = 3$, we can consider spaces of rings spanned by a single, finite-dimensional system of *generating rings*.

A subdivision algorithm is a recursion that generates a sequence of rings

within the span of such a system of generating rings. The word ‘ring’ will not lead to confusion since no rings in the algebraic sense will be considered in this context.

Smoothness at singularities. With rings contracting *ad infinitum* towards a singularity of the parametrization, it is necessary to use, in the limit, a differential ge-

[2] see e.g. [dHR93] or [PBP02,Chap. 17].

ometric characterization of smoothness. Smoothness is measured in a natural local coordinate system. Injectivity with respect to this coordinate system is crucial but not always guaranteed by subdivision algorithms; and the lack of second-order differentiability with respect to the coordinate system presents a challenge for characterizing shape. We therefore devote Chap. 2 of this book to a review of concepts of differential geometry specifically of surfaces with isolated singularities. This differential geometry of singularities is rarely discussed in the classical literature and is crucial for understanding subdivision surfaces.

1.4 Focus and scope

The analysis of subdivision on regular grids has been well-documented and we can point to a rich literature (see Section 1.7/14) on the subject. In particular, [CDM91,p. 18] gives a general technique for evaluating functions in subdivision form, polynomial or otherwise, at any rational parameter value. Differentiability of such functions can typically be established by proving contraction of difference sequences of the coefficients[3] The resulting surfaces are splines in the generalized sense discussed above.

The continuity and shape analysis in this book will therefore focus on the singularities corresponding to 'extraordinary rules'. These singularities are assumed to be isolated so that a *local analysis*, based the union of rings, suffices to establish *necessary and sufficient conditions for* C^1 *and* C^2 *continuity*.

We look at *stationary linear algorithms*. The analysis then combines the discrete, linear-algebraic view with the analytic differential geometric view, i.e., considers both the subdivision matrix and the surface parametrization.

This analysis develops *simple recipes* for checking properties of subdivision algorithms and their limit surfaces. Such recipes are needed to verify the correctness of newly proposed algorithms and to assess their strengths and deficiencies. An important component in deriving these recipes is to make assumptions explicit. For example, if we fail to check for 'ineffective eigenvectors' (Definition 4.19/82), we cannot conclude that the subdivision matrix ought to have a single leading eigenvalue of 1, a property that is often taken for granted. Or, to conclude that a C^1-subdivision algorithm generates a C^1-surface, we need to check that the input control points are 'generic' (Definition 5.1/90). Once such prerequisites have been established, even the 'injectivity-test' becomes simple (see, e.g., Theorem 5.24/111). We illustrate this process for three well-known subdivision algorithms and provide a framework for constructing new algorithms, in particular for generating C^2-surfaces.

[3] The technique relies on the following observation [CDM91, DGL91, Kob98b] [PBP02,p.117]: Let $q^m := [\ldots, q_i^m, \ldots]$ be a sequence with $2^{mk}\nabla^{k+1}q^m$ converging uniformly to zero as m tends to infinity. Then the limit q^∞ is a C^k function and for $j = 0 : k$, the sequences $2^{mj}\nabla^j q^m$ converge uniformly to the derivatives $\partial^j q^\infty$.

1.5 Overview

Chapter 2/21 reviews some little known material on the differential geometry of surfaces in the presence of singularities. The chapter lays the groundwork for most of the proofs in Chapters 3/45 – 7/131 and motivates the sequences of chapters in this book. Chapter 3/45 formally defines the objects of the investigation: splines on topological domains and their forced singularities. Chapter 4/63 introduces the refinement aspect for these splines and defines the resulting class of surfaces obtained by subdivision. We now narrow the focus to stationary algorithms, i.e., algorithms where the same rules are applied at each step.

Chapter 5/89 characterizes stationary subdivision algorithms that generate smooth surfaces, that is, at least C^1-manifolds. While a very general class of algorithms is covered here, particular scrutiny is given to 'standard algorithms' which are characterized by subdivision matrices with a double subdominant eigenvalue. In Chapter 6/115, the resulting powerful analysis techniques are applied to three well-known subdivision algorithms. In Chapter 7/131 we derive constraints that further restrict the class of admissible subdivision algorithms to those that are able to represent the full spectrum of second order shapes. A further restriction of this class finally yields C^2-subdivision algorithms, and we present a new framework for constructing such algorithms. Finally, in Chapter 8/163, we determine bounds on the distance of a subdivision to a proxy surface, and in particular to its control polyhedron. Further, the question of local and global linear independence of systems of generating splines is discussed.

For a *quick tour* through the material, one may proceed as follows. Not skipping the notational conventions in Section 1.6/13 below, Section 2.1/22 is *indispensable* for understanding whatever follows; also Section 2.3/29 should not be missed. In Chapter 3/45, Sections 3.2/47–3.4/53 are fundamental, as well as the whole of Chapter 4/63. In Chapter 5/89, Section 5.3/95 may be skipped on first reading. Chapter 6/115 provides examples by applying the techniques to specific algorithms; its content is not prerequisite to understanding the remaining chapters. Parts of the material in Chapters 7/131 and 8/163 are brand-new. Here, the exposition is less tutorial, but rather intended to prepare the ground for new research in the field.

1.6 Notation

As a mnemonic help, in particular to discern objects and maps into the range $\mathbb{R}^d$ from objects and maps into the bivariate domain, we use bold greek letters for objects and for maps into $\mathbb{R}^2$. For example, planar curves and reparametrizations, such as the 'characteristic' reparametrization, will be represented by bold greek letters. We use plain roman letters for real or complex-valued functions and constants. The constants, λ for eigenvalues and κ for curvature, are an exception to conform to well-established usage. Bold roman font is used, in particular, for points and functions in the embedding space $\mathbb{R}^d$. For example, the subdivision surface $\mathbf{x}$, its normal $\mathbf{n}$ and the control points $\mathbf{q}_i$ are so identified.

Points and functions in $\mathbb{R}^2$ and $\mathbb{R}^d$ are always understood as *row vector*, e.g.

$$\boldsymbol{\xi} = [\boldsymbol{\xi}_1, \boldsymbol{\xi}_2] \in \mathbb{R}^2, \quad \mathbf{x} = [\mathbf{x}_1, \dots, \mathbf{x}_d] \in \mathbb{R}^d.$$

Consequently, linear maps in $\mathbb{R}^2$ and $\mathbb{R}^d$ are represented by matrix multiplication *from the right*. For example,

$$\tilde{\boldsymbol{\xi}} := \boldsymbol{\xi} R, \quad R := \begin{bmatrix} \cos t & \sin t \\ -\sin t & \cos t \end{bmatrix}$$

is a counter-clockwise rotation about the origin by the angle t. We summarize:

Bold greek – point or map into $\mathbb{R}^2$ – row vector
Bold roman – point or map into $\mathbb{R}^d$ – row vector

As in Matlab, elements in a row of a matrix or vector are separated by a comma, while rows are separated by a semicolon. For example,

$$[1,\ 2,\ 3;\ 4,\ 5,\ 6] = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}.$$

We have made an effort to clarify concepts by a consistent use of names. For example, what appears currently in the literature as ‘characteristic map’ is called characteristic ring when we want to emphasize its structure as a map over a topological ring and distinguish it from the characteristic spline that is defined as a union of rings and their limit point (cf. Figure 4.3/67). Replicated from the Index, here are the key variables:

$\mathbf{x}$	spline	$\mathbf{x}^m$	m-th ring of $\mathbf{x}$
b_ℓ	generating spline	g_ℓ	generating ring
e_ℓ	eigenspline	f_ℓ	eigenring
$\boldsymbol{\chi}$	characteristic spline	$\boldsymbol{\psi}$	characteristic ring

Generating splines span the space of subdivision surfaces. They have no relationship with the formal power series of the z-transform that is sometimes called generating function (see also the footnote on page 16).

1.7 Analysis in the shift-invariant setting

Contributed by Nira Dyn

A ‘classical’ subdivision scheme on a *regular mesh* generates a limit object, such as function, curve, surface, from initial data consisting of discrete points (control points), parametrized by the vertices of the mesh. The limit object is obtained by two processes; first by recursive refinements of the control points, based on a fixed local refinement rule, and then by a limiting process on the sequence of control points generated by the recursive refinements.

The theory of subdivision schemes on regular meshes is quite different from the analysis presented in this book. It can be traced back to two papers by de Rham